برنامه نویسی به زبان

پاسکال

تالیف : مهندس مهدی محمّدی

نسخه آزمایشی

خرداد ماه ۱۳۸۸

ارتباط با ما:

W W W.P A S C A L 4 U.B L O G F A.C O M

الگوریتم و فلوچارت:

برای ارائهٔ یک راه حل برای یک مساله سه عامل مهم وجود دارد:

۱- *شناخت دقیق مساله*

- داده ها
- مجهولات
- ارتباط بین داده ها و مجهولات

مثال) میانگین ۵ عدد را محاسبه کنید؟ داده ها ۵ عدد ورودی است و مجهول میانگین اعداد است و رابطهٔ آن حاصل جمع ۵ عدد را تقسیم بر ۵ می کنیم تا مجهول بدست آید.

معمولا بدست آوردن رابطهٔ بین مجهول وداده ها به این آسانی نیست باید در مورد آن فکر و تامل کرد!

۲- *طرح نقشه حل مساله (ارائه راه حل)*: بعد ازاین که مساله را به دقت شناختیم باید برای آن راه حل ارائه کنیم.

راه حل هایی که برای یک مساله ارائه می شود به دو دسته تقسیم می شوند: ۱- راه حل غیر منطقی ۲- راه حل منطقی.(الگوریتمی و غیر الگوریتمی)

راه حلهای غیرمنطقی را نمی توان فرموله کرد و هیچ منطقی ندارند. انسانها برای با توجه به هوش و حواس خود از این راه حلها برای حل مساله استفاده می کنند اما یک رایانه نمی تواند به این صورت مساله را حل کند. مثلا عدد ۴۵۲۹ چند رقمی است؟ انسان با یک نگاه به آن متوجه می شود که چهار رقمی است اما رایانه نمی تواند ارقام را ببیند! یا اینکه عدد ۲۴۶ زوج است یا فرد؟

راه حل های منطقی راه حلی هستند که مساله گام به گام حل می شود و رابطه ایی بین داده ها و مجهولات وجود دارد؟ مثلا اگر عددی را تقسیم بر دو کنیم و باقیمانده صفر شود پس زوج است. برای تشخیص تعداد ارقام یک عدد چه راه حل منطقی ارائه می کنید؟

بنابراین برای طرح یک الگوریتم باید راه حل منطقی مساله را بدست آورید.

۳- *تحلیل را حل مساله*: بعد ازاینکه برای مساله راه حل منطقی یافتیم باید آن را تجزیه و تحلیل کنیم که آیا جواب می دهد یا نه؟ یا اینکه آنرا تعمیم دهیم تا مسائل بزرگتری حل شوند. یا اینکه مساله را به مسائل کوچکتری تقسیم کنیم.

الگورینم: کلمهٔ الگوریتم از نام ریاضی دان بزرگ ایرانی محمد بن موسی الخوارزمی گرفته شده است.

به مجموعه **دستورالعملها** که مراحل مختلف **انجام کاری**(حل یک مساله) را به **زبان دقیق** وبا **جزییات کافی** بیان کرده و در آن **ترتیب مراحل** و **شروع** و **خاتمه پذیر** بودن عملیات مشخص باشد الگوریتم می گویند.

خصوصیات یک الگوریتم را در تعریف آن مشاهده می کنید. منظور از زبان دقیق و جزییات کافی چیست؟

منظور از اجرای الگوریتم دنبال کردن مراحل مختلف آن است. شخصی یا ماشینی که الگوریتم را اجرا می کند مجری الگوریتم نامیده می شود.

یک الگوریتم فقط و فقط یک نقطهٔ شروع باید داشته باشد اما می تواند چندین نقطهٔ پایان داشته باشد.(حداقل یک نقطه پایان باید داشته باشد).

در الگوریتمهای ذکر شده در بعضی از کتابها عبارت R= N-[N \ M] * M را در مشاهده می کنید که منظور از آن R= N Mod M است. منظور از علامت \ تقسیم صحیح است و منظور از [ ] جزصحیح یک عدد است.(Mod و DIV به ترتیب باقیماندهٔ یک تقسیم و خارج قسمت صحیح یک تقسیم را نمایش می دهند.)

برای یاد گرفتن الگوریتم ها فقط تمرین و تلاش به شما کمک می کند. در ادامه چند مثال برای تعدادی مساله بیان می شود.

۱- خروجی الگوریتم زیر کدام است؟

۰- شروع ۱- عدد N را بگیر ۲- M=2 ۳- R=N – [N \ M] * M
۴- اگر R=0 آنگاه برو به مرحلهٔ ۸ ۵- M= M+1 ۶- اگر M<N برو مرحله ۳
۷- N را چاپ کن ۸- پایان

الف) درصورتیکه N فرد باشد آن را چاپ می کند ب) مقسوم علیه های عدد N را چاپ می کند
ج) در صورتیکه N عدد اول باشد آن را چاپ می کند. د) در صورتیکه N دارای بیش از دو مقسوم علیه باشد آنرا چاپ می کند.

۲- خروجی الگوریتم زیر کدامست؟

۱- شروع ۲- A را بگیر ۳- C=0 و B=0 ۴- اگر A<=0 بروی به مرحله ۱۰
۵- D= A Mod 10 ۶- C= C+D*10^B ۷- B=B+1 ۸- A=A \ 10
۹- برو به مرحله ۴ ۱۰ - چاپ C ۱۱- پایان

الف) معکوس یک عدد ب) حاصلضرب ارقام یک عدد ج) حاصل جمع ارقام یک عدد د) عدد ورودی

۳- حاصل اجرای الگوریتم زیر چیست؟

۱- شروع ۲- x=6 و y=2 ۳- a=(x+12) mod 5 ۴- b=y+4*3 \ a
۵- اگر a>b آنگاه a را چاپ کن و به مرحله ۷ برو ۶-b را چاپ کن. ۷- پایان

الف) ۳ ب)۶ ج)۴ د) ۱

۴- در الگوریتم روبرو، مرحله ۵ چند بار تکرار می شود؟ (X=5)

۱- شروع ۲- C=1 ۳- عددX را دریافت کن ۴-Y=C+X ۵- Y را بنویس
۶- C=3*C ۷- اگر C<50 برو به مرحله ۴ ۸ – پایان

الف) 3 ب)4 ج)17 د)18

۵- جای خالی در الگوریتم را طوری تکمیل کنید تا حاصل چاپ اعداد مضرب ۳ دورقمی باشد؟

الف) به مرحله ۲ برو
ب) به مرحله ۳ برو
ج) به مرحله ۶ برو
د) K را چاپ کن و مرحله ۶ برو

۰) شروع
۱) K=12
۲) K را چاپ کن
۳) K=K+3
۴) اگر K>99 آنگاه .....
۵) به مرحله ۲ برو
۶) پایان

۶- کار الگوریتم روبرو چیست؟

۱- A را دریافت کن ۲- K=2 ۳- X=A ۴- A را دریافت کن ۵- اگر X>A آنگاه X=A
۶- اگرK=10 برو به مرحله ۸ ۷- K=K+1 و برو به مرحله ۴ ۸- X را اعلام کن ۹- پایان

الف) چاپ جمع اعداد کوچکتر از 10 ب)یافتن کوچکترین عدد بین ۱۰ عدد دریافتی
ج) یافتن کوچکترین عدد بین دوعدد دریافتی با تکرار ۱۰ بار د) یافتن کوچکترین عدد بین ۹ عدد دریافتی

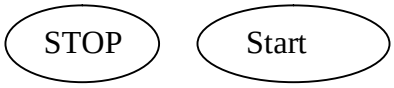
نمودار گردشی (flowchart):

دنبال کردن جملات یک الگوریتم بزرگ کار مشکلی است و جملات به زبانهای مختلفی نوشته می شود که بـرای همـه افـراد قابـل استفاده نیست به همین دلیل برای بیان الگوریتم از نمادهای خاصی استفاده می کنند که الگوریتم را بـه صـورت تـصویری نمـایش می دهد، به مجموعه ای از علایم ساده که الگوریتم را به صورت نماد (تصویر) تبدیل می کنند نمودار گردشی گفته می شود.
نمودار گردشی به زبان خاصی بستگی ندارد، وجود نمودار گردشی امکان اصلاح و تغییر الگوریتم را ساده تر می کند.

علائم نمودار گردشی: در ادامه علامتهای استانداردی که در رسم فلوچارت استفاده می شود را بیان می کنیم.

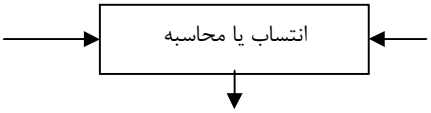
خط جریان: خطوطی که به صورت پلیکان مشخص می شوند و مسیر عملیاتی که را انجام می شود به ترتیب مشخص می کند. **è**

علامت شروع و پایان: بصورت بیضی مشخص می شوند، هر فلوچارت حداکثر یک شروع و حداقل یک پایـان دارد.(شـروع هـر نمـودار منحصر به فرد است)

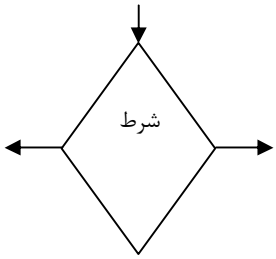


در مورد بیضی شروع(Start) خط جریان به آن وارد نمی شود و در مورد بیضی پایان از آن خط جریان خارج نمی شود، اما به بیضی پایان می تواند چندین خط جریان وارد شود.

علامت پردازش: از شکل مستطیل برای انجام و محاسبات و انتساب استفاده می شود، درون مستطیل می توان یک یا عمل محاسباتی یا انتساب نوشت (در این صورت ترتیب اجرا از بالا به پایین است)، از مستطیل فقط یک پلیکان خارج می شود ولی می توانـد چنـدین پلیکان به آن وارد شود.

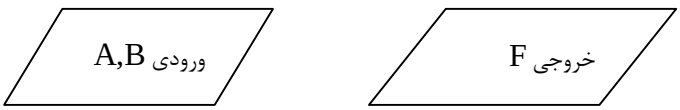


علامت تصمیم گیری: از شکل هندسی لوزی برای تصمیم گیری استفاده می شود، در داخل لوزی یک شرط نوشته می شود که جواب آن یا درست(TRUE)یا نادرست(FALSE) است، از لوزی حتما فقط دو پلیکـان (YES , NO) خـارج مـی شـود ولـی مـی توانـد چندین پلیکان به آن وارد شود.

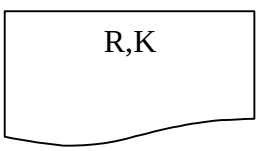


علامت ورودی و خروجی:

از علامت متوازی الاضلاع برای ورودی و هم خروجی استفاده می شود، اگر به عنوان ورودی مورد نظرباشد عبارت ورودی یا Input در آن نوشته می شود و اگر به منظور خروجی استفاده شود عبارت خروجی یا OutPut در آن نوشته می شود.



در متوازی الاضلاع ورودی(Input) نمی توان ثابت یا عبارت نوشت فقط باید متغیر نوشت، به متواری الاضلاع می تواند چندین پلیکان وارد شود اما فقط یک پلیکان خارج می شود.
در بعضی از کتابها علامت خروجی را به صورت روبرو مشخص می کنند.



علامت رابط(ادامه): برای قطع اجرای الگوریتم و ادامه دادن آن از محلی دیگر از علامت رابط استفاده می شود. علامت رابط به صورت یک دایره است. داخل دایره یکی از حروف انگلیسی را نوشته و یک دایره دیگر که محل بعدی اجرای الگوریتم را مشخص می کند همان حرف انگلیسی را می نویسیم.

(A)

زیرالگوریتم:

گاهی برای حل یک مساله بزرگ باید مساله را به مسایل کوچکتری تقسیم کنیم و سپس با حل مسئله های کوچک مسالهٔ بزرگ حل می شود. یا گاهی بخشی از یک الگوریتم چندین بار تکرار می شود که دراین صورت شما می توانید قسمت تکرار شونده را به صورت یک قسمت مستقل نوشته و آن را به دفعات صدا بزنید. زیر الگوریتم بخشی از الگوریتم اصلی است که برای حل قسمتی از مساله نوشته می شود.

مزایای استفاده از زیرالگوریتم ها: ۱- سبب کاهش حجم الگوریتم اصلی می شوند ۲- با اعمال تغیراتی در زیر الگوریتم می توان الگوریتم اصلی را تغییر یا تعمیم داد. ۳- سبب ساده تر شدن روش حل یک مساله می شود زیرا مساله به مسایل کوچکتری تقسیم شده است. ۴- واگذاری نوشتن هر زیرالگوریتم به یک شخص سبب صرفه جویی در وقت می شود.(کارگروهی)
زیرالگوریتمها همان زیربرنامه ها هستند که در مباحث پاسکال بحث خواهد شد.

تعدادی الگوریتم یا فلوچارت در ادامه آماده است که انتظار می رود در مورد حل آنها فکر کنید.

۱- فلوچارتی رسم کنید که اعداد زوج دورقمی را محاسبه و چاپ کند.
۲- فلوچارتی رسم کنید که مقسوم علیه های یک عدد ورودی را جمع و حاصل جمع را چاپ کند.
۳- الگوریتمی بنویسید که یک عدد از وردوی دریافت کند و زوج و فرد بودن آن را مشخص کند.
۴- فلوچارتی رسم کنید که یک عدد از ورودی دریافت کند و فاکتوریل آن را چاپ کند. 5!=1*2*3*4*5
۵- فلوچارتی رسم کنیدکه عدد ورودی مانند N را دریافت کند و حاصل زیر را چاپ کند.

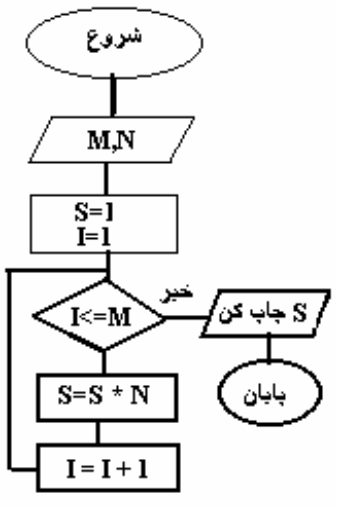
$$S=1/2 + 3/4 + 5/6 + \ldots + N/(N+1)$$

۶- فلوچارتی رسم کنید که یک عدد دریافت کند و معکوس آن را چاپ کند. 5934 **è** 4395
۷- فلوچارتی رسم کنید که اعداد چهار رقمی که از چپ و از راست بکجور خوانده می شوند را چاپ کند مانند اعداد3223 یا 4994
۸- فلوچارتی رسم کنید که یک عدد دریافت کند و رقم بزرگتر آن را چاپ کند.
۹- فوچارتی رسم کنید که سری اعداد فیبوناچی کمتر از ۱۰۰۰ را چاپ کند.
۱۰- فلوچارتی رسم کنید که یک عدد دریافت کند و مبنای ۸ آن را چاپ کند.

تست :

۷- نمودار گردشی روبرو چه هدفی را دنبال می کند؟ (M و N دو عدد طبیعی هستند).

الف) حاصل ضرب M و N را محاسبه و چاپ می کند.
ب)M را به توان N رسانده و چاپ می کند.
ج)N را به توان M رسانده و چاپ می کند.
د)حاصل تقسیم M بر N را محاسبه و چاپ می کند.



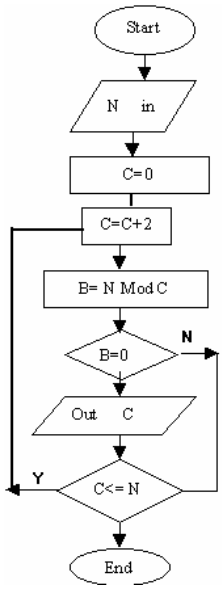
۸- هدف از فلوچارت روبرو چیست؟

الف) چاپ اعداد زوج کوچکتر یا مساوی عدد ورودی

ب)چاپ مقسوم علیه های زوج عدد ورودی

ج) چاپ وارون عدد ورودی

د) در حلقهٔ دائمی قرار می گیرد



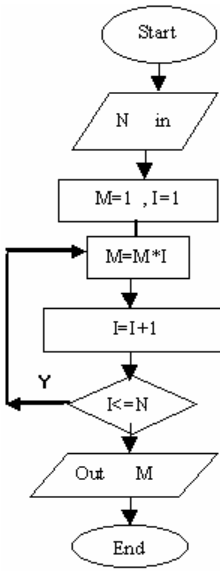
۹- خروجی نمودار گردشی مقابل کدامست؟ N=4

الف) 120

ب) 1 , 1 , 2 , 6 , 24

ج) 24

د)1



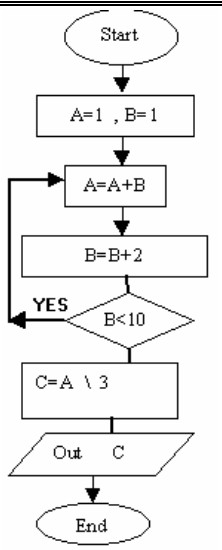
۱۰- خروجی نمودار گردشی مقابل کدام است؟

الف) 8

ب)17

ج) 26

د) 5



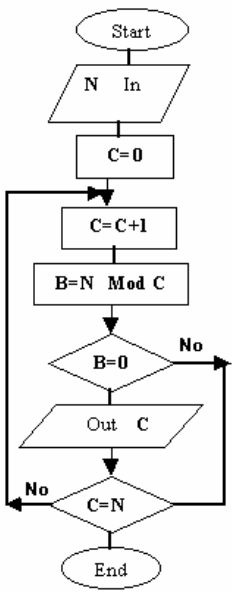
۱۱-کدام شکل هندسی علامت پردازش در یک نمودار گردشی است؟

الف) دایره　　ب) لوزی　　ج) متوازی الاضلاع　　د) مستطیل

۱۲-کدام شکل علامت رابط در فلوچارت است؟

الف) دایره　　ب) بیضی　　ج) مستطیل　　د) متوازی الاضلاع

۱۳- هدف از فلوچارت روبرو چیست؟



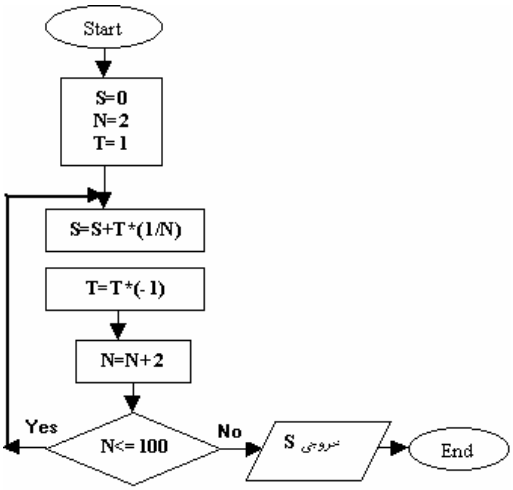
الف) چاپ مقسوم علیه های عدد مورد نظر

ب) چاپ وارون عدد

ج) چاپ اعداد ۱ تا عدد مورد نظر

د) اعداد زوج ۱ تا عدد مورد نظر

۱۴- فلوچارت مقابل محاسبه کدام سری زیر را انجام می دهد؟



الف) 1/2 + 1/4 + 1/6 +….+ 1/98

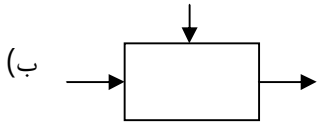
ب) 1/2 - 1/4 + 1/6 -….+ 1/98

ج) -1/2 + 1/4 - 1/6 +….- 1/98

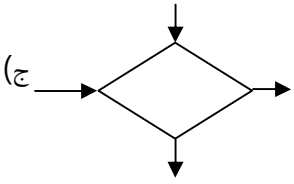
د) -1/2 - 1/4 - 1/6 -…..- 1/98

۱۵- کدام گزینه صحیح نیست؟

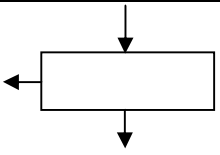
الف)



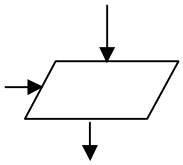
ب)



ج)



د)

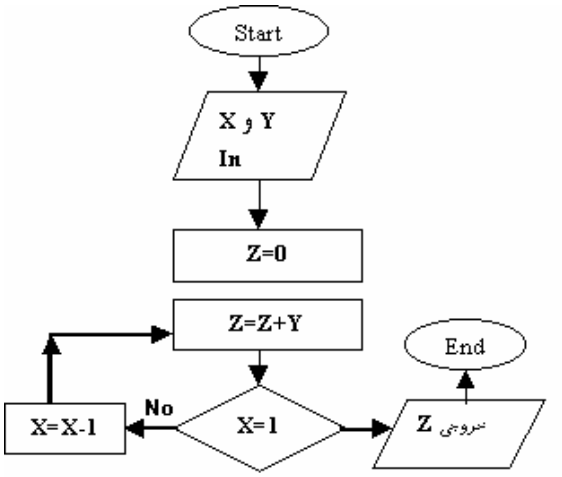


۱۶- الگوریتم روبرو جهت نمایش اعداد۱ الی ۵۰ روی صفحه نمایش است در خط ۵ کدام گزینه را باید قرار داد؟

```
۱- شروع
۲- X= 0
۳- X=X+1
۴- X را نمایش بده
۵- ؟
۶- پایان
```

الف) اگر X<=50 آنگاه برو به ۳
ب) اگر X<50 آنگاه برو به ۳
ج) اگر X>=50 آنگاه برو به ۳
د) اگر X=50 آنگاه برو به ۳

۱۷- فلوچارت روبرو چه خروجی دارد؟



الف) چاپ تفاضل X و Y
ب)چاپ X تقسیم بر Y
ج)چاپ مجموع X و Y
د)چاپ X*Y

پاســکال:

زبان پاسکال توسط پرفسور نیکلاث ویرث ابداع شد. هنگامی که این زبان ابداع شد مترجمهای متفاوتی برای آن از سـوی شـرکتهای متعددی نوشته شد مانند:(لهجه های مختلف یک زبان)

Ansi Pascal
Ms-Pascal
Borland Pascal
Turbo Pascal

در این جزوه سعی شده است با نکات و قواعد زبان پاسکال آشنا شوید و تاکید اصلی بر روی نکات جهت تست است. برای برنامه نویسی باید علاوه بر یادگرفتن قواعد می بایست تمرین و تلاش بیشتری داشت.

مزایا و معایب زبان پاسکال:

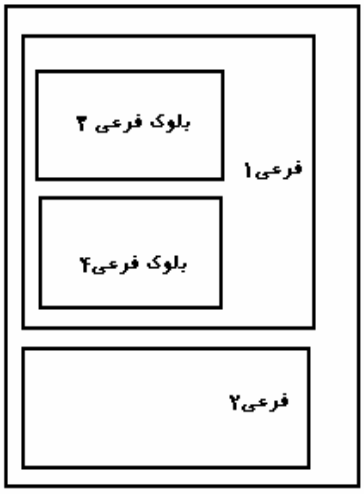
زبانهای برنامه نویسی متعددی جهت برنامه نویسی وجود دارد اما هر کدام برای یک هدف خاصی ایجاد شده اند. به عنـوان مثـال زبـان برنامه نویسی بیسیک جهت آموزش برنامه نویسی به افراد مبتدی ایجاد شده است و هدف آموزشـی دارد. زبـان پاسـکال جهـت برنامـه نویسی ساده و آموزش برنامه نویسی ساخت یافته ایجاد شده است.

اولین زبان برنامه نویسی سطح بالا، زبان ALGOL بود و جهت پیاده سازی الگوریتمهای ریاضی ایجاد شد.

مزایای زبان پاسکال:

۱- قابل فهم بودن دستورات(چون به زبان محاوره ای نزدیک است) ۲- داشتن توابع کتابخانه ای
۳- پشتیبانی از انواع داده ها(اعشاری- صحیح – بولین-رشته ایی – کاراکتری و...) ۴- ساخت یافته بودن

برنامه ساخت یافته به برنامه ایی گفته می شود که از بلوکهای پی درپی یا متداخل تشکیل شده است. برنامه هایی که **ساخت یافته** باشند **سه مزیت** دارند: **۱- اشکال زدایی ساده تر است ۲- خوانایی و درک برنامه ساده تر است. ۳- اعمال تغییرات در برنامه ساده تر است**.



معایب زبان پاسکال:

۱- نداشتن قابلیت انعطاف
۲- نداشتن قابلیت انتقال
۳- عدم امکان تعریف رکورد با طول متغیر

دقت داشته باشید که معایب شاید فقط توسط افراد حرفه ای قابل درک وفهم باشند.

اجــزای یک برنامه:

همانطور که یک کتاب به زبان فارسی از فصل ها و صفحات و پاراگراف ها وجملات
و کلمات و حروف تشکیل شده است یک برنامه پاسکال نیز از بلوک ها و دستورات و کلمات تشکیل شده است. **به کـوچکترین واحـد معنی دار در زبان پاسکال توکن(Token) می گویند**.

به طور کلی یک برنامه از **چهار دسته کلمه** تشکیل می شود:

1 - **کلمات رزرو شــده** : کلماتی هستند که در زبان تعریف شده اندو به منظور خاصی در نظر گرفته شده اند.
مانند **Begin , End , For , Program** و ...

2 - **شناســه ها** : که به دو دسته تقسیم می شوند **شناسه های استاندارد و شناسه های غیر استاندارد- شناسه های اســتاندارد** کلماتی هستند که از **طرف مترجم پاسکال** برای منظور خاصی تعریف شده اند. **توجه داشته باشید که این کلمـات، رزرو شـده نیستند. مانند Readln, Write,ABS و ...**

شناسه های غیر استاندارد شناسه هایی هستند که خود **برنامه نویس آنها** را تعریف می کند. مانند متغیرها ، ثابتها ، زیربرنامه ها نوعها و ...**برنامه نویس می تواند شناسه های استاندارد را مجددا تعریف نماید.**

```
Program Ali;
 Const
   F=13;
 Var
    A:integer;
     B:Real;
Begin
    Read(A);
    A:=A+1;
     B:=A/10;
    Writeln(A,B);
End.
```
یک برنامه پاسکال

3 - **ثابتـــها** : مقادیری هستند که **در طول اجرای برنامه تغییر نمی کنند** مانند عدد ۵ یا رشته 'ALI' . **ثابتها اعداد یا رشته هایی هستند که در برنامه نوشته می شوند.**

4 - **عـــلایم** : **کاراکتر(هایی) هستند که معنی خاصی می دهند.**

مثلا علامت * به معنای ضرب است وعلامت $ برای اعداد در مبنای ۱۶ به کار می رود. $A2=162

شناسه های غیـــر استاندارد :

برنامه نویس در برنامه خود نیاز به متغیرها یا ثابتها و زیربرنامه یا برچسب یا ... دارد که باید خود تعریف کند. هنگامی که می خواهید برای یک شناسه نام انتخاب کنید باید موارد زیر را رعایت کنید:

1- نام متغیر باید با یکی از حروف انگلیسی یا علامت زیرخط(_) شروع شود.

۲- حرف دوم به بعد متغیر می تواند ترکیبی از حروف انگلیسی یااعداد یا علامت زیرخط(_) باشد.

۳- استفاده از کاراکترهای ویژه صحیح نیست . مثل % <) ~ - + * & $! و غیره (جا خالی نیز مجاز نیست)
تنها کاراکتر ویژه که می توان استفاده کرد همان زیر خط است.

۴- حروف کوچک و بزرگ فرقی با هم ندارند یعنی شناسه A و شناسه a فرقی باهم ندارند.

۵- در تورپو پاسکال **حداکثر طول یک شناسه ۶۳ کاراکتر است.**

۶- از کلمات رزرو شده نمی توانید استفاده کنید. مانند For یا While یا Begin و ...

توابع کتابخانه ایی، رزرو شده نیستند. در توربوپاسکال هنگام برنامه نویسی **کلمات رزرو شده** با **رنگ سفید** مشخص می شوند. نام شناسه ها را سعی کنید کلمات بامعنی انتخاب کنید تا خوانایی برنامه را افزایش دهد.

ساختـــار برنامه:

```
عنوان
تعاریف و اعلانات
دستورات یا بلوک اصلی
```

هر برنامه پاسکال از **سه قسمت اصلی** تشکیل شده است:

1 - **قسمت عنوان** 2 - **قسمت تعاریف و اعلانات** 3 - **قسمت بلاک یا دستورالعمل ها**

1 - **قسمت عنوان** : در این قسمت برنامه نویس نام برنامه خود رایادداشت می کند. نام برنامه باید از قوانین نام گذاری شناسه ها پیروی کند.با کلمه رزرو شدهٔ PROGRAM می توان نام برنامه را مشخص کرد.

مانند: PROGRAM FM;

در زبان پاسکال استاندارد این خط *اجباری* است اما در توربوپاسکال نوشتن این خط اختیاری است.

2 - **قسمت تعاریف و اعلانات** : در این قسمت باید شناسه های به کار رفته در برنامه توسط برنامه نویس معرفی شوند مثل متغیرها، ثابتها ، برچسب ها، زیربرنامه ها و ....در پاسکال استاندارد **ابتدا ثابتها** سپس **نوع داده ها** و در آخر **متغیرها** تعریف می شوند. **نماد کلی یک برنامه پاسکال به صورت روبرو خواهد بود:**

```
Program
USES
Label
CONST
TYPE
VAR
BEGIN
مجموعه دستورات
END.
```
نماد کلی یک برنامه پاسکال

3 - **قسمت دستورات** : این قسمت با یک BEGIN شروع می شودو با یک **END.** پایان می یابد که به آن بلوک اصلی برنامه می گویند هر برنامه **فقط یک بلوک اصلی** دارد.

انواع داده ها در زبان پاسـکال:

1 - **داده های عددی** : این داده ها به دو دسته صحیح و اعشاری تقسیم می شوند. داده های عددی صحیح خود دارای انواع زیـر می باشند. هرکدام ازآنها با توجه به محدوده آنها و مقدار فضایی که در حافظه اشغال خواهد کرد توسط برنامه نویس استفاده می شوند. یک برنامه نویس خوب همیشه سعی می کند بهترین نوع را برای متغیرهای خود انتخاب کند.

انواع صحیـح

نوع	مخفف	ظرفیت	محدوده	علامتدار
Byte	Byt	۱ بایت	0 - 255	علامت ندارد
ShortINT	Sht	۱ بایت	-128 - +127	علامتدار
Integer	Int	۲ بایت	-32768 - +32767	علامتدار
Word	Wrd	۲بایت	0 - 65535	علامت ندارد
LongInt	Lng	۴ بایت	-2^{31}-1 تا $+2^{31}$	علامتدار
COMP	Cmp	۸ بایت		علامتدار

داده های اعشاری به انواع زیر قسمت می شوند :

انـواع اعشاری

نوع	مخفف	فضا	دقت اعشار
Single	Sgl	۴ بایت	۸-۷
Real	Rl	۶ بایت	۱۲-۱۱
Double	Dbl	۸ بایت	۱۶-۱۵
Extended	Xtd	۱۰ بایت	۲۰-۱۹

نحوهٔ نمایش اعداد اعشاری در پاسکال به دو صورت است:

۱- ممیز ثابت: مانند 3.45 یا –58.301

۲- ممیز شناور یا نماد علمی: مثل 34E-1 21.2E2 0.32E-1 1.23E1

برنامه نویس می تواند از انواع اعشاری و صحیح مختلف در تعریف متغیرهای خود استفاده کند.

متغیرهایی که از نوع صحیح هستند در محاسبات سرعتشان از اعداد اعشاری بیشتر است. علاوه براین نمایش اعداد اعـشاری کـاملا دقیق نیست. 8 = 7.999999999999

نکته: اگر متغیری را ازنوع Byte تعریف کنیم در این متغیر نمی توانیم عدد 300 را قرار دهیم. چرا؟

تحقیق: اگر در متغیری از نوع Integer عدد 32767 باشد و یک واحد به آن اضافه کنیم چـه اتفـاقی مـی افتـد و در آن متغیـر چـه عددی قرار می گیرد؟

- روش نمایش اعداد صحیح منفی در حافظه به صورت مکمل دو است.
- عدد 4 یک عدد صحیح محسوب می شود اما عدد 4.0 یک عدد اعشاری است.

پدیدهٔ سرریزی:

اگر سه متغیر از نوع Integer تعریف کنیم(A,B,C) آنگاه هر کدام از متغیرها می توانند حداکثر تـا عـدد ۳۲۷۶۷ را در خـود ذخیـره می کنند.اگر A:=32000; و B:=25000; باشد و بنویسیم C:=A+B; آنگاه در C چه عددی ذخیره می شود؟ چون حاصـل جمـع 57000 می شود و این عدد در متغیر C نمی تواند قرار گیرد به همین دلیل قسمتی از جواب در C ذخیره می شود. البته پیغـام خطـا رخ نمی دهد. به همین دلیل ممکن است در متغیر C یک عدد منفی باشد! به این پدیده سرریزی می گویند. ایـن پدیـده هنگـامی رخ می دهد که نتیجه محاسبه ای بیش از ظرفیت تعیین شده باشد. برای جلوگیری از این پدیده چه راه حلی پیشنهاد می کنید؟

2 - **نوع کاراکتر**: فقط یک نوع از این داده وجود دارد و آن نوع **CHAR** است که می تواند **یک** کاراکتر را در خود جـای دهـد و یک بایت فضا اشغال می کند.مخفف آن Chr است. یک کاراکتر می تواند مثال های زیر باشد:

'A' '(' ' ' '#' ' '' ' '5'

یک کاراکتر بین آپستروف قرار می گیرد. (' ')

نکته: هر کاراکتر دارای یک کد است که به آن **کد اسکی** آن کاراکتر می گویند. مثلا کد کاراکتر A عدد 65 اسـت و کـد کـاراکتر B عدد 66 است. این جدول در انتهای کتاب پاسکال وجود دارد. این جدول دارای ۱۲۷ کاراکتر با کدهای آنها است.

نکته: بعضی از کاراکترها قابل مشاهده نیستند. **کدهای بین 0 تا 31 کدهای کنترلی هستند** و مربـوط بـه کاراکترهـایی هـستند کـه مشاهده نمی شوند. مانند کد 7 که مربوط به صدای بیپ است. کد 10 مربوط به کاراکتری است که مکان نما یک سطر پایین مـی رود (کاراکتر LF). کد 13 نیز مربوط به کاراکتری است که مکان نما به ابتدای سطر فعلی می رود(کاراکتر CR).

نکته: منظور از 67# کاراکتری است که کد آن 67 است(همان کاراکتر C است). بنابراین اگر بعـد از علامـت # یـک عـدد بنویـسیم منظور یک کاراکتر است که کد آن را مشخص کرده ایم.

3 - **نوع رشته ایی**: نوع رشته ایی با نام **String** شناخته می شود و در آن می توان یک رشته از کاراکترها را قرار داد. مخفـف آن STR است. مقدار فضایی که آن اشغال می کند به نحوه استفاده از این نوع بستگی دارد. در ادامه این موضوع بحث می شود.

رشته های در زبان پاسکال در آپستروف قرار می گیرند. 'ALIREZA' رشته تهی '' '235'

#65#66 رشته شامل AB #7'ALI' رشته شامل یک کاراکتر بیپ و رشتهALI

- **حداکثر طول یک رشته می تواند ۲۵۵ کاراکتر باشد.**

4 - **نوع بولین**: (**BOOLEAN**) این نوع داده از ویژگی های مهم پاسکال اسـت. در ایـن نـوع داده فقـط دو مقـدار TRUE و FALSE می تواند قرار گیرد. **یک بایت** در حافظه اشغال می کند و مخفف آن Bln است.

متغیـــرها و تعریف آنها: متغیر مکانی از حافظه اصلی است که مقدارش را می توان در طول اجرای برنامه تغییر داد.

برنامه نویس باید همه متغیرهایی که در طول برنامه استفاده می کند را در قسمت تعاریف تعریف کند.برای تعریف متغیرها ابتـدا کلمـهٔ رزرو شدهٔ VAR را بنویسید و در ادامه متغیرها و نوع آنها را به صورت زیر بنویسید.

```
VAR
; نوع متغیر : نام متغیر
```

مثال:

```
VAR
A:INTEGER;
F,H:CHAR;
S:STRING;        N:BOOLEAN;
D:Single;        K:STRING[15];
```

نکته: در مثال بالا چند بایت فضا اشغال شده است؟

در مثال بالا متغیر S از نوع رشته ای است اما طول رشته مشخص نشده است. اگر طول رشته مشخص نشود، بطور پـیش فـرض طـول رشته ۲۵۵ در نظر گرفته می شود و این متغیر ۲۵۶ بایت فضا اشغال می کند. متغیر K طول آن ۱۵ تعیین شده است. بنابراین در ایـن متغیر می توان یک رشته با طول ۱۵ را قرار داد. این متغیر ۱۶ بایت فضا اشغال می کند. **همیشه رشته ها در پاسکال یک بایت بیشتر از طول رشته، فضا اشغال می کنند.**

تعداد بایت فضای اشغالی یک رشته= طول رشته +1

K:='ALIREZA';

#7	A	L	I	R	E	Z	A								

نکته: اگر در متغیر K یک رشته با طول بیشتر از ۱۵کاراکتر قرار دهیم هیچ خطایی رخ نمی دهد بلکه اضافه آن در نظـر گرفتـه نمـی شود.

K:='ABCDEFGHIJKLMNPQRSXY';

نکته: نام گذاری متغیرها باید از قوانین نام گذاری شناسه های غیر استاندارد پیروی کند.

شناسه های روبرو مجاز هستند: AB و Z21 و FOR_IF و INT و ...

شناسه های روبرو غیر مجازند: 4B و END و FOR و #A و G* و ...

نکته: **نمی توان دو متغیر هم نام داشت**، پیغام خطا صادر می شود. بطور کلی نمی توان دو شناسه غیراستاندارد هم نام در برنامه داشته باشیم.

```
Program H;
VAR
   A,B,H:Real;
    B,M:Boolean;
```

برنامه روبرو دارای دو خطا است.

نکته: در توربوپاسکال هنگامی که یک متغیر تعریف می شود مقدار اولیه داخل آن اگر عددی باشد صفر است و اگر رشته ایی باشد تهی است.

ثابتــها و تعریف آنها:

از ثابتها معمولا برای بالا بردن خوانایی برنامه و سادگی در ویرایش مقادیر ثابت برنامه استفاده می شود.

ثابتها در زبان پاسکال به دو دسته تقسیم می شوند:

1 - **ثابت بــدون نوع** 2 - **ثابت نــوع دار (با نوع)**

1 - **ثابت بدون نوع**: این نوع ثابت مقدارش در **طول اجرای برنامه نمی تواند تغییر کند**. برای تعریـف ثابتهـا از کلمـه رزرو شـده **Const** استفاده می شود. مثال:

```
Const      F=14;
P='AMIN';
```

تحقیق: برنامه روبرو چند خطا دارد؟

```
Program   MB;
Var
    Z=Byte;
Const
     MB=2.4E-1;
Begin
  Z:=-12;
  MB:=MB+3;
End.
```

2 - **ثابت نــوع دار**: این نوع ثابت را می توان **نوعی متغیر** فرض کرد. یعنی **می توان مقدار آن را در طول اجرای برنامه تغییر داد**. به عبارت دیگر **ثابت نوع دار متغیری است که مقداردهی اولیه می شود**. مثال:

```
Const
  R:integer =12;
  T=25.7;
```

در مثال بالا R یک ثابت نوع دار (از نوع صحیح) است اما T یک ثابت بدون نوع است. مقدار R را می توان در طول اجرای برنامه تغییـر داد اما مقدار T را نمی توان تغییر داد.

مثال :

```
Const
        A=2+3;
        W:Boolean=False;
        H=A*2;
        P:Integer=3;
Var
   T:Shortint;
Begin
      Write(A , H, W,P);
       P:=15;        T:=4;
       P:=P+T;      W:=12>8;
       Write(P);
End.
```

نکته: در زبان پاسکال **; (سیمیکلون)** جدا کنندهٔ دستورات از یکدیگر است.

علامت =: (انتساب)

از این علامت **جهت قرار دادن یک عبارت در یک متغیر** استفاده می شود.

; عبـــارت =: متغیر

نکته: **عبارت باید با نوع متغیر همخوانی داشته باشد**. مثلا اگر عبارت از نوع بولی است باید متغیر نیز از نوع بولین باشد.

نکته: یک عبارت که حاصل آن عدد صحیح می شود را می توان در یک متغیر اعشاری ریخت اما عکس آن صحیح نیست.
A:=25+4; (A می تواند صحیح یا اعشاری باشد).
درصورتی که نوع متغیر با نوع حاصل عبارت یکی نباشد در زمان ترجمه پیغام خطای Type Mismatchصادر می شود.
نکته) از علامت انتساب فقط در قسمت دستورات (بدنه برنامه)استفاده می شود.

انواع عبـــارات در زبان پاسکال:

یک عبارت مجموعه ای از عملگرها و عملوندها است مثل A+2*B یا A/3 > 5

۱- عبارات ریاضی : حاصل این عبارات یک عدد صحیح یا اعشاری خواهد شد. مثل B*8+2 And 12

۲- عبارات رشته ای : رشته ها در زبان پاسکال داخل دو علامت آپستروف (') قرار می گیرند. مثل 'ALI'
حاصل این عبارات نیز یک رشته خواهد شد مثل 'Ali'+'Reza'='AliReza'
در زبان پاسکال طول یک رشته حداکثر ۲۵۵ کاراکتر خواهد بود.

۳- عبارات بولی : حاصل این عبارات می تواند True یا False باشد. مانند (5 > 8) OR (9>4)

عبارات ساده : به عباراتی که در آن تعدادی عملگر از یک نوع به کار رود عبارت ساده می گویند. A*2+4-8/2
عبارات مرکب : عبارتی است که عملگرهای به کار رفته درآن بیش از یک نوع باشد. A+5 > 8 AND 4

جدول تقـدم عملگرها :

( )
NOT ±
*** / DIV MOD AND SHL SHR**
+ - OR XOR
= > < <> >= <=

نکته: تقدم ها از بالا با پایین است.

نکته: عملگرهایی که دارای تقدم یکسانی هستند، عملگری که **سمت چپ تر(در عبارت)** است مقدم تر است.

نکته: پرانتزها تقدم عملگرها را تغییر می دهند.

انـواع عملگرها در زبان پاسکال :

عملگرهای ریاضی : + - * / MOD DIV + -

نکته: حاصل عملگرهای جمع و تفریق(+ و-) و ضرب بستگی به عملوندها دارد.

15.8-2.8=13.0 3*6=18 2.5*2=5.0 12.5+1.5=14.0 12+3=15

نکته: **حاصل عملگر تقسیم(/) همیشه یک عدد اعشاری است.**

20/5=4.0 12/5=2.4

نکته: **عملوندهای مربوط به DIV و MOD باید از نوع عدد صحیح باشند و حاصل آن نیز یک عدد صحیح است.**

15 Mod 3.5 خطا دارد.

24 mod 3 = 0
24 Div 3 = 8
25 mod -3 = 1
-25 mod -3 = -1

25 div -3 = -8
-25 div -3 = 8
7 mod 14 =7
7 div 14 = 0

نکته: **علامت حاصل عملگر Mod به علامت عملوند اول بستگی دارد.**

نکته: Mod و Div کلمات رزرو شده هستند.

حاصل عبارت روبرو چند است؟ 4+35 mod 2 *12 –(6-2) /4

عملگرهای منطقـی: Not AND OR XOR SHL SHR

این عملگرها عملیات های خود را به صورت بیت به بیت انجام می دهند. حاصل این عملگرها یک عدد خواهد بود.

جدول مربوط به عملگرها:

X	Not X
1	0
0	1

X	Y	X Xor Y
0	0	0
0	1	1
1	0	1
1	1	0

دوعملوندهم ارزش، حاصل صفر است.

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

And مانند ضرب است.

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

OR شبیه جمع است.

نکته: این جداول برای **عملگرهای بولی نیز به همین صورت است با این تفاوت که بجای 0 عبارت False و بجای 1 عبارت True** را به کار می بریم.

نکته: حاصل **عملگر Xor** (یای انحصاری) هنگامی False است (صفر است) که هر دو عملوند آن یکسان باشند. یا به عبارت دیگر هنگامی True است (یک است) که دو عملوند آن یکسان **نباشند**.

مثال: 16 AND 27 = 16

$27=(00011011)_2$ $16=(00010000)_2$

00010000
00011011
00010000 = 16

مثال: 19 XOR 35=48

$35=(00100011)_2$ $19=(00010011)_2$

00010011
00100011
00110000 =48

مثال: 14 Xor 25 And Not 21 =6

$14=(00001110)_2$ $25=(00011001)_2$ Not 21 $=(11101010)_2$ $21=(00010101)_2$

14 XOR $(00001000)_2=(00000110)_2$ =6 25 And $(11101010) =(00001000)_2$

نکته: همیشه اعداد را وقتی به **مبنای دو** می برید آنها را **هشت رقمی یا ۱۶ رقمی** کنید.

نکته : اگر یک **عدد فرد را به مبنای دو ببریم اولین رقم آن از سمت راست یک** خواهد بود و اگر یک **عدد زوج** را به مبنای دو ببریم **اولین رقم ان از سمت راست صفر** خواهد بود.

$27=(11011)_2$ $14=(1110)_2$

سئوال: اگر دو عدد فرد را باهم OR کنیم حاصل فرد است یا زوج؟

جواب : چون هر دو عدد فرد بوده اندبنابراین رقم کم ارزش آنها یک بوده است که وقتی باهم OR می شوند حاصل یک خواهد شد پس حاصل فرد است.

سئوال: اگر دوعدد فرد را باهم Xor کنیم حاصل زوج است یا فرد؟

سئوال: اگر یک عدد فرد را NOT کنیم حاصل فرد است یا زوج؟

سئوال: اگر یک عدد فرد و یک عدد زوج را باهم AND کنیم حاصل زوج است یا فرد؟

سئوال: اگر یک عدد فرد را با یک عدد زوج باهم Xor کنیم حاصل زوج است یا فرد؟

مثال: 2*6 OR 12 =

ابتدا * انجام می شود سپس حاصل با 12 ، OR می شود. 12 OR 12 =12

نکته: **هر عددی با خودش OR یا AND شود حاصل خودش می شود**. X Or X =X

X And X = X

به روابط زیر دقت کنید و برای آنها دلیل بیاورید.

X XOR 0 =X

Not (Not (X))=X

X OR NOT X= 255

X AND NOT X = 0

X XOR X =0

$0 \le X \le 255$ == è Not(X) = 255 –X NOT(0)=255-0=255

NOT(1)=255-1=254 Not(120)=255-120=135

مثال: حاصل عبارت روبرو چند است؟ 42 OR 13 AND 20*2

عملگرهای SHL و SHR به معنای شیفت به چپ یا شیفت به راست هستند.

مثال: 14 SHL 1 = ?

$14=(00001110)_2$

14 SHL 1 = $(00011100)_2$=28

مثال: 19 SHR 2=? 19 = $(00010011)_2$

$(00010011)_2$ **è** $(00000100)_2$ = 4

نکته: در ازای صفرها یا یک هایی که بیرون ریخته می شوند در سمت دیگر صفر اضافه می شود.

نکته: هنگامی که **عددی یک بار SHL می شود مانند این است که ضرب در ۲** شده است.

نکته: هنگامی که **عددی یک بار SHR می شود مانند این است که DIV بر ۲** شده است.

```
127 SHR  3 = 127 DIV 2  DIV 2 DIV 2=15
19  SHR  2 = 19 DIV 2 DIV 2 = 4
15  SHL  3 = 15*2*2*2=120
 9  SHL  5 = 9*2*2*2*2*2*2
```

سئوال: حاصل عبارت روبرو چند است؟

115 - (12 SHL 3 AND 8) Mod 15

عملگرهای رابطـه ایی: = =< > <> < >= <=

این عملگرها دارای تقدم یکسانی هستند. حاصل این عملگرها True یا False (بولین) خواهد بود. از این عملگرهـا بیـشتر در شـرطها استفاده می شود.

12 > 14 **è** False 25 <> 25 **è** False FALSE > TRUE **è** False

18 >= 18 **è** True TRUE > FALSE **è** TRUE

عملگرهای بولی: XOR OR AND NOT

این عملگرها شبیه عملگرهای منطقی هستند اما حاصل این عملگرها (بولی) بولین خواهد بود.

اولویت عملگرهای بولی بیشتر از عملگرهای ریاضی و عملگرهای رابطه ایی است.

جدول مربوط به این عملگرها مانند جدول عملگرهای منطقی است با این تفاوت که بجای صفر،False و بجای یک، True قرار دهید.

(12 > 8) XOR (9> 15) = **è** True (13 > 14) OR NOT(10 > 15) == **è** True

(14 <> 19) AND (12 =13) == **è** False

معمولا این عملگرها برای ایجاد شرطهای مرکب با عملگرهای رابطه ایی به کار می روند.

عملگرهای رشته ایی: + (**عملگر ترکیب(الحاق)**)

این عملگر جهت اتصال (ترکیب) دو رشته به کار می رود. 'Ali'+'Reza'='AliReza'

به مثالهای زیر و تفاوت های آنها دقت کنید :

'4'+'5'='45' 4+5=9 و 5+4=9 '5'+'4'='54'

نکته) عملگر ترکیب دارای خاصیت جابجایی نیست.

عملگرهای رابطه ایی را می توان برای رشته ها نیز به کار برد. برای مقایسه دو رشته **حرف اول یک رشته با حرف اول رشته دیگر** مقایسه می شوند(**کد اسکی آنها با هم مقایسه می شوند**).

'ALI' > 'REZA' **è** False 'M' > 'AMPLK' **è** True
'ALI' < 'APLBBN' **è** True 'ABCD' >= 'ABCE' **è** False

خوانایـــی برنامه:

برنامه نویس باید سعی کند برنامه ایی را که می نویسد خوانا باشد تا افراد دیگر هم بتوانند به آسانی برنامه را درک کننـد. بـرای اینکـه برنامه خوانا باشد این موارد را رعایت کنید:

۱- انتخاب نام مناسب برای شناسه های غیر استاندارد

۲- کنگره دار کردن برنامه(تورفتگی)

۳- قراردادن توضیحات در برنامه:

در برنامه پاسکال، برنامه نویس می تواند با قرار دادن توضیحات در برنامه (هر کجا که نیاز به توضیح است) درک برنامـه را آسـان کنـد. برای درج توضیحات می توانید از علامت{} استفاده کنید. ابتدا علامت } را قرار دهید و سپس توضیحات را بنویسید و سپس { را قـرار دهید.می توان از علامت زیر نیز استفاده کرد.

```
Program   Average;                                  (*      توضیحات      *)
Var
  IntA,IntB:Integer;    RlC:Real;
Begin
  Read(IntA,IntB);   { input data}
  RlC:=(IntA+IntB)/2;   { calculate average}
  Writeln(RlC);    { output average }
End.
```

۴- روش نام گذاری متغیرها به صورت مجارستانی: برای اینکه نوع یک متغیر را از روی اسم آن بتوانیم تشخیص دهـیم مـی تـوانیم از این روش برای نام گذاری متغیرها استفاده کنید. در این روش مخفف نوع متغیر در ابتدای نام متغیر آورده می شود. مثلا اگر یـک متغیر از نوع Real با نام F می خواهید تعریف کنید اسم آن را RlF قرار دهید. مثالهای دیگر: BlnZero , IntConter

این روش بوسیله برنامه نویسان شرکت میکروسافت (آقای چارلزسیمونی) به کار گرفته می شود.

۵-نام متغیرها را سعی کنید کوتاه نکنید.

برچسبـــها:

از برچسب ها برای مشخص کردن خطوط خاصی از برنامه استفاده می شود.معمولا از برچسب ها به همراه دسـتور goto اسـتفاده مـی شود. با استفاده از دستور Goto می توان کنترل اجرای دستورات را از خطی به خط دیگر منتقل کرد. همه برچسبها باید در برنامـه در قسمت تعاریف تعریف شوند. برای تعریف برچسبها از کلمهٔ رزروشدهٔ Label استفاده می شود.برای نام گذاری برچسب ها از قوانین نام گذاری شناسه های غیراستاندارد استفاده می کنیم. البته یک استثناء وجود دارد و آن این است که برای نـام گـذاری برچـسب هـا مـی توان از اعداد صحیح بین 0 تا 9999 استفاده کرد.

```
Label
     L1,5;
Var
  K:integer;
Begin
      K:=0;
 5:    K:=K+2;
 L1:  Writeln(K);
      K:=K+1;
       If  K<100 then
           Goto L1;
       If  K<150  then
           Goto 5;
End.
```

برای مشخص کردن برچسب یک خط ابتدا نام برچسب را می نویسیم و سپس علامت: را قرار می دهیم و دستور را می نویسیم.

نکته: نام برچسب نمی تواند 5A باشد زیرا حرف اول آن عدد است.

استفاده از دستور Goto در برنامه به هیچ عنوان پیشنهاد نمی شـود زیـرا درک منطـق برنامه و دنبال کردن مسیر اجرای آن به منظور عیب یابی برنامه مشکل است و خوانـایی برنامه کم می شود.

نکته: شناسه های غیراستانداردی که کاربر تعریف می کند نباید هم نام باشند. بـه عنـوان مثال کاربر نمی تواند یک برچسب با نام A تعریف کند و یک متغیر نیز با نام A تعریـف کند. در این صورت پیغام خطا صادر می شود.

تست

۱-کدامیک از اسامی زیر برای یک شناسه غیر مجاز است؟

الف) for_if ب) prog ج) A6B د) if

۲-طول نام متغیرها در زبان پاسکال می تواند حداکثر ..... کاراکتر باشد.

الف) ۶۴ ب) ۲۵۵ ج) ۶۳ د) ۸

۳- متغیر نوع shortint چند بایت فضا اشغال می کند؟

الف) ۲ بایت ب) ۱ بایت ج) ۴ بایت د) حداقل ۲ بایت

۴-کوچکترین واحد معنی دار در زبان پاسکال ..... است.

الف) کاراکتر ب) شناسه ج) توکن د) برچسب

۵-معمولا برچسب ها به همراه کدام دستور استفاده می شوند؟

الف) Writeln ب) For ج)Readln د)Goto

۶-انواع اعداد صحیح در زبان پاسکال کدامند؟

الف) Longint ,Word, Shortint,Integr,Byte ب) Longint,Integer,Single,Byte

ج)Integer,Single,Comp,Byte د) Integer,Byte,Extended,Word

۷-عباراتی که درون ..... قرار می گیرند دارای حق تقدم بالاتری نسبت به دیگر عملگرها دارند.

الف)آپستروف ب) پرانتز ج) کروشه {} د) علامت [ ]

۸-کدام یک از شناسه های زیر در پاسکال مجاز نیست؟

الف) ASCII ب)Case ج)Pascal د) INT

۹-حاصل عبارت **12 + 6 / 3 + 20 MOD 6** کدام است؟

الف) 8 ب) 3.6 ج) 16 د) 13.2

۱۰-حاصل عبارت **12 XOR 25 SHL 2 AND 14** کدام است؟

الف) ۱۰ ب)۸ ج) ۴ د)۱۲

۱۱-ثابت نوع دار ثابتی است که:

الف)در طول اجرای برنامه مقدار آن را می توان تغییر داد. ب) قابل چاپ نمی باشد

ج)مقدار اولیه آن را نمی توان تغییر داد. د) در هنگام تعریف مقدار آن مشخص نیست.

۱۲- عملوندهای عملگر DIV از چه نوعی هستند؟

الف)صحیح و اعشاری ب) اعشاری ج) صحیح د) رشته ای

۱۳-عبارت **(a) XOR 0 b OR**) معادل کدام گزینه زیر است؟(a وb عدد صحیح هستند)

الف) a XOR b ب) b OR a ج) b AND a د) b AND NOT(a)

۱۴- کدام شناسه در زبان پاسکال مجاز است؟

الف) Ali R ب) then ج) MEHRDAD7 د) 5Array

۱۵-کدام گزینه زیر درست است؟

الف) عبارتهای بولی یک نتیجهٔ رشته ای دارند. ب)عبارتهای بولی یک عدد اعشاری بر می گردانند.

ج)عبارتهای بولی یک عدد صحیح بر می گردانند. د) عبارتهای بولی ارزش درست یا نادرست دارند.

۱۶- اگر متغیرهای x, y از نوع صحیح باشند در این صورت حاصل عبارتهای X*Y و X/Y به ترتیب چه نوعی خواهد بود.(از راست به چپ)

الف)اعشاری – صحیح ب)اعشاری – اعشاری ج) صحیح – صحیح د) صحیح – اعشاری

۱۷-کدام گزینه صحیح است؟

الف) Const name := 5; ب) Const name = 5; ج) Const name = 5 د) Const name := 5

۱۸-خروجی دستور (170+60) AND NOT $18 XOR 13 کدام است؟

الف) 29 ب) 17 ج) 21 د) 25

۱۹-کدام یک از اسامی زیر برای یک شناسه غیر مجاز است؟

الف) INC ب) Write ج) const د) هر سه مورد

۲۰-در یک برنامه پاسکال هر سطر حداکثر چند کاراکتر می تواند باشد؟

الف)۶۳ ب) ۱۲۷ ج) ۲۵۵ د)محدودیتی ندارد

۲۱-کدامیک از شناسه های زیر در پاسکال مجاز نیست؟

الف)ABS ب) DOWNTO ج)A3B د)INT

۲۲-حاصل عبارت (12*5 DIV 3+8 MOD 6) کدام است؟

الف)۱۴ ب)۲۲ ج)۰ د)۱۲

۲۳-با فرض اینکه **a=10** و **b=8** باشد حاصل عبارت زیر چیست؟ **a DIV –3 +16/b*3**

الف)۳ ب) ۰ ج) ۳- د)عبارت فوق نادرست است.

۲۴-خروجی دستور **WRITE(10 XOR (NOT 255))** چیست؟

الف) ۱۰ ب) مکمل دو عدد۱۰ ج) ۵ د) ۱۱

۲۵-حاصل 167 shr 3 کدامست؟

الف) 20 ب)2 ج)1336 د)7

۲۶-اگر a و b از نوع منطقی باشند معادل a=b کدامست؟

الف) a XOR b ب) NOT (a XOR b) ج) a OR b د) a AND b

۲۷-در رابطه با ترتیب عملگرها کدام گزینه درست است(ازراست به چپ)

الف) *, Not, mod ب) Not, >, mod ج) *, +, and د) Not, div, or

۲۸-Real دارای دقت چند رقم اعشار است؟

الف) ۱۷ ب) ۱۰ ج) ۱۱ د) ۹

۲۹-کدام یک از انواع داده های زیر شامل مقادیر منفی نیز می باشد؟

الف)Byte ب)Shortint ج)Char د)Word

۳۰-حاصل عبارت 2 or 3*2+120 mod 4 چیست؟

الف) ۲ ب) ۶ ج) ۳۶ د)۴

۳۱-کدام شناسه در زبان پاسکال صحیح است؟

الف) 2a ب) X-Y ج) TO د)SQR

۳۲-حاصل عبارت **1+ 13 22 Xor = > 13 AND 22 OR 24** کدام است؟

الف) True ب) 23 ج)False د)عبارت اشتباه است.

۳۳-در پاسکال، بلاک اصلی برنامه بلاکی است که :

الف) با توجه به لزوم در ابتدا یا انتها اجرا می شود. ب) در انتها اجرا می شود.

ج) در ابتدای برنامه قرار دارد. د) در انتهای برنامه قرار دارد.

۳۴-داده نوع Comp از چه نوعیست؟

الف) اعشاری Double ب) اعشاری Single ج) صحیح بلند د) صحیح

۳۵-چنانچه متغیرهای زیر با روش مجارستانی نامگذاری شده باشند مجموع فضای اشغالی آنها کدامست؟

CmpNumber , LngCount ,XtdNo

الف)18 ب)22 ج) 20 د)24

۳۶- حاصل عبارت روبرو کدام است؟ **17 + 6 shl 2 xor 4 * 7**

الف)64 ب) 53 ج)21 د)315

۳۷-علامت ..... نشانه پایان سطر در پاسکال است.

الف) . ب) : ج) , د) ;

۳۸-اگر a و b از نوع بولی باشند، معادل عبارت (a≤b)>b کدام است؟

الف)a and not (b) ب)a or b ج)not(a) and b د) Not(a or b)

۳۹-در چه صورت در عبارت روبرو نتیجه عمل نامعلوم خواهد بود؟ **X mod Y**

الف) Y=0 ب)X=0 ج)X>Y د)X<Y

۴۰-عبارت معادل با a≤(a≥b) اگر a و b از نوع منطقی باشند کدامست؟

الف)A XOR B ب)A and B ج)A OR B د)True

۴۱-خلاصه عبارت B and (B Or False) کدام است؟

الف) True ب)False ج)B OR False د)B

۴۲-کدام گزینه درست و از نوع صحیح است؟

الف) A mod B * Div C ب) A (Div B Mod C)

ج)A/B +C Div D د) A Div B Mod C

۴۳-کدام گزینه برای تعریف یک ثابت درست است؟

الف) Const a=12.5:real; ب) Const a=12.5-9.5;

ج) Const a,b=12; د)Const a:Real;

۴۴-وجود کدام گزینه در ساختار برنامه به زبان توربوپاسکال برای اجرا ضروری است؟

الف) دستورات (Statement) ب)عنوان (Header) ج) ثابتها د) متغیرها

۴۵-در پاسکال استاندارد ترتیب تعریف و یا اعلان شناسه ها کدام است؟

الف) نوع داده ها – متغیرها – ثابتها ب) متغیرها – نوع داده ها – ثابتها

ج) ثابتها – نوع داده ها – متغیرها د) ثابتها – متغیرها – نوع داده ها

۴۶-کدام گزینه از بخشهای اصلی و اجباری در برنامه پاسکال استاندارد است؟

الف)علائم و ثابتها ب) تعاریف و اعلانات ج) شناسه ها د) عنوان

۴۷-کدام گزینه در مورد علامت ; (semicolon) در زبان پاسکال صحیح است؟

الف)به عنوان پایان دستورها مورد استفاده قرار می گیرد
ب) برای جداسازی دستورها است
ج) اولین و آخرین دستور نیاز به ; دارند.
د) آخرین دستور نیاز به ; دارد.

۴۸-کدام گزینه صحیح است؟

الف) IntA:=IntB Mod RlD
ب) IntC:= BytF Mod ShtA
ج) RlC=IntA Div BlD
د) IntA:= RlC Div ShtD

۴۹-خروجی دستور مقابل چیست؟ Write('BABY'<'BABAK')

الف)1
ب)True
ج)False
د)خطا تولید می شود.

۵۰-تابع روبرو معادل کدام گزینه است؟ (A≤B) AND A

الف) A AND B
ب) A OR B
ج) A XOR B
د) Not A AND B

۵۱-کدام کلمه رزرو شده در پاسکال است؟

الف)Integer
ب)Var
ج)true
د)Read

۵۲-اگر تعریف روبرو مورد نظر باشد کدام گزینه باعث خطا می شود؟

```
Var a:integer;
B :Real;
```

الف) B:=a*B
ب) B:=a Mod B
ج) B:=a-B
د) B:=a/B

تحقیق: آیا رابطه روبرو همیشه صحیح است؟ (A SHR 1) SHL 1 = A

دستورات ورودی / خروجـــی :

در زبان پاسکال دستورات Read و Readln برای دریافت داده ها از ورودی در برنامه به کار می روند.ایـن دو دسـتور بـا هـم در یـک مورد اختلاف دارند که بحث خواهد شد و بقیه موارد و نکات آنها شبیه هم است.

READ(متغیر); یا READLN(متغیر);

نکته) داده هایی که از صفحه کلید وارد می شود باید با نوع متغیرذکر شده در دستور ورودی مطابقت داشته باشد. به عنـوان مثـال اگـر عدد 12.9 از صفحه کلید وارد می شود متغیر موجود در دستور ورودی هم باید اعشاری باشد. در غیر این صورت پیغام خطا صادر مـی شود و برنامه قطع می شود.

نکته) به خطاهایی که در حین اجرای برنامه رخ می دهد خطای زمان اجرا می گویند. Run Time Error

مثال: اگر در ازای دستور Read(A,B,C); (متغیر A از نوع صحیح و متغیر B از نوع کاراکتری و متغیر C از نوع رشـته ایـی اسـت) داده های 231 5 12 وارد شود آنگاه عدد ۱۲ وارد متغیر A و کاراکتر 5 وارد متغیر B و رشته '231' در متغیر C قرار می گیرد.

نکته) هنگام وارد کردن داده ها نمی توانید یک عبارت بنویسید. مثال در ازای دستور Readln(A) اگر A از نوع اعـشاری باشـد نمـی توانید 12+3 بنویسید اما می توان 12.3E-1 را وارد کرد. (نمایش اعداد به صورت علمی)

```
Const  H=12;
  Z:Real=18.2;
.
.
.
Read(H);  خطا
Read(Z);
```

نکته) در دستورات ورودی نمی توانید عبارت بنویسید. Read(A+1) خطا دارد.

نکته)از ثابت بدون نوع نیز نمی توانید در دستورات ورودی استفاده کنید. اما ثابت بانوع خطایی ندارد.

نکته)اگر دستور Readln را بدون ذکر متغیر در برنامه بنویسیم باعث می شود برنامه موقتا متوقف می شود تا کاربر کلیدEnter را بزند تا اجرای برنامه ادامه پیدا کند.

Readln;

تحقیق: اگر متغیر S از نوع رشته ایی باشد و در ازای دستور Read(S); عبارت 2*25 را وارد کنیم چه اتفاقی می افتد؟

نکته) اگر داده های ورودی کمتر از متغیرهای ذکر شده در دستور read یا readln باشد رایانه برای دریافت بقیه داده هـا منتظـر مـی ماند. Read(a,b,c,d); == **è** 12 13.5

نکته) اگر داده های وارد شده بیشتر از تعداد متغیرهای ذکرشده در دستور ورودی Readln بیشتر باشد داده های اضـافی حـذف مـی شوند.

Readln(a,b); === **è** 17 13 25 19 داده های ۱۷ و ۱۳ در a و b قرار می گیردو بقیه حذف می شوند.

نکته)اگر داده های وارد شده از تعداد متغیرهای ذکرشده در دستور ورودی Read بیشتر باشد داده های اضافی حذف نمی شوند بلکـه در حافظه می مانند تا در دستورات Read یا Readln بعدی استفاده شوند.

مثال:

```
Read(a);  == è 45  32  21  70 30
…..
Readln(K,M);
…..
Read(Z);
```

```
Read(a,b); = è 12  15  17  18
……
Read(C);
……
Readln(H,F);
```

نکته) از متغیر نوع Boolean نمی توان در دستورات ورودی استفاده کرد. مثلا اگر متغیر B از نوع Boolean باشد دستور Read(B); خطا است.

برای چاپ مقدار متغیرها یا عبارت یا ثابتها می توان از دستورات Write یا Writeln استفاده کرد.

تفاوت دستور Write و Writeln در این است که در دستور Writeln پس از چاپ مقادیر در مانیتور مکان نما به ابتدای سطر بعـدی منتقل می شود. اما در دستور Write مکان نما در همان سطر باقی می ماند.درواقع می تـوان گفـت کـه در دسـتور Writeln پـس از چاپ مقادیر بر روی مانیتور دو کاراکتر CR و LF را به مانیتور می فرستد. Writeln; == Write(#10#13);

CR با کد اسکی ۱۳ باعث می شود مکان نما (چراغ چشمک زن) به ابتدای سطر فعلی بیاید و LF با کد اسـکی ۱۰ باعـث مـی شـود مکان نما یک سطر پائین بیاید.

مثال) دستور;Writeln(A,'ABC',5*2+3) مقدار موجود در متغیر A را چاپ می کند و به دنبال آن رشته ABC و عدد ۱۳ را چاپ می کند و یک سطر پائین می رود.

نکته) دستور ;Writeln باعث می شود مکان نما یک سطر پایین برود.

نکته) دستور ;(Write(False<True باعث می شود که عبارت TRUE بر روی مانیتور ظاهر شود. بنابراین متغیرهای از نوع بولین را می توان در دستورخروجی قرار داد.

نکته)هنگامی که اعداد اعشاری را بوسیله دستور Write یا Writeln در خروجی چاپ می کنیم اعداد به صورت علمی چاپ می شوند. در ادامه جهت چاپ آنها به صورت ممیز ثابت بحث می شود.

تحقیق) دستور ;Writeln(#7) و ;(Write(#13#10 چه چیز را در خروجی چاپ می کنند؟

نکته) دستور ;(Write('''') باعث می شود که عبارت ' برروی مانیتور چاپ شود. دستور ;(Writeln(''',,''' چه چیزی را چاپ می کند؟ دستور;(Write(''' چه چیزی را چاپ می کند؟

فرمت بندی خروجی:

شما می توانید داده های را که قرار است در مانیتور چاپ کنید با یک فرمت خاصی چاپ کنید.

مثال: دستور ;Writeln('MAHDI') عبارت MAHDI را در ابتدای سطر چاپ می کند. اما دستور ;Writeln('MAHDI':8) عبارت MAHDIآآآ را چاپ می کند. (آ نشانه جا خالی است).عدد ۸ نشانه یک باند(میدان) هشتایی است که عبارت MAHDI از سمت *راست* آن پر می شود و جا های خالی آن باقی می ماند.

			M	A	H	D	I

مثال) ;Write(124:7) عدد ۱۲۴ را دریک باند ۷ تایی چاپ می کند که در نهایت ۴ جا خالی در سمت چپ عدد می ماند.

اعداد صحیح و رشته ها دارای یک باند برای فرمت بندی هستند.

نکته) اگر باند تعریف شده از تعداد کاراکتری که چاپ می شود کمتر باشد باند صرف نظر می شود.

مثال)دستور ;Write(1258:3) عدد ۱۲۵۸ را بدون باند چاپ می کند.(خطایی رخ نمی دهد).

برای اعداد اعشاری نیز می توان پهنای باند تعریف کرد. که در این صورت اعداد اعشاری در خروجی به صورت ممیز ثابت نمایش داده می شوند. اعداد اعشاری دارای دو پهنای باند هستند .

مثال) ;Writeln(126.2314**:10:2**) که باند اولی (عدد ۱۰) مربوط به کل عدد است و باند دومی مربوط به تعداد اعشار آن است. دستور ذکر شده عدد 126.23 را به صورت زیر چاپ می کند.

				1	2	6	.	2	3

نکته) هنگامی که اعشار اضافه دور ریخته می شود عدد در اعشار باقی مانده گرد می شود.

مثال) ;Writeln(125.4751:7:2) در این دستور عبارت 125.48 چاپ می شود.

نکته) در صورتی که در باند اولی تعداد رقم ها از تعداد باند بیشتر باشد باند مربوط به کل عدد در نظر گرفته نمی شود.

Writeln(54.456:3:5); == **è** 54.45600 Write(19.96:2:1); == **è** 20.0

Writeln(19.152:0:1); == **è** 19.2 Write(17.184:5:0); == **è** 17

Write(19.512:1:0); == **è** 20 Writeln(3.451:6:1); == **è** 3.5

تحقیق: دستور ;Write(True) چه چیز را در خروجی چاپ می کند؟

تست: خروجی قطعه برنامه روبرو کدام است:

```
Y:=2.36;
Writeln(Y:4:1);
```

الف) 2.3 ب)2.30 ج)2.36 د)2.4

تست: اگر متغیر Pi از نوع Real باشد خروجی دستور Write کدام است؟

```
Pi:=3.1416;
Write(Pi:0:3);
```

الف)3.142 ب)3.141 ج)3 د)0

نکته) ;(Write(#13#10 معادل دستور ;Writeln است.

دستورات در زبان پاسکال:

در زبان پاسکال بین هر دستور برای جدا کردن دستورات از یکدیگر علامت ; قرار می گیرد. بنابراین این فکر که در انتهای هر خط باید یک علامت ; قرار داد اشتباه است. آخرین دستورالعمل می تواند ; نداشته باشد. کلمات Begin و End دستوالعمل نیستند و تنها برای مشخص کردن شروع و پایان بلاک به کار می روند. بنابراین بعد از Begin و قبل از End از علامت ; استفاده نمی شود.

دستورات در زبان پاسکال به دو دسته تقسیم می شوند:

- دستورات ساده: ۱- دستور انتساب ۲- دستور پرشGoto ۳- دستور فراخوانی پردازه(زیرروال)
- دستور ساختاری:
  - دستور مرکب: دستور مرکب، از تعدادی دستور دیگر که بین کلمات ذخیره شدهٔ Begin و End قرار گرفته اند تشکیل شده است.

به تورفتگی دستورات دقت کنید.(کنگره دار کردن)
دستور قبل از End می تواند ; نداشته باشد.
چون دستور مرکب به عنوان یک دستور درنظر گرفته می شود باید از دستورات دیگر جدا شود به همین دلیل بعد از End علامت ; قرار می گیرد.

```
Begin
   A:=A+1;
   Writeln(A);
   A:=B*2
End;
```

- دستورات شرطی: در ادامه بحث می شود.
- دستورات تکرار: در ادامه بحث می شود.
- دستورWith: که در پاسکال پیشرفته بحث می شود.

دستورات شــرطی:

دستورات شرطی دستوراتی هستند که برنامه نویس برای اجرای یک یا چند دستور دیگر شرط قرار می دهد. دستورات شرطی به دو دسته تقسیم می شوند **۱- دستور If-Then ۲- دستور Case**

1 - دستور If-Then :

دستور If خود به دو شکل است ۱- If –then و ۲- If then Else

شکل کلی دستور If then:

If شرط(ها) then
; یک دستور ساده یا یک دستور ساختاری

در شکل کلی دستور روبرو دستور ساده یا دستور ساختاری درصورتی انجام می شود که شرط(ها) True باشد.

```
If   A>10   then
     Write('Ali');
```

مثال: در مثال روبرو درصورتی رشتهٔ Ali چاپ می شود که A>10 باشد.

نکته) فقط یک دستور بعد از then مربوط به IF است و این دستورمیتواند ساده یا مرکب یا ساختاری باشد.

```
IF  (Z<>0)  Xor  (H<=5)  then
  Begin      دستور مرکب
          Writeln('*');
           A:=A+4;
           Read(F);
  End;
```

```
IF   S<>'ALI'   then
     Write('*');
Write('M');
```

نکته) دستور If با بلاک مربوط به آن کلا یک دستور محسوب می شود.

نکته) دقت کنید که بعد از Then علامت **;** نباید باشد. در صورتی که علامت **;** قرار گیرد هیچ خطایی رخ نمی دهد بلکه علامت **;** به عنوان تک دستور If در نظر گرفته خواهد شد.(خطای منطقی)

```
IF   B>3 then ;
  Write('*');
Write('Z');
IF  NOT D<4 then  Writeln('S');
```

مثال) در مثال روبرو دستور ;('*')Write مربوط به IF نیست.
زیرا علامت **;** بلاک IF محسوب می شود. به همین دلیل دستور ;('*')Write حتما یک بار اجرا می شود. دستور ;('Z')Write نیز یک بار اجرا می شود و مربوط به IF نیست. اما دستور ;('S')Writeln مربوط به IF دوم است و درصورتی انجام می شود که NOT D<4 درست باشد.(D<4 نباشد)

```
Label   AB;
Var
    H:Shortint;
Begin
     H:=0;
AB: Writeln(H);
     H:=H+2;
     IF  H<100 then
          Goto  AB;
End.
```

تحقیق) برنامه روبرو چه خروجی دارد؟
الف) اعداد زوج از 0 تا 101 ب) اعداد زوج دو رقمی
ج) اعداد زوج کوچکتر از 100 و بزرگتر مساوی 0
د) اعداد فرد کوچکتر از 100

نکته) در شرط دستور IF می توان یک عبارت بولی قرار داد تا شرطها باهم ترکیب شوند.
نکته) کلمات IF و THEN و ELSE کلمات رزرو شده (ذخیره شده)هستند.

2 - دستور IF-THEN ELSE:

شکل کلی دستور IF Then ELSE :

مثال:

```
IF   A>8   then
      Writeln('ALI')
 Else
      Writeln('REZA');
F:=F+1;
```

```
IF  شرط(ها)  THEN
        یک دستور ساده یا ساختاری
ELSE
    ; یک دستور ساده یا ساختاری
```

در مثال بالا در صورتی که A>8 درست باشد دستور ('ALI')Writeln انجام می شود و در غیر اینصورت دستور ('REZA');Writeln انجام می شود. توجه داشته باشید که از این دو دستور فقط از آنها انجام خواهد شد. دستور ;F:=F+1 مربوط به دستور IF نیست و بعد از اجرای دستور IF then Else انجام خواهد شد.

```
IF  (M=10) OR (H<=20) then
 Begin
     Writeln('*');
      Z:=12;
 End
Else
    Writeln('AMIN');
```

مثال: در مثال روبرو دستور مرکب بلاک IF است و دستور;('AMIN')Writeln مربوط به بلاک Else است.
نکته) دستور قبل از Else علامت **;** نمی خواهد در صورت استفاده از علامت **;** قبل از Else پیغام خطا در زمان ترجمه رخ می دهد. چرا؟

تحقیق) فلوچارت مربوط به دستورهای IF –Then و IF then Else را رسم کنید.

```
Program  ODD_EVEN;
Var
   K:integer;
Begin
     READLN(K);
     If   (K  MOD  2) =0  then
           Wrietln('EVEN')
       Else
           Writeln('ODD');
End.
```

مثال)برنامه ای بنویسید که یک عدد از ورودی دریافت کند و زوج و فردبودن آن را مشخص کند؟

نکته) کلمهٔ Else بخشی از دستور IF است و به تنهایی به کار نمی رود.

IF های تو در تو(متداخل) :

همانطور در شکل کلی دستور IF مشخص است تک دستور مربوط به IF می تواند یک دستور ساختاری دیگر از جملـه یـک IF دیگـر باشد.

```
IF  B then          { B:Boolean}
  IF  C>15  then
      Writeln('ALI');
```

در مثال روبرو IF دومی بلاک IF اولی است. درصورتی که متغیر بولینB TRUE باشد IF دوم بررسی می شود که در صورت درست بودن C>15 رشتهٔ ALI چاپ می شود.

در مثال بالا می توان هر دو IF را دریک IF ادغام کرد.
به عبارت دیگر دستور روبرو و مثال بالا یک کار انجام می دهند و معادلند.

```
IF  (B) AND  ( C > 15)  then
    WRITELN('ALI');
```

مثال :

باید مشخص کنید که هر Else مربوط به کدام IF است.

```
IF  K>10   then
    IF  H=5   then
          Writeln('$')
    Else
          Writeln('%')
 Else
    Writeln('*');
```

(K>10) = False **è** '*'
(K>10) = True و (H=5)=True **è** '$'
(K>10)=True و (H=5)=False **è** '%'

مثال) مشخص کنید که در چه صورت چه چیزی چاپ می شود؟

```
IF  W<>12  then
    IF   P<4  then
      Begin
        IF  A    then
              Write('#');
       End
       Else
           Write('+')
   Else
       Write('~');
```

IF های تو در تو موقعی استفاده می شود که از بین چند حالت یک حالت را بخواهیم طبق شرط ها انتخاب کنیم.IF هـای تـو در تـو خوانایی برنامه را کاهش می دهد. به همین دلیل توصیه می شود بجای IF های تو در تو از دستور Case استفاده شود.

دستــور Case :

دستور Case زمانی به کار می رود که بخواهیم از بین چندین حالت، یک حالت را انتخاب کنیم و معادلIF های تو در تو است.
شکل کلی دستور Case :

```
Case  عبارت یا متغیر  OF
  ; دستور ساده یا ساختاری۱ :مقدار یا مقادیر۱
  ; دستور ساده یا ساختاری۲ :مقدار یا مقادیر۲
  ; دستور ساده یا ساختاری۳ :مقدار یا مقادیر۳
  …..
  Else
      ; دستور ساده یا ساختاری
End ;
```

مثال:

```
      A:=12;
Case A+3  OF
  1,12:   Writeln('OK');
  15,14:Writeln('ALI');
  16:Writeln('AMIN');
End;
```

روند اجرایی دستور Case به این شکل است که عبارت یا متغیر جلوی Case مقدارش مشخص می شود و سپس این مقدار با هرکـدام از مقادیر مقایسه می شود با هر کدام که برابر بود دستور مربوط به آن اجرا می شود. اگر مقدار با هیچ کدام یک از مقـادیر برابـر نـشود دستور مربوط به Else (در صورت وجود) انجام می شود.

```
Readln(F);
Case  F   OF
  0..9: Writeln('ONE');
  10 .. 99: Writeln('TWO');
  ELSE
    Writeln('Other');
END;
```

نکته) قسمت Else در دستور Case اختیاری است.

نکته) دستور Case عبارت Begin ندارد اما End; دارد.

نکته) اگر بجای دستور ;('Writeln('ONE بخواهیم چندین دستور بنویسیم باید از دستور مرکب استفاده کنیم.

نکته) در قسمت مقادیر نباید از متغیر یا ثابت با نوع استفاده کنید.

نکته) عبارت روبروی Case باید از نوع ***اسکالر*** باشد. در غیر اینصورت در زمان ترجمه خطا صادر می شود.

نوع داده اسکالـر:

نوع داده ایی که قابل شمارش باشد را اسکالر می گویند. داده های اسکالر ماننـد : اعـداد صـحیح (بـه غیـر از Comp) – کـاراکتری– بولین

کاراکترها قابل شمارش هستند یعنی بعد از حرف 'A' حرف 'B' است و قبل از حرف'A' نیزکاراکتر '@' است. (جدول کد اسکی)

اعداد اعشاری و رشته ها اسکالر نیستند.

نکته) اگر در دستور Case عبارت Else نداشته باشیم ممکن است هیچکدام از حالت ها انتخاب نشود.

نکته) مقادیر در دستور Case می توانند تکراری باشند اما در هنگام بررسی مقادیر اولین حالت که انتخاب شود اجرا مـی شـود و بقیـه حالت ها در نظر گرفته نمی شوند.

تحقیق) برنامه روبرو چند خطا داد؟(برنامه در پاسکال استاندارد است)

```
Program  H;
Var
    H:Integer;
Const
    X:20;
Begin
  Readln(H);  X:=X+1;
  Case  H/10  OF
      1,3:
          Writeln(X);
      2,1:Writeln('F');
          H:=12;
      9..5:Writeln('X');
  End;
End;
```

تست :

۵۳-کدام گزینه در مورد متغیر A که از نوع Boolean می باشد نادرست است؟

الف) WRITE(A) ب) A:=TRUE; ج) A:=False; د) READ(A)

۵۴- اگر **A=68.951** باشد حاصل اجرای دستور **WRITE(A:5:1)** کدام است؟

الف) **68. 9** ب) **68. 96** ج) **69. 0** د) **69. 1**

۵۵- خروجی برنامه روبرو کدام است؟

```
Var  D:real;
Begin
  D:=6 XOR 0;
  Case D of
    1. . 7: writeln('BAD');
    8. . 12 : Writeln('NOT BAD');
   13. . 20 : writeln('OK');
  end;
end.
```

الف) BAD ب) Not BAD ج) OK

د) در زمان ترجمه پیغام خطا می دهد.

۵۶-کدام یک از انواع داده های زیر فضای بیشتری اشغال می کند؟

الف) Longint ب) Word

ج) Real د) Single

۵۷- چنانچه بخواهیم از چندین حالت(بیش از دو حالت) یک حالت را انتخاب کنیم کدام ساختار مناسبتر است؟

الف) IF ب) Case ج) Goto د) For

۵۸- در صورتی که از دستور ورودی Readln بدون ذکرمتغیرهای ورودی استفاده نمایید باعث خواهد شدکه کامپیوترمنتظر فشردن کلید ..... شود؟

الف) ALT ب) Ctrl ج) Esc د) Enter

۵۹- حاصل عملگرهای رابطه ایی از نوع ..... و حاصل عملگر / (تقسیم) از نوع ..... است.

الف) ریاضی - اعشاری ب) بولی – صحیح ج) بولی- اعشاری د) ریاضی – بستگی به عملوند ها دارد

۶۰- کدام گزینه درست می باشد؟

الف) X mod Y=X-(X div Y)*Y ج) X mod Y = X*(X div Y)-Y

ب) X mod Y=X-(X div Y)-Y د) X mod Y=X*(X div Y)+Y

۶۱- حاصل اجرای دستور مقابل چیست؟

```
Var  Z: string;
Begin
     Readln(Z);
     Case  Z  of
          1: Write(1);
          2: Write(2);
     end;
end.
```

الف) چاپ اعداد ۱ یا ۲ ج) خطا

ب) چاپ عدد دریافتی Z د) چاپ اعداد ۱ و ۲

۶۲ - اگر A:=329 باشد حاصل A SHR 4 کدام است؟

الف) ۲۰ ب) ۲ ج) ۲۱ د) ۲۹

۶۳- کدام گزینه در مورد نتیجه حاصل از اجرای دستور Write(#13); صحیح است؟

الف) مکان نما یک سطر پایین می رود ب) صدای بوق از بلندگوی سیستم شنیده می شود

ج) مکان نما به ابتدای سطر فعلی می رود د) مکان نما به ابتدای سطر بعدی می رود.

۶۴- استفاده از کدام نوع داده در عبارت دستور Case مجاز نمی باشد؟

الف) Longint ب) Byte ج) Real د) Shortint

۶۵- حاصل عبارت روبرو چند است؟ ((43 shr 1) shl 2) xor 84

الف) ۲۴۳ ب) ۶۳ ج) ۲۳ د) ۰

۶۶- نتیجه کدام عبارت بولی برابر True است؟

الف) 'TEST'>'test' ب) 'TEST'<'test' ج) 'TEST'>='test' د) 'TEST'='test'

۶۷- عبارت جبری $3<X\leq 5$ معادل کدام عبارت بولی در پاسکال است؟

الف) (X<=5) AND (X>3) ب) (X<=5) OR (X>3)

ج) (X>3) AND (X>=5) د) (X>3) OR (X>=5)

۶۸- خروجی برنامه زیر کدام است؟

```
Var  x,y:byte;
    A,b,f:boolean;
Begin
   X:=5;  y:=6;
   A:=X<=y;
    B:=Y<X;
  If  a<b  then   f:=a<b
    Else   f:=b<a;
 Write(F);
End.
```

الف) False ب) خطای Type Mismatch ظاهر می شود

ب) True (به مقدار x و y بستگی دارد.) ج) True (به مقدار x وy بستگی ندارد)

۶۹- علامت : همواره در ..... بکار می رود؟

الف)مقدار دهی متغیرها ب) تعیین نوع متغیرها

ج)معرفی ثابتها د) پایان سطرها

۷۰- اگر متغیر رشته ایی به طول ۱۰ تعریف کرده باشید حافظه اختصاص داده شده به آن چند بایت است؟

الف) ۱۱ ب)۹ ج)۱۰ د) بستگی به رشته داخل آن دارد

۷۱- کدام نوع عبارت را می توان در جلوی دستور Case در پاسکال نوشت؟

الف)Byte ب) Double ج) String د)Real

۷۲- اگر X:=3456.981 باشد حاصل اجرای دستور Write(X:0:1) چیست؟

الف)3456.0 ب)3456.9 ج)3457.0 د)3457

۷۳- اگر دستورات داده شده به ترتیب اجرا شوند و داده های ورودی به صورت زیر باشند مقدار متغیرهای P1,P2,P3 کدام است؟

```
Readln(P1,P2);  1  2  3
Read(P3);  4  5  6
```

الف) 1 , 2 , 3 ب) 1 , 2 , 4

ج) 1 , 2 , 5 د) 1 , 4 , 5

۷۴- اگر مقدار J برابر ۳ باشد با توجه به دستورات روبرو خروجی کدام است؟

```
IF  J>4  then ;
    J:=2*J;
IF   J<=3  then
  J:=3*J;
Writeln(J);
```

الف) 3 ب) 6 ج) 18 د)9

دستورات تکرار (حلقه):

دستورات تکرار برای تکرار اجرای تعدادی دستور به کار می روند. برای ایجاد حلقه ها از دستورات تکـرار اسـتفاده مـی شـود. سـه نـوع دستور تکرار در زبان پاسکال وجود دارد:

1 - دستور While Do 2 - دستور Repeat Until 3 - دستور For

هر کدام از این دستورات دارای ویژگی های خاصی هستند که برنامه نویس با توجه به نیاز خود یکی از آنها را انتخاب خواهد کرد.

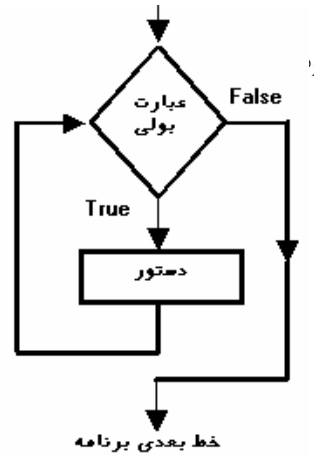
1 - دستور While Do :

شکل کلی دستور:

While عبارت بولی Do

یک دستور ساده یا دستور ساختاری ;

در هنگام اجرای دستور While ابتدا عبارت بولی(شرط) محاسبه می شود در صـورتی کـه نتیجـه برابـر **True** باشـد دسـتور مربـوط بهWhile اجرا می شود و دوباره عبارت بولی بررسی می شود و این کار دوباره تکرار می شود. تا زمـانی کـه مقـدار عبـارت بـولی برابـر True باشد حلقهٔ While تکرار می شود.



نکته) اگر در اولین بار که شرط بررسی می شود شرط نادرست باشد آنگاه حلقه اصلا تکرار نمی شو
پس احتمال دارد که حلقه تکرار نشود.

مثال)برنامه ای بنویسید که یک عددمثبت از ورودی بخواند و رقمهای آن را چاپ کند.

```
Var
  R, N:Integer;
Begin
  Readln(N);
  While  N> 0  DO
     Begin
        R:=N  Mod  10;
        Writeln(R);
        N:=N  DIV  10;
      End;
End.
```

مثال) در برنامه روبرو دستور ;Writeln(X) چند بار اجرا خواهد شد؟ الف) 0 ب) 1 ج)4 د)5

```
X:=1;
While  X<5  Do
   Begin
       Writeln(X);
        X:=X+1;
    End;
```

اگر شرط(عبارت بولی) مربوط به While همیشه True باشد و اصلا False نشود آنگـاه حلقـه While هرگـز پایـان نمـی یابـد و در نتیجه یک حلقه بی نهایت بار است.(بی نهایت بار تکرار می شود) در این حالت کامپیوتر در یک وضعیت ثابت می ماند.

عبارت بولی(شرط) مربوط به حلقهٔ While باید طوری تنظیم شود که زمانی False شود.

نکته) اگر بعد از کلمهٔ رزرو شدهٔ Do علامت **;** قرار گیرد آنگاه علامت **;** به عنوان دستور(بلاک) while در نظر گرفته مـی شـود. در این صورت دو حالت رخ می دهد:

۱- شرط While درست باشد که در این صورت یک حلقه بی نهایت ایجاد شده است. ۲- شرط While نادرست باشد که حلقه while اصلا تکرار نمی شود.(خطای منطقی)

مثال) مثال قبل را در صورتی که بعد از Do علامت **;** قرار گرفته شده است حل کنید.

مثال)دستورات K:=K-1; و S:=S-1; هر کدام چند بار تکرار می شوند؟

```
K:=10;  S:=7;
While  K<>10  Do
      K:=K-1;
While  S>0  Do;
   S:=S-1;
```

نکته) زمانی از حلقه while برای ساخت حلقه استفاده می شود که تعداد تکرار حلقه مشخص **نباشد**.

مثال) برنامه ای بنویسید که یک جمله را از ورودی دریافت کرده تعداد کاراکتر t و T را بشمارد .(جمله به صفر ختم می شود)

```
var
   count:integer;
   ch:char;
begin
  writeln('Please Enter Statement');
   read(ch);
    while ch<>'0' do
      begin
        if (ch='t') or (ch='T') then
            count:=count+1;
         readln(ch);
      end;
    writeln('Number of characters= ',count);
    readln;
end.
```

مثال) برنامه ای بنویسید که N عدد از کاربر دریافت کند و ماکزیمم و مینیمم این اعداد را چاپ کند.(آخرین عدد ورودی صفر است).

```
var
  n,max,min:integer;
begin
   readln(n);
   max:=n;
   min:=n;
   while n<>0 do
     begin
       readln(n);
       if (n<>0)  and (n>max) then
           max:=n;
       if (n<>0) and (n<min) then
           min:=n;
     end;
    writeln('max=',max, '  min=',min);
end.
```

2 - دستور Repeat- Until :

شکل کلی دستور:

Repeat

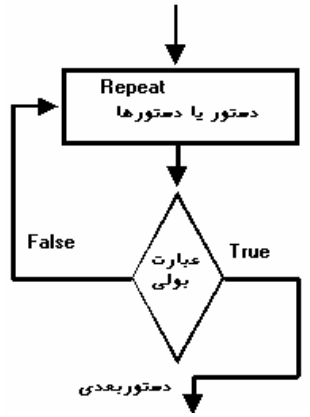
; دستور یا دستورها

; عبارت بولی Until

در این حلقه ابتدا دستور یا دستورها اجرا می شوند سپس عبارت بولی (شرط) بررسی می شـود اگـر مقـدار آن **FALSE** باشـد آنگـاه دوباره دستور یا دستورها تکرار می شوند. این عمل تا زمانی که شرط برابر False باشد تکرار می شود. شرط خروجی از این حلقـه ایـن است که شرط True شود.

مثال :

```
A:=2;
Repeat
  Writeln(A);
   A:=A+2;
Until  A>10;
Writeln('*');
```



فلوچارت مربوط به حلقهٔ Repeat:

نکته) بررسی شرط حلقه Repeat در انتهای حلقه صورت می گیرد به همین دلیل حتما یک بار انجام می شود اما شرط حلقهٔ While در ابتدای حلقه بررسی می شود به همین دلیل ممکن است یک بار هم انجام **نشود**.

نکته) حلقه While در صورتی که شرط False شود حلقه به اتمام می رسد اما در حلقهٔ Repeat اگر شرط True شود حلقه به اتمام می رسد.

نکته) کلمهٔ رزرو شدهٔ Until پایان حلقه را مشخص می کند به همین دلیل این دستور نیـازی بـه Begin و End نـدارد. یعنـی اگـر تعداد دستورات در بدنهٔ حلقه هم بیش از یکی باشد نیازی به استفاده از دستور مرکـب نیـست. بدنـه حلقـه از خـط Repeat تـا خـط Until است.

نکته) کلماتDO و While وrepeat و Until ذخیره (رزرو)شده هستند.

نکته) اگر برنامه نویس شرط حلقهٔ Repeat را طوری تنظیم کند که هرگز True نشود یک حلقه بی نهایت ایجاد می شود.

مثال) در مثال زیر شرط حلقه همیشه Flase است و هرگز از حلقه خارج نمی شویم،حلقه بی نهایت بار تکرار می شود.

```
Repeat
  Writeln('*');
   Writeln('ALI');
Until  False;
```

نکته) حلقه Reapet زمانی استفاده می شود که بخواهیم بدنه حلقه حداقل یک بار تکرار شود و تعداد تکرار مشخص نباشد.

تست)خروجی برنامه روبرو کدام است؟

```
Var   X:Integer  ; Y:Boolean;
Begin
  X:=1;
  Repeat
     Write(X:3);
      X:=X+2;
       Y:=X=11;
   Until  Y;
End.
```

الف) اعداد زوج کوچکتر از 10 ب) اعدادفرد ۱ تا ۹

ج) عدد ۱ د) Run Time Error

مثال) خروجی برنامه روبرو چیست؟

الف) معکوس عدد ورودی　　ب)عددی ورودی

ج) مجموع ارقام عدد ورودی　　د) رقم یکان عدد ورودی

```
Readln(X);  S:=0;
Repeat
  R:=X  MOD  10;
  X:=X  Div  10;
  S:=S+R;
Until   X<>0
Writeln(S);
```

مثال) برنامه ای بنویسید که مجموع اعداد یک تا صد را محاسبه و چاپ نماید.

```
Var
     i,sum:integer;
begin
    i:=1;
    sum:=0;
  repeat
       sum:=sum+i;
       i:=i+1;
  until i>100;
     writeln(sum);
      readln;
end.
```

مثال) برنامه ای بنویسید که یک عدد دریافت کند و چاپ کند که این عدد اول است یا مرکب.

```
Var
   c,n,t,r:integer;
 begin
    readln(n);
    t:=0;  c:=1;
    repeat
      r:=n  mod c;
      if r=0 then
        t:=t+1;
      c:=c+1;
    until  c>n;
  if t=2 then
    writeln('avval')
   else
    writeln('morakab');
 end.
```

3 - دستور FOR :

شکل کلی دستور For :

FOR متغیراسکالر :=مقدار اولیه TO یا DownTo مقدار نهایی DO
یک دستور ساده یا ساختاری ;

مثال) در مثال روبرو دستور ;('*')Write هشت بار تکرار می شود.

```
For A:=2 TO 9 Do
  Write('*');
Write('AMIN');
```

ابتدا A=2 می شود و دستور ('*')Write انجام می شود سپس A=3 میشود و این کار تکرار می شود تا زمانی که A=9 شود. البته برای A=9 نیز دستور ('*')Write انجام می شود.
دستور ;('AMIN')Write مربوط به for نمی شود و بعد از اینکه هشت تا * چاپ شد رشتهAMIN یک بار چاپ می شود.

نکته) متغیر مربوط به شمارش حلقهٔ for را شمارندهٔ حلقه می نامند.(متغیر اسکالر)

نکته) شمارنده حلقه در هر بار اجرای حلقه به طور خودکار یک *واحد* اضافه یا کاهش پیدا می کند. اگر از کلمـهٔ **TO** اسـتفاده کـرده باشید شمارندهٔ حلقه یک واحد افزایش می یابد که اصطلاحا می گویند **For افزایشی** ایجاد شده اسـت. امـا اگـر از کلمـهٔ **DownTo** استفاده کنید شمارندهٔ حلقه یک واحد کاهش می یابد که اصطلاحا می گویند **For کاهشی** ایجاد شده است.

مثال) مثال روبرو یک for کاهشی است که شمارندهٔ آن K است که از ۱۰۰ شروع می شود و یک واحد، یک واحد کاهش می یابد.

```
For K:=100 DownTo 1 Do
  IF  (K MOD 2)=0 then
      Writeln(K);
```

در مثال روبرو دستور IF که یک دستور ساختاری است بلاک مربوط به For است.
و در مثال روبرو اعداد زوج بین 100 تا 1 چاپ می شوند.
و دستور ;(K)Writeln بلاک مربوط به IF است که یک دستور ساده است.

نکته) در For افزایشی مقدار اولیه از مقدار نهایی باید کوچکتر یا مساوی باشد. در غیر اینصورت خطایی رخ نمی دهد بلکه حلقه تکرار نمی شود. درصورتی که مقدار اولیه و مقدار نهایی مساوی باشدحلقه یک بار تکرار می شود.

مثال)در مثال روبرو حلقهٔ For افزایشی است اما مقدار اولیه(3) از مقدار نهایی (1) بزرگتر است به همین دلیل دستور ;(T)Write اصلا تکرار نمی شود و دستور ;('*')Write انجام می شود.

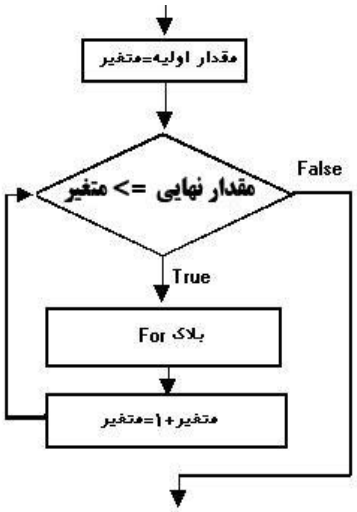
```
For T:=3  To 1 Do
  Writeln(T);
Write('*');
```

نکته) در For کاهشی مقدار اولیه از مقدار نهایی باید بزرگتر یا مساوی باشد. درغیر اینصورت خطایی رخ نمی دهد بلکه حلقـه تکـرار نمی شود. درصورتی که مقدار اولیه و مقدار نهایی مساوی باشد حلقه یک بار تکرار می شود.

مثال)در مثال روبرو حلقه For کاهشی است اما مقدار اولیه(1) از مقدار نهایی(5) کوچکتر است. به همین دلیل دستور ;(C)Writeln اصلا تکرار نمی شود و دستور ;('#')Write انجام می شود.

```
For C:=1  DownTo 5 Do
  Writeln(C);
Write('#');
```

نکته)مقدار اولیه و مقدار نهایی و شمارندهٔ حلقهٔ for باید از نوع اسکالر باشند در غیر اینصورت پیغام خطا صادر می شود.
نکته) کلمات **DO و For و Downto و TO** رزرو شده هستند.
نکته) هنگامی از حلقهٔ **For** استفاده می شود که **تعداد تکرار** حلقه **مشخص باشد**.



فلوچارت روبرو مربوط به عملکرد دستور For افزایشی است. دقت کنید که شرط ورود به حلقه در ابتدای حلقه چک می شود به همین دلیل احتمال دارد که حلقه اصلا تکرار نشود.

نکته) تعداد تکرار دستورات داخل بلاک for بستگی به مقدار اولیه و نهایی حلقه دارد. بافرض اینکه دستورات داخل حلقه روی متغیر شمارنده تاثیری نگذاشته باشد می توانیم بوسیله روابط زیر تعداد تکرار حلقه را بدست آورد

For افزایشی ç تعداد تکرار = مقدار نهایی – مقدار ابتدایی + 1

For کاهشی ç تعدادتکرار = مقدرا اولیه – مقدار نهایی + 1

شرط تکرار حلقه درست بودن شرط ابتدای حلقه است. دقت کنید که شرط در دستور نهفته است.

نکته) اگربعد از کلمهٔ Do علامت **;** قرار دهید خطایی رخ نمی دهد بلکه علامت **;** به عنوان بلاک for در نظر گرفته می شود.(خطای منطقی) که در این صورت یک حلقه تاخیر ایجاد شده است.

مثال) خروجی قطعه برنامه روبرو کدام است؟

```
For H:=1  To 12  Do;
  Write(H);
```

الف) اعداد ۱ تا ۱۲ ب) عدد ۱ ج) عدد ۱۲ د) برنامه خطا دارد

چون بعد از Do علامت ; قرار گرفته شده است پس دستور ;Write(H) مربوط به for نیست. به همین دلیل یک حلقه تاخیر ایجاد شده است. و متغیر H ابتدا ۱ و سپس ۲ و ... و در نهایت ۱۲ می شود و از حلقه تاخیر خارج می شود و دستور Write انجام می شود که مقدار H ، ۱۲ است. (گزینه ج)

نکته) پس از خاتمهٔ حلقهٔ For معمولا مقداری که در متغیر شمارنده قرار دارد قابل اعتبار نیست. اما در توربوپاسکال مقدار آن برابر مقدار نهایی است.

نکته) مقدار اولیه و نهایی حلقه می تواند عبارت باشد(عبارتی که حاصل آن اسکالر شود) اما مقدار عبارتها **یک بار ارزیابی می شود** و دیگر **تغییر نمی کند.**

مثال) در مثال روبرو مقدار نهایی حلقه در ابتدای حلقه برابر 6 می شود و حلقهٔ for از یک تا شش تکرارمی شود با اینکه مقدار M در هر بار اجرای حلقه تغییر می کند اما دیگر تاثیری در مقدار نهایی حلقه ندارد. اعداد ۱ تا ۶ چاپ می شود.(M از نوع صحیح است)

```
M:=5;
For A:=1  To  M+1  Do
 Begin
  Writeln(A);
   M:=M+2;
 End;
```

معمولا مقدار شمارندهٔ حلقه را در بدنهٔ حلقه **تغییر نمی دهند** اما اگر این کار را انجام دهیم تعداد تکرار حلقه تغییر خواهد کرد.

مثال) در قطعه برنامه روبرو مقدار اولیه 2 است و مقدار نهایی 12 است.شمارندهٔ حلقه H است که در هر بار اجرا بطور خودکار یک واحد به آن اضافه می شود و درهر بار اجرا در بدنهٔ For نیز **دو واحد** به آن اضافه می کنیم. پس **سه واحد، سه واحد** افزایش می یابد. اعداد 2 5 8 11 چاپ می شود و حلقه **چهار بار** تکرار می شود و مقدار نهایی شمارندهٔ حلقه 13 است.

```
S:=2;
For H:=S  To 12  Do
 Begin
   Writeln(H);
    H:=H+2;
    S:=17;
  End;
```

تست) حاصل اجرای برنامهٔ روبرو کدام است؟

```
Const  K:integer=100;
Begin
   For  K:=100  downto  1  do
        Write(K);
End.
```

الف) چاپ اعداد ۱۰۰ تا ۱ ب) ۱۰۰

ج) ۱ د) خطا دارد چون K از نوع ثابت است.

مثال)برنامه ای بنویسید که 5 عدد از ورودی دریافت کند و سپس تعـداد اعـداد زوج و فـرد را شـمرده و در خروجـی چـاپ نماید.

```
Var
   i,even,odd,number:integer;
begin
     writeln('Please Enter Five Numbers ');
       even:=0;
       odd:=0;
            for i:=1 to 5 do
               begin
                   read(number);
                    if (number mod 2)=0 then
                         even:=even+1
                    else
                          odd:=odd+1;
                end;
      writeln('Number of even= ',even);
      writeln('Number of odd= ',odd);
end.
```

مثال)برنامه ای بنویسید که یک عدد از ورودی دریافت کند و چاپ کند که این عدد اول است یا مرکب.

```
var
  c,n,t:integer;
begin
  readln(n);
  t:=0;
  for c:=1  to n do
   if   n   mod   c = 0 then
      t:=t+1;
  If   t=2   then
    writeln('avval')
  else
    writeln('morakab');
end.
```

تمرین) خروجی برنامه روبرو چیست؟

```
var
  a:boolean;
begin
   for a:=false to true do
     write(a);
end.
```

تمرین)خروجی برنامه روبرو چیست؟

```
var
  c:char;
begin
  for c:='A'  to 'F' do
    write(c);
end.
```

حلقه های متداخل:

همانطور که در شکل کلی دستورات حلقه مشخص شده بود بلاک مربوط به حلقه می تواند یک دستور ساختاری باشد از جمله یک حلقه دیگر. در این صورت دو حلقه متداخل ایجاد می شود.

نکته) تعداد تکرار داخلی ترین دستور حلقه های متداخل برابر حاصلضرب تعداد تکرار هر کدام از حلقه ها است.

مثال) در مثال روبرو دو حلقه متداخل ایجاد شده است. دستور Write('*'); مربوط به For دومی است و for دومی بلاک مربوط به For اولی است. برای H=1 حلقهٔ دوم پنج بار تکرار می شود یعنی پنج بار * چاپ می شود و برای H=2 دوباره حلقهٔ دوم پنج بار تکرار می شود یعنی .... بنابراین مجموعا ۸×۵=۴۰ بار * چاپ می شود.

```
For H:=1  To 8 Do
  For  Z:=3  downto  -1 Do
    Write('*');
```

مثال) خروجی قطعه برنامه روبرو چیست؟
حلقهٔ While و حلقهٔ For با هم ادغام شده اند.

```
A:=1; N:=0;
While  A<9  Do
 Begin
  For  Z:=1  To  4  Do
       N:=N+Z;
   A:=A+3;
 End;
Writeln(N);
```

نکته)هر سه دستور مربوط به حلقه ها می توانند با هم ترکیب شوند و حلقه های متداخل را ایجاد کنند.

مثال)برنامه ای بنویسید که خروجی زیر را داشته باشد :

```
1
1  2
1  2  3
1  2  3  4
```

```
Var
   i,j:integer;
begin
 writeln;
  for i:=1 to 4 do
     begin
        for j:=1 to  i  do
          write(j:3);
          writeln;
      end;
  readln;
end.
```

تمرین)خروجی برنامه روبرو چیست؟

```
var
  i,j:integer;
begin
for i:=1 to 5 do
 begin
  for j:=i downto 1 do
      write(j:3);
    writeln;
 end;
end.
```

مثال) برنامه ای بنویسید که اعداد اول بین 100 تا 200 را چاپ نماید.

```
var
   c,n,t:integer;
begin
   for n:=100 to 200 do
   begin
    t:=0;
   for c:=1  to n do
     if n mod c = 0 then
        t:=t+1;
   if t=2 then
     writeln(n);
   end;
end.
```

دستورات Break و Continue و Exit :

دستور Break باعث شکسته شدن حلقه ایی که داخل آن هستیم می شود.

مثال) در مثال روبرو اعداد 10 تا 5 چاپ می شود و به همین تعداد هم * چاپ می شود اما هنگامی که A=4 فقط عدد 4 چاپ می شود و دستور Break انجام می شود و از حلقه While خارج می شویم.

دقت کنید که هنگامی که دستور break انجام می شود کلا از حلقه خارج می شویم و دیگر بقیه دستورات حلقه انجام نمی شود اما بقیه دستورات برنامه دنبال می شود.

دستور Break را می توان برای هر کدام از دستورات حلقه بکار برد.

```
A:=10;
While  A>0  Do
  Begin
    Writeln(A);
     IF  A<=4  Then
        Break;
      Write('*');
     A:=A-1;
  End;
```

دستور continue باعث می شود :

کنترل اجرای حلقه به ابتدای حلقه منتقل شود.

مثال) در مثال روبرو هنگامی که S=5 درست میشود آنگاه دستور Write(S) انجام نمی شود. خروجی برنامه چیست؟

دستور Continue را می توان برای هر کدام از حلقه ها بکار برد.

اگر حلقه های متداخل داشته باشیم Continue و Break مربوط به هر حلقه فقط روی همان حلقه تاثیر می گذارد.

```
For  S:=9   DownTo  1
Do
  Begin
     Write('*');
     IF  S=5  then
          Continue;
      Write(S);
  End;
```

در مثال روبرو دستور Break به دلیل اینکه در بدنه While است پس فقط حلقهٔ while را می شکند . دستور Continue مربوط به حلقه for است و پس در صورت اجرا شدن Continue به ابتدای حلقه for می رویم.

```
For ……
   Begin
     While   …..
        Begin
             Break;

        End;
           Contine;
    End;
```

دستور Exit :

هدف : انتقال برنامه به خارج از بلوک فعلی

استفاده از این دستور در هر بلوک از برنامه باعث می شود که کنترل برنامه بلافاصله به خـارج از آن بلـوک انتقـال یابـد.دسـتور Exit اگردر بدنه اصلی بلوک یک پروسیجر یا تابع قرار گرفته باشد ، باعث خروج از پروسیجر یا تابع شده و کنترل برنامه بـه اولـین دسـتوری که بلافاصله پس از دستورفراخوانی آن پروسیجر یا تابع قرار گرفته پرش می کند.

اگر دستور Exit در بلوک اصلی برنامه باشد ، برنامه بلافاصله خاتمه می یابد.

تست :

تست۱) منظور از درجه ی عملگر کدام است؟

الف) تعداد عملگرهای موجود در یک عبارت　　ب) تعداد عملوندهای موجود در یک عبارت

ج) تعداد عملوندهای عملگر　　د) اولویت اجرای عملگر

تست۲) کدام گزینه یک شناسه استاندارد زبان پاسکال است؟

الف) For　　ب)Readln　　ج)Program　　د)String

تست۳) سرعت انجام محاسبات با کدام نوع داده نسبت به بقیه پایین تر است؟

الف)Longint　　ب)Real　　ج)Word　　د)Integer

تست ۴)در پاسخ به اجرای قطعه برنامه ی مقابل داده های زیر را وارد کرده ایم خروجی کدام است؟

```
Readln(X,Y,Z);  3 7 12 14
Read(A,B);   16  24  36
Write(A,B);
```

الف)16 14　　ب)36 24　　ج)24 16　　د)پیغام خطا

تست ۵) اگر I یک متغیر از نوع صحیح و r یک متغیر از نوع اعشاری باشد کدام دستور صحیح است؟

الف)r:=I Div 6;　　ب)I:=I+r　　ج)I:= r Mod 10;　　د)I:=I /32;

تست :

۱ -خروجی برنامه روبرو کدام است؟

```
N:=2;
For I:=1 to 2 do
   For k:=1 to 4 do
      N:=N+I+k;
```

الف) 36　　ب)32　　ج)29　　د)34

۲-در کدام ساختار تکرار برای مجموعه دستورات حلقه(بیش از یک دستور) نیازی به Begin-End نیست؟

الف) While-Do　　ب)For　　ج)For و Repeat-Until　　د) Repeat-Until

```
A:=6;
For   B:=1 to A do
     Begin
       A:=A-1;
       B:=B+1;
       Writeln(B);
     End;
```

۳ -خروجی برنامه روبرو کدام است؟

الف) **2 , 3 , 4 , 5 , 6**　　ب) **2 , 4**

ج) **1 , 2 , 3 , 4 , 5 , 6**　　د) **2 , 4 , 6**

۴- اگر متغیری به صورت ; **N:string** تعریف شود چند بایت فضا برای آن اشغال خواهد شد؟

الف)256　　ب)127　　ج)255　　د)254

۵ - خروجی برنامه روبرو کدام است؟

```
Begin
For  L:=6 downto   1 do;
  Write(L);
End.
```

الف) 1　　ب) 123456

ج) 654321　　د) 6

۶-کدام یک از ساختارهای حلقه حداقل یک بار انجام می شود؟

الف) For　　ب) While　　ج) Repeat　　د) الف و ج

۷- خروجی برنامه روبرو کدام است؟

```
For H:=FALSE  TO TRUE DO
 For  A:=1 TO  3  DO
      WRITE(A+1,'H');
```

الف) 2FALSE3FALSE4FALSE2TRUE3TRUE4TRUE
ب)برنامه در زمان ترجمه خطا دارد.
ج) 2H3H4H2H3H4H
د) 1H2H3H1H2H3H

۸- خروجی برنامه روبرو کدام است؟

```
For W:=4   downto 1.0  do
   Write(W+1);
```

الف) 5432 ب) 4321 ج)54321
د) برنامه خطا دارد.

۹- کدام گزینه در مورد دستور Repeat-Until صحیح است؟

الف)از این حلقه زمانی استفاده می شود که دفعات تکرار آن معلوم باشد.
ب)در صورتی که دستورات داخل حلقه بیش از یک دستور باشد باید آن را بصورت دستور مرکب نوشت.
ج) درصورتی که حاصل عبارت بولی حلقه نادرست باشد دستورات حلقه فقط یک بار اجرا می شود.
د) کلمات ذخیره شده ی Repeat و Until ابتدا و انتهای حلقه را مشخص می کند.

۱۰ - خروجی برنامه روبرو کدام است؟

```
A:=1;
Repeat
  A:=A+1;
  If  A>10 then
 Break;
  IF A mod 4 =0 then
    Continue;
 Writeln(A);
Until False;
```

الف)**2, 3, 4, 5, 6, 7, 8, 9, 10** ب)**2, 3, 5, 6, 7, 9, 10**
ج) **2** د)**2, 3**

۱۱- در برنامه روبرو دستور Writeln(X) چند بار تکرار می شود؟

```
X:=1;
While  X<5   Do ;
 Begin
      Writeln(X);
      X:=X+1;
 End;
```

الف) هیچ بار چون حلقه خاتمه نمی یابد. ب)یک بار
ج) ۴ بار د) ۵ یار

۱۲- در حلقه های متداخل تعداد دفعات تکرار دستور حلقه داخلی برابر است با:

الف) حاصل جمع تعداد تمامی حلقه ها ب) حاصل جمع تعداد تکرار حلقه داخلی و حلقه خارجی
ج)حاصلضرب تعداد تکرار تمامی حلقه ها د) حاصلضرب تکرار حلقه داخلی در حلقه خارجی

۱۳- در برنامه روبرو چندبار TRUE و چند بار False چاپ میشود؟

```
For A:=4 downto 1 do
  For F:=A  to 4  do
    If  F Mod 2 =0 then
      Writeln(F<=A);
```

الف) چهار بار True و دو بار False ج) شش بار True
ب) دو بار True و چهار بار False د) شش بار False

۱۴- خروجی برنامه روبرو کدام است؟

```
N:=1;
For W:=4   downto   1  do
   Begin
     F:=4;
                 While   F>W  do
        Begin
          N:=N+W;
                F:=F-1;
      END;
   END;
Writeln(N);
```

الف) **10**　　ب)**11**

ج) **36**　　د)**21**

۱۵- اگر دستور داخل حلقه for-to روی متغیر شمارنده تاثیر نداشته باشد و مقدار نهایی متغیر شمارنده برابر مقدار اولیه آن باشد تعداد تکرار دستور داخل حلقه و مقدار متغیر شمارنده بعد از اجرای حلقه کدام است؟

الف)صفربار- مقدار نهایی　　ب) یک بار – (۱+مقدار اولیه – مقدار نهایی)

ج) یک بار - مقدار نهایی　　د) یک بار – ۲

۱۶- خروجی برنامه روبرو چیست؟

```
X:=0;
FOR  I:=1  To  10  Do
IF   I>=5  then  X:=1;
IF  I<5  OR  X<>0  THEN Write(I);
```

الف) 12345678910　　ب)5678910

ج)12345　　د)10

۱۷- خروجی برنامه روبرو کدام است؟

```
Var  X:Integer;  Y:Boolean;
Begin
    X:=1;
     Repeat
          Write(X:3);
          X:=X+2;
           Y:=X=11;
      Until  Y;
End.
```

الف) صفر　　ب)اعداد فرد ۱ تا ۹

ج) عدد ۱　　د) Run Time Error

۱۸ - خروجی برنامه مقابل کدام است؟

```
For  K:=15  to  14  do
For  H:=1  to  2   do
For  N:=4   to  5   do
  Write('OK');
```

الف) ۶ بار چاپ OK　　ب) یک بار چاپ OK

ج) پیغام خطا　　د) هیچکدام

۱۹ - پس از اجرای برنامه روبرو مقدار متغیر count کدام است؟

```
For  H:=1  TO  18  DO
     Count:=60-H*3;
```

الف) **114**　　ب) **0**　　ج) **6**　　د) **-9**

۲۰ - دستور Write در برنامه روبرو چند بار اجرا می شود؟

```
A:=1;   F:=1;   W:=True;
              While Not W do
   Begin
            F:=F*A;
            A:=A+1;
             If   A>4 then
                 W:=False;
   End;
Writeln(F);
```

الف) یک بار　　ب) سه بار　　ج) اصلا اجرا نمی شود.　　د) ده بار

پیشتر بدانیم :

چند نکته در مورد انواع داده ها در زبان پاسکال:

داده های عددی **Single** و **Double** و **Extended** و **Comp** را انواع داده **8087** می گویند. این نوع داده ها برای انجام محاسبات بر روی آنها به کمک پردازشگر نیاز دارند. کمک پردازشگر یک سخت افزار است که برای انجام محاسبات بر روی اعداد اعشاری با دقت بالا استفاده می شود. در کامپیوتر های قدیمی (مدل XT) انجام محاسبات اعشاری با دقت بالا به کندی صورت می گرفت.

برای اینکه بتوانیم از داده های نوع 8087 استفاده کنیم باید کامپایلر پاسکال را برای استفاده از کمک پردازشگر مطلع سازیم. برای این کار می توانیم از دو روش استفاده کنیم.

۱- در منوی Option گزینهٔ Compiler را انتخاب کنید و گزینه 8087/80287 را √ (تیک)بزنید. در صورتی که √ نخورده باشد فقط می توانید از نوع Real برای اعداد اعشاری استفاده کنید.

۲- استفاده از رهنمود {$N+} در ابتدای برنامه:

رهنمودها:

علامت **$** چنانچه پس از علامت **{** به کار رود برای مترجم مفهوم خاصی دارد. این نوع علایم که نحوهٔ ترجمه برنامه را تغییر می دهند و مفهوم خاصی برای مترجم دارند و در واقع هدایت کنندهٔ مترجم در امر ترجمه برنامه هستند **رهنمود** نامیده می شوند.

اگر می خواهید برنامهٔ شما که دارای نوع دادهٔ ۸۰۸۷ است در کامپیوتر هایی که کمک پردازنده ندارند نیز اجرا شود باید نوع داده ۸۰۸۷ را شبیه سازی کنید که بوسیلهٔ دو روش ذکر شده در فوق این عمل را انجام می دهند.

علامت N+ استفاده از نوع داده ۸۰۸۷ و علامت E+ شبیه سازی نرم افزاری آن را مشخص می کند. {$N+,E+}

رهنمود {$N+,E-} باعث می شود برنامه فقط بر روی رایانه هایی که دارای کمک پردازشگر هستند اجرا شود.

نکته) اگر علامت $ به همراه عدد در پاسکال بیاید به معنای این است **که عدد در مبنای شانزده(Hex)** است. به عنوان مثال منظور از عدد $12 عدد 18 در مبنای ده است. دقت کنید که جلوی $ فقط عدد ثابت می آید. $3 همان عدد 3 در مبنای ۱۰ است و $D همان عدد ۱۳ در مبنای ده است. (A,B,C,D,E,F) منظور از عدد $23 چیست؟ منظور از #$41 کاراکتر 'A' است.

توابع کتابخانه ای:

در زبان پاسکال تعدادی تابع کتابخانه ایی وجود دارد که برنامه نویس بتواند به راحتی از آنها برای انجام بعضی کارها استفاده کند. به عنوان مثال تابع ABS قدر مطلق یک عدد را بر می گرداند. این توابع از قبل در پاسکال تعریف شده اند. هر تابع ممکن است دارای پارامتر ورودی باشد و یک خروجی نیز دارد. در ادامه هر تابع را با عملکرد آن و پارامتر ورودی و خروجی آن بیان می کنیم.

نکته) توابع کتابخانه ایی هیچکدام کلمهٔ رزرو شده نیستند.

تابع **ABS**: یک پارامتر ورودی دارد که باید عددی باشد(صحیح یا اعشاری) و خروجی آن یک عدد مثبت یا صفر خواهد بود که قدر مطلق عدد ورودی است. ABS(0)=0 ABS(-6)=6 ABS(5)=5

تابع SQR: یک پارامتر ورودی دارد که باید عددی باشد و خروجی آن مجذور(مربع) عدد ورودی است. بنابراین حاصل آن همیشه یک عدد مثبت یا صفر خواهد بود. SQR(0)=0 SQR(-4)=16 SQR(5)=25

تابع SQRT : یک پارامتر ورودی دارد که باید از نوع عددی باشد و خروجی آن جذر(رادیکال) عدد ورودی است. بنابراین عدد ورودی نمی تواند عدد منفی باشد زیرا زیر رادیکال نمی تواند منفی باشد. خروجی این تابع از نوع Real خواهد بود.

Sqrt(0)=0.0 SQRT(-9)= خطا SQRT(81)=9.0

SQRT(ABS(-9))=3.0 SQR(SQRT(A))=A

تابع Pi : این تابع پارامتر ورودی ندارد اما خروجی آن از نوعReal است و عدد **π** که همان 3.14159 را برمی گرداند.

تابع Frac : این تابع یک پارامتر از نوع Real دارد و خروجی آن نیـز از نـوع Real اسـت و قـسمت اعـشار عـدد ورودی را برمـی گرداند. در مثال روبرو عدد 0.545 چاپ می شود.

```
A:=12.345;
Writeln(Frac(A+1.2));
```

Frac(5)=0.0 Frac(6.89)=0.89
FRAC(-3.0245) = -0.0245

تست) دستور ;Write(Frac(-45.0821):4:1) چه چیزی را چاپ می کند؟

الف) 0.0821– ب) 0.1 ج) -0.1 د)-0.0

تابع Trunc: این تابع یک عدد Real به عنوان ورودی دریافت می کند و قسمت صحیح آن را به صورت Integer بر مـی گردانـد. پس خروجی آن عدد صحیح است. دقت کنید که عدد را گرد نمی کند. Trunc(45.874)= 45

Trunc(0.125)=0 Trunc(-24.99) = -24 Trunc(Frac(X)) = 0

تابع INT : این تابع شبیه تابع Trunc است با این تفاوت که خروجی آن از نوع Real است. خروجی اعشاری دارد.

INT(0.125)=0.0 Int(45.874)=45.0 INT(-24.99)= -24.0 INT(FRAC(X))=0.0

تابع Round : ورودی این تابع از نوع Real است و خروجی آن از نوع Longint است و عدد ورودی را گرد می کند.

Round(52.48) **è** 52 Round(52.5) **è** 53 Round(19.61) **è** 20
Round(12) **è** 12 Round(-24.53) **è** -25 Round(-4.39) **è** -4

تابع ODD : این تابع یک عددصحیح(Integer) را دریافت می کند وخروجی آن از Boolean است یعنـی True یـا False . در صورتی که عدد ورودی فرد باشد True بر می گرداند و در صورتی که زوج باشد False بر می گرداند.

ODD(14) **è** False ODD(17) **è** True ODD(-21) **è** True

تست) دستور ;Write(ODD(-1)) چه چیزی چاپ می کند؟

الف) 0 ب)1 ج)False د)True

نکته) دقت داشته باشیدکه اگر نوع پارامتر ورودی با آن چیزی که تابع نیاز دارد همخوانی نداشته باشد پیغـام خطـا صـادر مـی شـود. ODD(12.6) خطاست.

تابع Random : این تابع پارامتر ورودی ندارد اما خروجی آن یک عدد Real (اعشاری) است و این تابع یک عدد شانـسی بـین 0 تا 1 را بر می گرداند. 0≤Random <1 . Random **è** 0.745

تابع Random(X) : این تابع یک عدد از نوع Word (صحیح) دریافت می کند عددی از نوع Word بین 0 تا عـدد ورودی بـه صورت شانسی برمی گرداند. 0≤Random(6)<6

نکته) برای اینکه اعداد شانسی تولید شده در هر بار اجرای برنامه متفاوت باشند از **پروسیجر ;Randomize** استفاده می کنیم.

تابع EXP : در ریاضی عدد نپرن برابر 2.718 است و با نماد e نمایش داده می شود. این تابع یک عدد Real دریافت می کنـد و یک عدد Real نیز برمی گرداند. $EXP(3)=e^3$

EXP(X) یعنی عدد نپرن به توان X .

EXP(0)=1 EXP(1)=e Exp(Exp(0))=e

تابع LN : این تابع همان لگاریتم طبیعی است. ورودی این تابع یک عدد Real و خروجی آن نیز یک عدد Real است.

$LN(X) = LOG_e X$ پیغام خطا **è** LN(0) LN(e)=1 LN(1)=0

نکته) پارامتر ورودی تابع LN نمی تواند منفی یا صفر باشد.پیغام خطا صادر می شود.

LN(EXP(X)) **è** X EXP(LN(X)) **è** X LN(EXP(0)) **è** 0

$X^Y = EXP(Y*LN(X))$ LN(LN(e)) **è** 0

LN(A/B) =LN(A)-LN(B) LN(A*B)=LN(A)+LN(B)

توابع مثلثاتی SIN وCOS : این توابع یک پارامتر برحسب رادیان دریافت می کنند (از نوع Real) و یک عدد Real برمی گردانند و مقدار Sin یا Cos را محاسبه می کنند. مثلا برای محاسبه Sin زاویه 30 درجه عبارت Sin(30*PI/180) را می نویسیم.

تابع ArcTan : این تابع یک پارمتر Real دریافت می کند و یک عدد Real بر می گرداند. عدد خروجی برحسب رادیان است. مثلا ArcTan(X) مقدار زاویه ایی را که تانژانت آن برابر X است را بر حسب رادیان بر می گرداند.

تابع Length : این تابع یک رشته به عنوان ورودی دریافت می کند و خروجی آن یک عدد Integer است. و طول رشته دریافتی را برمی گرداند.

Length(’’’’) **è** 1 Length(’ALI’+’REZA’) **è** 7

Length(’ALI’) **è** 3 Length(’AMIN’) **è** 4 Length(’’) **è** 0

تابع CHR : از این تابع برای دریافت کاراکتر معادل کد ASCII استفاده می شود. ورودی این تابع یک عدد بین 0 تا 255 است (از نوعBYTE) و خروجی آن از نوع CHAR است. CHR(65) **è** ’A’

کاراکتر بوق **è** CHR(7) CHR(ABS(-66)) **è** ’B’ خطا **è** CHR(-66)

نکته) کد **اسکی** کلید **ESC برابر 27** است.

تابع Upcase: یک پارامتر ورودی از نوع کاراکتر دارد و خروجی آن نیز از نوع کاراکتر(Char) است و حرف کوچک را به حرف بزرگ تبدیل می کند. UPCASE(’n’) **è** ’N’ UPCASE(’d’) **è** ’D’ UPCASE(’A’) **è** ’A’

هنگامی که یک کلید را در صفحه کلید فشار می دهید یک *یا* دو کد تولید می شود و در بافر صفحه کلید قرار می گیرد.بعضی کلیدها را تک کدی و بعضی را دوکدی می گویند. مثلا وقتی کلید A را می زنید کد 65 تولید می شود. اما وقتی کلید F1 را می زنید دو کد تولید می شود اول کد 0 که به آن **کد نول** می گویند و دوم کد50 تولید می شود. یا هنگامی که کلید ↑ را می زنید کد های 0 و 72 تولید می شود. کلید های Enter وESC تک کدی هستند. (در واقع بعضی از کلید های کنترلی نیز کلیدهای تک کدی می باشند)

تابعReadKey : این تابع پارامتر ورودی ندارد اما خروجی آن از نوع CHAR است و بافر صفحه کلید را خوانده و یک کاراکتر از آن را بر می دارد و بر می گرداند. این تابع مربوط به یونیت Crt است که در اینجا بحث می شود.بادستور C:=ReadKey; داخل متغیر C یک کاراکتر از بافر صفحه کلید قرار می گیرد. اگر بافر صفحه کلید خالی باشد منتظر فشار دادن یک کلید می ماند. اگر کلیدهای دوکدی زده شود دو کد تولید می شود و بوسیلهٔ دوبار خواندن بافر می توان آن را تشخیص داد.

تابع ORD : از این تابع برای دریافت عددی متناظر با ترتیب اعضای نوع داده استفاده می شود. یا به عبارت دیگر مرتبه یک نوع اسکالر را مشخص می کند. همانطور که قبلا در مورد انواع اسکالر گفته شد داده های اسکالر قابل شمارشند و قبل و بعد دارند. پارامتر ورودی تابع ORD باید یک نوع دادهٔ اسکالر(صحیح – کاراکتری – بولین) باشد و خروجی آن یک عدد از نوع Longint (صحیح) است.

ORD(’A’) **è** 65

ORD(’B’) **è** 66 ORD(0) **è** 0 ORD(1) **è** 1 ORD(34) **è** 34

ORD(-65) **è** -65 ORD(FALSE) **è** 0 ORD(TRUE) **è** 1

ORD(25/5) **è** خطا ORD (’TRUE’) **è** خطا ORD(’1’) **è** 49

تابع Pred : این تابع یک ورودی از نوع اسکالر دارد و مقدار قبل از پارامتر ورودی را بر می گرداند. بنابراین ورودی و خروجی آن از نوع اسکالر است.

Pred('B') **è** 'A' Pred(3) **è** 2

Pred(True) **è** False Pred(False) **è** True Pred('F') **è** 'E'

Pred('AC') **è** خطا Pred (-3) **è** -4 Pred(1) **è** 0

ORD(CHR(PRED(67))) **è** 66 CHR(ORD(PRED(67))) **è** 'B'

Pred(ORD(ODD(48))) **è** -1 Pred('0') **è** '\' Pred('A') **è** '@'

تابع Succ : این تابع دارای ورودی و خروجی اسکالر است و کار آن عکس کار Pred است و عنصر بعدی ورودی را برمی گرداند.

SUCC(1) **è** 2 SUCC('A') **è** 'B' SUCC(-7) **è** -6

SUCC(Pred(X)) **è** X SUCC(False) **è** True SUCC(True) **è** True

ORD(SUCC('A')) **è** 66 SUCC(10) **è** 11 SUCC('Z') **è** '['

SUCC (5.7) **è** خطا Succ('9') **è** SUCC(A)=A+1

در ادامه تعدادی پروسیجرهای (زیرروال های) کتابخانه ایی را توضیح می دهیم. پروسیجرها شبیه توابع هستند با این تفاوت که خروجی ندارند و مقداری را برنمی گردانند. البته می توانند پارامتر ورودی داشته باشند. پروسیجرها چون خروجی ندارند نمی توان آنها را در عبارات صدا زد(فراخوانی کرد). پروسیجرها را باید به تنهایی فراخوانی کرد. به عنوان مثال دستور Writeln(ABS(-5)*2); صحیح است زیرا ABS یک تابع است و می توان آن را در عبارات و دستورات دیگر به کار برد. دستور C:=DEC(A)*2; اشتباه است زیرا DEC یک پروسیجر است و نمی تواند در عبارات یا دستور خروجی قرار گیرد و باید به صورت DEC(A); فراخوانی شود.در فصل بعدی در مورد توابع و پروسیجرها و اختلاف آنها بیشتر بحث می شود.

پروسیجر INC : این پروسیجر یک پارامتر ورودی از نوع اسکالر دارد و باید پارامتر ورودی آن حتما متغیر یا ثابت بانوع باشد. به مثالهای زیر دقت کنید:

```
A:=12;
INC(A);
Writeln(A);
```

```
A:=16;
INC(A,5);
Writeln(A);
```

```
S:=8;
INC(S,-3);
Writeln(S);
```

```
H:=45;
INC(H,0);
Writeln(H);
```

نکته) دستور INC(4); غلط است.زیرا **پارامتر ورودی نباید ثابت باشد**. پارامتر ورودی **نباید عبارت** باشد.

نکته) پروسیجر INC(A,4); معادل دستور A:=A+4; است و پروسیجر INC(A); معادل A:=A+1; است.

```
Z:='B';
INC(Z);
Write(Z);
```

```
M:=5.3;
INC(M+1);
Write(M);
```

پروسیجر DEC : **این پروسیجر عکس پروسیجر INC است و پارامترهای ورودی آن نیز شبیه INC است.**

```
S:=5;
DEC(S);
Write(S);
```

```
F:=12;
DEC(F,4);
Writeln(F);
```

```
H:='C';
DEC(H,2);
Writeln(H);
```

```
D:=5;
DEC(D,-3);
Writeln(D);
```

نکته) INC(A,-1); معادل DEC(A); است و DEC(B,-3); معادل INC(B,3); است.

نکته) DEC(A,5); معادل A:=A-5; است.

تست)

۱- خروجی تابع INT از چه نوعی است؟

الف) Integer ب) Char ج) Real د) Boolean

۲ - خروجی برنامه روبرو به ترتیب کدام است؟ (چپ به راست)

```
A:=10;   B:=-13.6;
DEC(A);
Writeln(A,Round(B),Trunc(B));
```

الف) 11,-14,-13 ب) 10,-14,-14
ج) 9,-13,-14 د) 9,-14,-13

۳- خروجی دستور روبرو کدام است؟ Write(CHR(ORD(PRED('B'))+2));

الف) 67 ب) 68 ج) C د) D

۴- حاصل تابع Frac از چه نوعی است؟

الف) Char ب) Integer ج) Word د) Real

۵- نوع پارامتر ورودی و خروجی تابع Odd(X) چیست؟

الف) اعشاری ، منطقی ب) اعشاری، اعشاری ج) صحیح، منطقی د) صحیح ، صحیح

۶ - با اجرای برنامه مقابل در خروجی چه چیزی چاپ خواهد شد؟

```
For K:=1  To  3  Do
 For M:=2  To  6  Do
   Begin
        Z:=frac(Int(K*M));
         W:=Z*M+1;
   End;
Write(W:2:0);
```

الف) 1 ب) 109 ج) 108
د) 2

۷ - کدامیک از توابع پاسکال می تواند عدد 6.6 را به 6 تبدیل می کند؟

الف) ODD ب) Round
ج) Trunc د) Frac

۸- کدام گزینه حرف F را چاپ می کند؟

الف) Write(CHR(ORD('F'))); ب) Write(ORD('F'));
ج) Write(ORD(CHR('F'))); د) Write(Pred('F'));

۹- حاصل اجرای حلقه مقابل چیست؟

```
For  A:=-5  Downto  -10 Do
 Begin
      Write(Ord(ABS(A):2);
       Dec(A);
 End;
```

الف) 10 8 6 ب) 9 7 5
ج)هیچ چیز در خروجی چاپ نمی شود د) -6 -8 -10

۱۰-پس از اجرای فرمان روبرو کدام گزینه صحیح است؟ Write(Round(-13.9))

الف) -14 ب) -13 ج) 13 د) 14

۱۱ - خروجی برنامه روبرو کدام است؟

```
A:=100;
Writeln(DEC(A,2));
```

الف) 98 ب) 102 ج) 100 د)خطا

۱۲-کدامیک از رهنمودهای زیر باعث میشود که داده های ۸۰۸۷ را زمانی بتوان استفاده کرد که کمک پردازشگر ۸۰۸۷ موجود باشد؟

الف) {$N+} ب) {$N+,E-} ج) {$N+,E+} د)هیچکدام

۱۳-نتیجهٔ اجرای دستور;(Write(Succ(True)) چیست؟

الف)False ب)True ج)1 د)خطای Run Time Erro

۱۴-خروجی دستور زیر کدام است؟ WRITE(CHR(ord(Pred(Upcase('b')))));

الف)A ب)C ج)a د)65

۱۵-عبارت مقابل معادل کدام عبارت جبری است؟ INT(A+B)/FRAC(B)+SQR(A)

الف) $\frac{[A+B]}{[B]+A^2}$ ب) $\frac{[A+B]}{B-[B]+A^2}$ ج) $\frac{[A+B]}{B-[B]}+A^2$ د) $\frac{[A+B]}{[B]}+A^2$

۱۶-پس از اجرای دستور روبرو مقدار متغیر X کدام خواهد بود؟ X:=SQRT(ABS(INT(-9.1)));

الف)-2.8 ب) 2.8 ج)-3 د)3

۱۷-کدام گزینه از کلمات رزرو شده زبان پاسکال می باشد؟

الف)Integer ب)Readln ج)VAR د)TRUE

۱۸-خروجی دستور روبرو کدام است؟ Write(Ord(0));

الف)0 ب)1 ج)False د)True

۱۹-پس از اجرای دستور روبرو مقدار X کدام خواهد بود؟ X:=Pred(3.5);

الف)2 ب)3 ج)4 د)از تابع Pred استفادهٔ غیر مجاز شده است.

۲۰-خروجی دستور روبرو کدام است؟ Write(ODD(2));

الف)1 ب)3 ج)False د)True

۲۱-حاصل عبارت (CHR(ORD('a') AND ($CF در خروجی برابر کدام گزینه است؟

الف)$97 ب)65 ج)A د)a

۲۲-حاصل عبارت روبرو چیست؟ 5+2/8*3-SQRT(8 DIV 2) +2 MOD Round(6 div 4)

الف)-0.375 ب)3.75 ج)4.75 د)5.25

```
For A:=1 to 150 Do
   X:=Succ(Abs((A+1))+125 Mod 5 – 19 Div 3;
```

۲۳ - خروجی قطعه برنامه چیست؟

الف)153 ب)187 ج)790 د)146

۲۴-حاصل عبارت مقابل چیست؟ Round(Exp(Ln(5.6)-Ln(2))+ 10 Mod -3

الف)5 ب)2 ج)4 د)3

۲۵- خروجی برنامه روبرو کدام است؟

```
D:=20;
Writeln($D);
```

الف) 32 ب)0 ج)13 د)20

۲۶- کدام رابطه صحیح است؟(X یک عدد اعشاری است)

الف) X=TRUNC(X) ب)X=TRUNC(X)+FRAC(X)

ج)X=Round(X)+Frac(X) د) X=Round(X) +INT(X)

۲۷-حاصل عبارت روبرو چیست؟ Trunc(Exp(3*Ln(2)))

الف)6 ب)8 ج)5 د)9

یونیت Unit :

توابع کتابخانه ایی که در قسمت قبلی بیان کردیم همگی داخل یک فایل هستند که قبلا تعریف شده اند. این توابع توسط برنامه نویسان دیگر نوشته شده است و شما می توانید از آنها استفاده کنید. **یونیت، مجموعه ای از تعاریف شامل نوع داده ها، ثابتها، متغیرها و زیربرنامه ها است که به طور مستقل ترجمه می شود و در برنامه های دیگر استفاده می شود**. هر یونیت در یک فایل با **پسوند TPU** ذخیره شده است. توربوپاسکال ۷ دارای تعدادی یونیت است که هر کدام از آنها شامل تعدادی زیربرنامه است. مانند یونیت های CRT و Graph (توابعی جهت رسم خط و دایره و اشکال گرافیکی) و System و Dos و ....

هر برنامه نویس می تواند یونیت ایجاد کند و در برنامه های خود از آن استفاده کند.

هنگامی که می خواهیم از توابع یا بطورکلی از تعاریف موجود در یک یونیت استفاده کنیم باید در قسمت تعاریف یونیت را مشخص کنیم، **برای این کار از کلمهٔ ذخیره شدهٔ Uses استفاده می شود**. مانند **Uses Crt;**

یونیت System شامل تعاریف پرکاربرد استاندارد توربوپاسکال است مانند دستورات Readln و Writeln وinteger وreal و غیره. توابع کتابخانه ای که قبلا بیان شد نیز در یونیت System قرار دارند.

نکته) یونیت system از سوی مترجم به طور خودکار در برنامه استفاده می شود و نیازی به تعریف آن در قسمت Uses نیست.

در قسمت های بعدی تعدادی از توابع موجود دریونیت Crt را بیان می کنیم.

زیربرنامه ها (توابع و پروسیجرها):

هنگامی که می خواهیم یک برنامهٔ بزرگ بنویسیم تعداد دستورات آن (تعداد خطوط برنامه) بسیار زیاد می شود که در این صورت عیب یابی برنامه مشکل و وقت گیر خواهد بود. به همین دلیل برنامه را به قطعات کوچکتری تقسیم می کنیم و برای هر قطعه یک زیربرنامه (زیرالگوریتم) می نویسیم. حال ترکیب همهٔ این زیربرنامه ها برنامهٔ اصلی را تشکیل می دهند. حال هر زیربرنامه می تواند به تنهایی نوشته شده و به طور مستقل آزمایش شود. هر زیربرنامه مانند یک برنامه دارای بخشهای عنوان، تعاریف و بلاک است. **زیربرنامه ها به دو دسته تقسیم می شوند: تابع (فانکشن) و پردازه(زیرروال یا پروسیجر)**. این دو دسته شباهت ها و اختلافاتی با هم دارند که بیان خواهد شد.

مزایای استفاده از زیربرنامه ها : ۱- قابلیت خوانایی برنامه افزایش می یابد. ۲-درک مسائل کوچکتر آسانتر است. ۳- از یک زیربرنامه می توان در یک یا چند برنامه مختلف به دفعات استفاده کرد. ۴-اشکال زدایی آسانتر است. ۵- کارگروهی را امکان پذیر می کند. ۶- سرعت طراحی و نوشتن برنامه را بالا می برد.

نکته)زیربرنامه ها در بالای بلاک اصلی تعریف می شوند.

پردازه ها (زیرروال - پروسیجرها) :

برای تعریف یک پردازه از کلمهٔ رزرو شده **Procedure** استفاده می شود. شکل کلی :

```
Procedure نام پردازه ( نوع آنها و پارامترهای ورودی);
تعاریف مربوط به این پردازه
Begin
……..
End;
```

نام پردازه چون یک شناسه غیراستاندارد است باید از قوانین نام گذاری شناسه های غیراستاندارد پیروی کند. تعاریف مربوط به هر پردازه مخصوص همان پردازه است و فقط در محدودهٔ همان پردازه می تواند استفاده شود. قسمت تعاریف می تواند شامل تعریف متغیر یا ثابت یا هر شناسهٔ دیگر(حتی یک زیربرنامه دیگر) باشد.

در برنامه اصلی می توان یک پردازه را فراخوانی کرد(صدا زد). برای این کار ***فقط نام آن را نوشته و پارامترهای ورودی آن را به آن تحویل می دهیم***. هنگامی که در بلاک اصلی پروسیجر را صدا می زنیم کنترل اجرای برنامه به قسمت پروسیجر صدازده شده می رود و همهٔ دستورات نوشته شده در قسمت دستورات آن انجام می شود و سپس به برنامه اصلی برمی گردد و دنبالهٔ برنامه را اجرا می کند.

نکته) دقت کنید که هر پروسیجر دارای قسمت دستورات است که با END با علامت **;** به پایان می رسد.

نکته) یک برنامه می تواند چندین پروسیجر داشته باشد.

```
Var     H:integer;
Procedure  Power(M:integer);
  Var
       Z:integer;
  Begin
      Z:=M*M;
      Writreln(Z,H);
       Writeln('*');
  End;
Begin
  H:=6;
  Writeln('@');
  Power(H);
  Writeln('$');
End.
```

مثال)در مثال روبرو پروسیجر Power تعریف شده است. متغیر Z مربوط به پروسیجر Power است که اصطلاحا به این متغیر، متغیر محلی می گویند.
پارامتر ورودی پروسیجر تعریف شده یک عدد Integer است.

نکته)تعداد و نوع پارامترهای ورودی با آنچه در هنگام فراخوانی به آن تحویل می دهیم باید همخوانی داشته باشد در غیر صورت پیغام خطا در زمان ترجمه صادر می شود.

هنگامی که این برنامه اجرا می شود ابتدا بلوک اصلی اجرا می شود یعنی H:=6; و Writeln('@'); اجرا می شوند سپس با فراخوانی پروسیجر به داخل بلاک دستورات پروسیجر Power می رویم و دستور Z:=M*M; و Writeln(Z,H); و Writeln('*'); اجرا می شوند با رسیدن به انتهای بلاک پروسیجر به بلاک اصلی بازمی گردد و دستور Writeln('$'); اجرا می شود و برنامه خاتمه می یابد.

دقت کنید که تا زمانی که پردازه را صدا نکنیم پردازه اجرا نمی شود.

در مورد متغیر Z باید گفت که این متغیر یک متغیر محلی است یعنی فقط در بلاک مربوط به پروسیجر Power قابل استفاده است. اگر این متغیر را در بلاک اصلی استفاده کنیم پیغام خطا صادر می شود.(در زمان ترجمه) عمر متغیرهای محلی از زمانی آغاز می شود که دستور صدازدن پروسیجر اجرا می شود و زمانی عمر آنها به پایان می رسد که اجرای پروسیجر به پایان برسد و به محل صدا زدن آن برگردیم.

اما در مورد متغیر H باید گفت که این متغیر **یک متغیر سراسری است یعنی در همه جای برنامه قابل دسترسی است**. عمر متغیرهای سراسری از زمان اجرای برنامه شروع می شود و تا زمانی که برنامه پایان یابد ادامه دارد.بنابراین **مدت عمر آنهـا، مـدت زمـان اجـرای برنامه است و طولانی است**. مثلا متغیر H را می توان در بلاک اصلی استفاده کرد و در پردازهٔ Power نیز می تواند استفاده شود.

نکته) پارامترهای ورودی نیز مانند متغیرهای محلی هستند و مدت عمر آنها به طول اجرای پروسیجر بستگی دارد یعنـی وقتـی پروسیجر اجرایش تمام می شود عمر پارامترهای ورودی نیز تمام می شود. در مثال بالا پارامتر ورودی M فقط در بلاک پروسیجر قابـل استفاده است.

در پاسکال پارامترها به **دو دسته** تقسیم می شوند:

۱- پارامترهای مجازی یا صوری(ظاهری) (Formal parameters)

۲- پارامترهای واقعی(حقیقی) (Actual parameters)

به پارامترهایی **که هنگام تعریف پروسیجر مورد استفاده قرار می گیرند پارامتر مجازی** می گویند. به پارامترهایی که هنگام فراخوانی پروسیجر تحویل پروسیجر می شوند پارامترهای واقعی می گویند. **در ازای هر پارامترمجازی یک پارامتر واقعی** وجود دارد.**پارامترهای مجازی به سه دسته تقسیم می شوند**:

1 - **پارامترهای مقدار(ارزشی – value)**

2 - **پارامترهای متغیر(مرجع - Variable)**

3 - **پارامترهای ثابت(const)**

پارامتـر مقـدار (Value) یا پارامتر ورودی(Call by Value) :

پارامتر مقدار به پارامترهایی گفته می شود که فقـط می توان از آنها برای ارسال مقادیر به زیربرنامه استفاده کرد. به عبارت دیگر برای ورود داده ها به پروسیجر استفاده می شوند. اگر این پارامتر (پارامتر مجازی مقدار) تغییر کند هیچ تاثیری بـر روی پـارامتر واقعـی آن ندارد. **مثال**:

```
Var  Z,K:Real;
Procedure  ADD(D:Real; Var  N:Real);
  Begin
      D:=D+2;
      N:=N+5;
      Writeln(D+N);
  End;
Begin
   K:=14;  Z:=25;
   ADD(K,Z);
    Writeln(K,Z);
End.
```

در مثال روبرو پارامترورودی D از نوع Value است هنگامی که پروسیجر Add فراخوانی می شود مقدار K وارد D می شود و چون D از مقدار(Value) است تغییرات D هیچ تاثیری بر روی K ندارد. اما پارامتر مجازی N از نوع متغیر(Variable) است. که بحث خواهد شد. در این برنامه مقدار K ابتدا 14 است و هنگامی که پروسیجر Add انجام می شود همچنانچه 14 می ماند.

پارامتر واقعی مربوط به این پارامتر مجازی می تواند ثابت یا متغیر یا یک عبارت باشد. مثلا در این مثال هنگام فراخوانی ADD می توان عبارت ;Add(K*3,Z) را قرار داد.

تمرین) مشخص کنید که با اجرای برنامه زیر برای دستورهای Writeln چه اعدادی در خروجی چاپ می شوند؟

```
program ex1;
var
  x,k:integer;
procedure test(m:integer;n:integer);
  var
    z:integer;
begin
   z:=m+n;
   m:=m+2;
   n:=n+7;
   x:=x+1;
   writeln(x,m,n);
end;
begin
    x:=12;
    k:=50;
    test(x,k);
    writeln(x,k);
end.
```

راهنمایی : برای جواب دادن به اینگونه سوالها ابتدا جدولی مطابق جدول زیر رسم می نماییم و سپس با مقدار دهی متغیرها به راحتی و با کمترین اشتباه پاسخ صحیح را بدست می آوریم :

سراسری	سراسری	محلی	محلی	محلی
x	k	m	n	z
12	50	12	50	62
13		14	57	

همان طور که مشاهده می کنید با اجرای برنامه روبـرو در دسـتور Writeln داخل پروسیجر برای متغیرهای xوmوn به ترتیب اعـداد 13 و 14 و 57 وبرای دستور Writeln داخل بلاک اصـلی برنامـه برای متغیرهای xوk به ترتیب اعداد 13 و 50 چاپ می شود.

تمرین) مشخص کنید که با اجرای برنامه زیر برای دستورهای Writeln چه اعدادی در خروجی چاپ می شوند؟

```
program ex2;
var
  x,k:integer;
procedure test(m:integer;k:integer);
 var
   x:integer;
begin
   x:=m+k;
   m:=m+2;
   k:=k+7;
   x:=x+1;
   writeln(x,m,k);
end;
begin
    x:=12;
    k:=50;
    test(x,k);
    writeln(x,k);
end.
```

رسم جدول :

محلی	سراسری	محلی	سراسری	محلی
k=k' محلی	k	m	x'	x
50	50	12	62	12
57		14	63	

حال شما مشخص کنید که با اجرای این برنامه چه اعدادی بـــرای دو دستور Writeln در خروجی چاپ خواهد شد؟

writeln(x,m,k); ⇨ .....

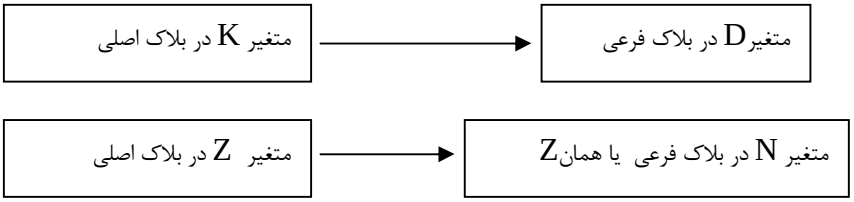
writeln(x,k); ⇨ .....

پارامتـــر متغیـــر (Variable) یا مرجـــع (Call by Reference) :

این پارامتر **برای دریافت و ارسال مقادیر به پروسیجر استفاده می شود یعنی هم برای ورودی داده ها به زیربرنامه استفاده می شود و هم برای دریافت داده از زیربرنامه. به همین دلیل به این پارامترها، پارامتر خروجی نیز می گویند.** نشانهٔ **پارامترخروجی** وجود کلمهٔ **Var** قبل از نام پارامتر در خط عنوان زیربرنامه است. هنگامی که پارامتر مرجع (خروجی) در بدنهٔ زیربرنامه تغییر می کند **تغییرات آن در پارامترواقعی آن اعمال می شود**. می توان گفت که پارامتر مرجع و پارامتر واقعی آن **یک محل در حافظه هستند با دو نام**.

در مثال صفحه قبل پارامتر مجازی N از نوع مرجع است که پارامتر واقعی آن Z است. Z مقدار 25 را تحویـل N مـی دهـد سـپس در بدنه پروسیجر N:=N+5; اجرا می شود به همین دلیل N مساوی 30 مـی شـود و چـون یـک پـارامتر مرجـع اسـت Z نیـز 30 مـی شود.وقتی پروسیجر تمام می شود Z مساوی 30 شده است.

نکته) پارامتر واقعی مربوط به یک پارامتر مجازی مرجع(متغیر) حتما باید یک متغیر(یاثابت با نوع) باشد در غیر این صورت خطا است. مثلا در مثال قبلی نمی توان بجای ADD(K,Z); دستور ADD(K,Z*4); را نوشت.



تمرین) مشخص کنید که با اجرای برنامه زیر برای دستورهای Writeln چه اعدادی در خروجی چاپ می شوند؟

```
program T;
var
  x,k:integer;
proceduretest(var m:integer;k:integer);
begin
   m:=m+1;
   x:=x+4;
   k:=k+7;
   writeln(x,m,k);
end;
begin
    x:=20;
    k:=8;
    test(x,k);
    writeln(x,k);
end.
```

رسم جدول :

-	سراسری	محلی
m (مرجع) x	k	k=k' محلی
20	8	8
21		15
25		

writeln(x,m,k); ⇨ .....

writeln(x,k); ⇨ .....

پارامتـرثابت(Const):

این پارامترها مانند پارامترهای مرجع هستند با این تفاوت که مقدار پارامتر ثابت را در طول اجرای زیربرنامه نمی توان تغییر داد. بـرای شناسایی این پارمترها از کلمهٔConst استفاده می شود. اگر مقدار آن را در بدنهٔ بلاک زیربرنامه تغییر دهیم در زمان ترجمـه خطـا رخ می دهد.

در **مثال** روبرو پارامتر **w از نوع ثابت** است و **نمی توان مقدار w را در بدنهٔ پروسیجر test تغییر داد.**

```
program T;
var
  x:integer;
procedure test(const w:integer);
begin
   w:=w+1;
   writeln(w);
end;
begin
    x:=20;
    test(x);
    writeln(x);
end.
```

نکته) پارامتر واقعی مربوط به پارامترثابت می تواند متغیرثابت یا عبارت باشد.

متغیر w محلی است و متغیر x سراسری هست.

خروجی برنامه روبرو کدام است؟

نکته) در مثال روبرو متغیرهای A,B,C سراسری نیستند زیرا این متغیرها در پایین پروسیجر تعریف شده اند. این متغیرهای محلی هستند و منطقه قابل استفاده آنها در بلاک اصلی برنامه است.

```
Procedure  M(Var x,y:Integer ; Z:Integer);
Begin
  X:=X-Y*2;
  X:=X  div  Y  Mod  X;
  Z:=X+Y;
End;
Var
   A,B,C:Integer;
Begin
  A:=2;   B:=8;   C:=6;
  M(B,A,C);
  Write(A, B , C);
End.
```

نکته)

هنگامی که تعداد پارامترهای مجازی بیشتر از یکی باشد بین آنها ; قرار می دهیم.

خروجی برنامه روبرو چیست؟

```
Var
    M:Integer;   H:Byte;
Procedure  AVG(Var  A:Byte);
  Begin
   M:=M+3;
    Write(H);
    A:=A-2;
    Write(H);
  End;
Procedure  FK(M:integer);
 Begin
      Write('*');
       AVG(H);
       M:=M+8;
  End;
Begin
    M:=2;   H:=10;
   FK(M);
  Write(H,M);
  AVG(H);
   Write(H,M);
End.
```

مثال) پروسیجری بنام ComputeSumAve بنویسید که مجموع و میانگین دو عدد را محاسبه و نتیجه را در برنامه اصلی چاپ نماید.

```
Var
    Num1,Num2:integer;
    Total,Average:Real;
Procedure ComputeSumAve(Num1,Num2:integer; Var sum,ave:Real);
 Begin
       Sum:=Num1+Num2;
       Ave:=sum/2;
 End;
Begin
     Readln(Num1,Num2);
     ComputeSumAve(Num1,Num2,Total,Average);
     Writeln('The Sum is= ' , Total:8:2,'The Average is= ', Average:8:2);
     Readln;
End.
```

تست:پس از اجرای برنامه روبرو خروجی کدام است؟

```
Var   A:integer;
Procedure  PK(m,n:integer);
  Begin    writeln(m+n);
           A:=10;
  End;
Begin   A:=20;
      PK(A,5);
       Writeln(A);
 End.
```

الف)25 ب) 25 ج) 10

20 10 25

د) برنامه خطا دارد

تست: خروجی برنامه روبرو کدام است؟

الف)3 2 ب)6 2

ج)2 6 د) 3 6

```
Const  X:integer=2;
   Y:integer=3;
Procedure K(X:Integer;Var  Y:Integer);
Begin
   X:=X+Y;
    Y:=X-Y;
     Y:=Y*3;
End;
Begin
  K(X,Y);
  Write( X,Y);
End.
```

تست:خروجی برنامه روبرو چیست؟

الف) 9 ب)2

ج)7 د)پیغام خطا

```
Const    X:Byte=2;
Procedure  P(X:byte);
Begin
    X:=X+7;
End;
Begin
Write(P(X));    End.
```

مثال) برنامه بنویسید که ابتدا دو عدد مثبت yوx رادریافت کند و سپس میانگین هماهنگ آنرا حساب کند :

توضیح : برای محاسبه میانگین هماهنـگ مقلـوب x را بـا مقلوب y جمع می کنیم و تقسیم بر دو می کنیم ، حاصـل را مقلوب می کنیم .عدد بدست آمده میانگین هماهنگ است.

```
procedure magh(n:integer ; var m:integer);
begin
   m:=0;
   while n>0 do
     begin
       m:=m*10 + n mod 10;
       n:=n div 10;
     end;
end;
procedure avgH(x,y:integer; var z:integer);
   var
     a,b,c:integer;
  begin
    magh(x,a);
    magh(y,b);
    c:=(a+b) div 2;
    magh(c,z);
  end;
var
  h,x,y:integer;
begin
   readln(x,y);
   avgH(x,y,h);
   writeln(h);
end.
```

تابــع یا فانکشن(Function):

توابع نوعی زیربرنامه هستند و مانند پروسیجرها پارامترهای ورودی (از نوع مرجع یا مقدار یا ثابت) دارند. اما تابع حداقل یک خروجـی دارد. در مواردی که حداقل به یک خروجی نیاز داریم از تابع استفاده می کنیم.

برای تعریف تابع از کلمهٔ رزرو شدهٔ **Function** استفاده می شود. هرآنچه را که بـرای پروسـیجرها بحـث کـردیم بـرای تـابع هـا هـم همانطور است. **تنها تفاوت تابع با پردازه در این است که تابع دارای یک خروجی است**. شکل کلی تعریف تابع:

```
Function   نام تابع ( پارامترهای مجازی و نوع آنها ) : نوع خروجی تابع ;
    تعاریف مربوط به تابع
Begin
…….. دستورات
End ;
```

نمونه هایی از عنــوان چند تابع :

```
1)FUNCTION Menu : Integer ;
2)FUNCTION Test (r : Real):Real ;
3)FUNCTION Test1 (x,y : Integer):Integer;
4)FUNCTION Test2 (x:integer ; VAR i:Integer):Boolean;
```

در **تابع اول** نام تابع Menu می باشد و خروجی آن ازنوع Integer است.یعنی یک عدد صحیح را بر می گرداند.**تابع دوم** یک پـارامتر ورودی اعشاری بنام r دارد و خروجی آن از نوع Real است.یعنی یک عدد اعشاری وارد آن می شود و یـک عـدد اعـشاری از آن خـارج می گردد.**تابع سوم** دو پارامتر ورودی صحیح بنامهـای yوx دارد،یعنـی دو عـدد صـحیح وارد آن مـی شـود و خروجـی آن نیـز از نـوع Integer است و یک عدد صحیح را بر می گرداند. **تابع چهارم** یک پارامتر ورودی صـحیح (x) و دو خروجـی دارد کـه یکـی از طریـق پارامتر i است وازطریق آن یک عدد صحیح خارج می گردد و دیگری از نوع بولی است که یکـی از دو مقـدار True یـا False بـر مـی گردد.

نکته) نام تابع از قوانین نام گذاری شناسه های غیراستاندارد پیروی می کند.

نکته) نام تابع مانند یک متغیر محلی در طول اجرای تابع می تواند استفاده شود. **معمولا در داخل بلاک تابع، حداقل یک دسـتور انتساب به صورت زیر بایـد باشد. زیرا مقداری را که تابع بر می گرداند در نام تابع ریخته می شود.**

; مقدار خروجی =: نام تابع

در مثال روبرو **OE یک تابع است** که دارای یـک پـارامتر ورودی از نـوع مجـازی مقدار(Value) است و خروجی آن از نوع Boolean است. متغیـر Z یـک متغیـر محلی است و متغیر A یک متغیر سراسری است و همه بحث هـایی کـه در مـورد متغیرهای محلی و سراسری داشتیم اینجا نیز مطرح است.

در مثال روبرو نام تـابعOE اسـت کـه در طـول اجـرای تـابع یـک متغیـر ازنـوع Boolean است و حداقل یکبار در طول اجرای تـابع بایـد مقـداردهی شـود زیـرا مقداری را که تابع برمی گرداند در OE باید ذخیره شود.

نکته) چون تابع یک مقداری را بر می گرداند نحوهٔ صدا زدن (فراخوانی کردن) تابع با پروسیجر متفاوت است. تابع می تواند در یک عبارت مانند یـک متغیـر استفاده شود یا در دستور خروجی قرار گیرد. به طور کلی به تنهایی فراخوانی نمی شود.

```
Var   A:Integer;
Function   OE(F:integer):Boolean;
 Var
    Z:Real;
  Begin
      If  (F  Mod 2)=0  then
           OE:=False
        Else
           OE:=True;
      Z:=14.6;
   End;
Begin
    A:=19;
    Writeln(OE(A));
     Write('*');
End.
```

مثال) برنامه ای بنویسید که یک عدد از ورودی دریافت کرده سپس توسط تابعی بنام Fact فاکتوریل عـدد را محاسـبه و در برنامه اصلی چاپ نماید.

```
Var
   N:integer;
Function     Fact (M:integer) : longint;
  Var
P,I:integer;
 Begin
   P:=1;
   For I:=1   to    M  do
         P:=P*I;
   Fact:=P;
 End;
Begin
      Readln(N);
      Writeln('Factorial is = ' , Fact(N) );
      Readln
End.
```

تفـاوت های پردازه و تابع:

۱- خط عنوان پردازه با Procedure شروع می شود اما تابع با Function شروع می شود.

۲- در انتهای خط عنوان تابع، نوع دادهٔ خروجی (برگشتی) تابع مشخص می شود.

۳-هر تابع حداقل یک خروجی دارد اما پردازه می تواند دادهٔ خروجی نداشته باشد. (دادهٔ خروجی در پردازه یعنی پارامتر مجازی از نوع مرجع)

۴-در بلاک هر تابع **حداقل یک دستور برای ریختن مقدار خروجی** در نام تابع وجود دارد.

۵-نحوهٔ فراخوانی تابع با پردازه متفاوت است.

مثال) خروجی برنامه روبرو کدام است؟

```
Const  K:Integer=5;
Function  S(Var  A:Integer):Longint;
  Var       C:Char;
  Begin
       S:=A*A;
       A:=A+3;
       K:=K+2;
       Write(K,A);
  End;
Begin
     Writeln(S(K));    Write(K);
End.
```

مثال) خروجی برنامه روبرو کدام است؟

الف) 24 ب)19 ج)9 د)20

```
Const   X:Byte=3;   I=2;
Function  F(Var X:byte ; I:Byte):Byte;
Begin
   X:=X*2+I;
   F:=X;
End;
Begin
  X:=X+F(X,I)+X;
   Write(X);
End.
```

نکته) در عبارت X:=X+F(X,I)+X; هر عملگری که اولویت اول دارد انجام می شود و هر عملگری که الویت داشته باشد ابتدا عملوند سمت چپ آن محاسبه می شود سپس عملوند سمت راست آن.

در مثال داده شده ابتدا X+F(X,I) محاسبه می شود که X سمت چپ برابر 3 است و حال باید F(X,I) را محاسبه کرد.

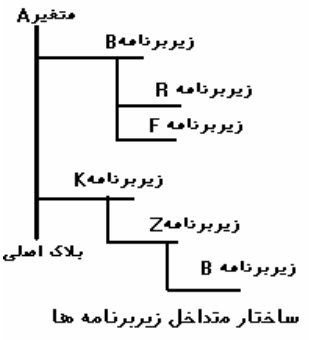
زیربرنامه های متداخل:

گاهی هنگام تعریف یک پروسیجر یا یک تابع در قسمت تعاریف آن ممکن است تابع یا پروسیجر دیگری تعریف شود که در این صـورت زیربرنامه های متداخل تشکیل می شوند. **مثال**:

```
Var  M:Integer;
Procedure   AZ(F:Integer);
  Var  K:Char;
   Function  HH():Char;
     Var  I:Integer;
       Begin
            I:=Random(26);
             HH:=CHR(I+65);
        End;
   Begin
       Write(F);
       Write(HH());
       K:=#68;
    End;
Begin
     M:=14;
     AZ(M);
     Write('*');
End.
```

در مثال روبرو تابع HH خود قسمتی از پروسیجر AZ است به همین دلیل فقـط در پروسیجر AZ می توان ازآن استفاده کرد (آن را صدا زد).
متغیر I یک متغیر محلی برای تابع HH است و متغیر K یک متغیر محلـی اسـت بـرای AZ و HH . یعنـی در هـر دو زیربرنامـه HH و AZ مـی توانـد دسترسـی داشته باشد.
متغیر M یک متغیر سراسری است.
اصطلا حا به محدودیت های دسترسی زیربرنامه ها و متغیر ها **Scope** می گویند.
که این Scope را می توان به رابطه پدر و فرزندی تشبیه کرد.
پروسیجر AZ فرزنـدی از بـلاک اصـلی اسـت و متغیـر M چـون در بـالای همـه زیربرنامه ها و بلاک اصلی تعریف شده است مربوط به همه است و تابع HH فرزند پروسیجر AZ است به همین دلیل AZ می تواند HH را فراخوانی کند و ....

نکته) اگر ساختار یک زیربرنامه را به صورت یک درخت (پدر و فرزندی) درنظر بگیریم آنگاه هر پدری می تواند فرزند خود را صدا بزند و هر زیربرنامه ایی (به غیر از بلاک اصلی) می تواند خودش را فراخوانی کند(توابع بازگشتی)
و هر زیربرنامه می تواند زیربرنامهٔ برادر بالای خود را صدا بزند!
مثلا زیربرنامهٔ F می تواند خودش و R را صدا بزند.
یا زیربرنامهٔ K می تواند زیر برنامهٔ Z و B و خودش را صدا بزند.
برای بقیه زیربرنامه ها خودتان پیدا کنید!



ساختار متداخل زیربرنامه ها

نکته) همان طور که گفتیم در صورتی می توان از یک زیر برنامه در زیربرنامه ای دیگر استفاده کرد که **قبـلا تعریف شـده باشـد** .در توربو پاسکال نقص فوق به کمک **اعلان forward** قابل حل است. بدین صورت که اگر زیربرنامه ای هنگام تعریف با اعلان forward همراه باشد،**بدون رعایت از پیش تعریف شدن** می تواند ، در زیربرنامه های دیگر ظاهر شود.

شکل اعلان به روش forward بصورت زیر می باشد :

Procedure Name (پارامترها) ; forward ;

نکته) forward کلمه ذخیره شده است.

مثال :

همان طور که مشاهده می کنید در مثال روبرو با اعلان forward بـرای پروسـیجرp2 مـی توانیم در داخل بلاک پروسیجر p1 پروسیجر p2 را فراخوانی کنیم،یعنی در حالـت عـادی و بدون اعلان forward نمـی توانـستیم از داخـل پروسـیجر p1 پروسـیجر p2را فراخـوانی کنیم.

```
Procedure p2 ; forward ;
Procedure p1;
Begin
      P2;
End;
Procedure p2;
Begin
End;
```

زیربرنامه های بازگشتی: (Recursive)

هنگامی که یک زیربرنامه خودش را فراخوانی می کند یک زیربرنامه بازگشتی ایجاد می شود. معمولا بعضی مسائل ماهیتی بازگشتی دارند یعنی یک عمل انجام می شود اما بر روی داده های مختلفی، مانند محاسبهٔ فاکتوریل یک عدد!6*7=!7

استفاده از زیربرنامه ای بازگشتی دو عیب دارند:

۱- حافظه ای که برای نگهداری پارامترها در هر مرحله فراخوانی، مصرف می کنند(Stack)

۲- به دلیل فراخوانیهای مکرر، اجرای زیربرنامه های بازگشتی کند است.

چگونه متوجه شویم که یک زیربرنامه بازگشتی هست یا نه؟ اگر در بدنهٔ یک زیربرنامه **خود زیربرنامه** صدا زده شده باشد **بازگشتی** است؛ همچنین در اکثر زیر برنامه های بازگشتی از If Then – Else استفاده می شود.

برای اینکه عملکرد یک زیربرنامهٔ بازگشتی را متوجه شویم راهی بهتر از رسم Stack آن نیست.

که روش رسم Stack در کلاس بحث خواهد شد.

```
Function  K(n:integer):Integer;
Begin
   If   n=1  then
        K:=1
   Else
     K:= n * K(n-1);
End;
Begin
   Writeln(K(5));
End.
```

تست) خروجی برنامهٔ روبرو کدام است؟

الف) 720 ب)120 ج)24 د)0

مثال) خروجی برنامه روبرو کدام است؟

```
Function   M(D:Integer):Integer;
 Begin
    If  D=1 then
         M:=1
     Else
          M:= 3+ M(D  div 2);
End;
Begin
  Writeln(M(40));
End.
```

نکته) توابع بازگشتی به نحوی نوشته می شوند که در یک مرحله ، دیگرکار تکرار نشود درغیر اینصورت برنامه تا بی نهایت تکرار می شود و حافظه پر می شود و پیغام می دهد.

```
Function   Z(x:Word):Word;
Begin
   If   X<=1 then
    Z:=1
   Else
        Z:=Z(x-2) + Z(x  Div  2);
End;
```

تست) اگر در ابتدا X:=12; باشد خروجی تابع روبرو چیست؟

الف)10 ب)20 ج)22 د)30

```
Function OP(x,y:integer):integer;
Begin
   If  X=Y   then
      OP:=0
      Else
           If  X>y  then
               OP:=OP(X-Y,Y)
            Else
                OP:=X;
End;
```

تست) تابع روبرو چه عملی را انجام می دهد؟ x,y>=1

الف) X/Y ب) X div Y

ج) X mod Y د) X*Y

برای درک بهتر زیربرنامه های بازگشتی به توضیحاتی که در زیر برای تابع فاکتوریل آورده شده است توجه کنید :

تابع فاکتوریل بر روی مجموعه اعداد صحیح مثبت ، به دو صورت زیر تعریف می شود :

تعریف اول :

$$N ! = 1 \times 2 \times 3 \times \ldots \times (N-1) \times N$$

در مثال صفحه ۵۸ برای بدست آوردن فاکتوریل یک عدد از طریق تابع **Fact** از تعریف اول تابع فاکتوریل استفاده کرده ایم.

تعـــریف دوم :

$$N ! = \begin{cases} N \times (N-1)! & N \geq 2 \\ 1 & N = 0 \text{ یا } N = 1 \end{cases}$$

طریقه محاسبه 3 ! **با توجه به تعریف دوم در جدول زیر مرحله به مرحله نشان داده شده است.**

1) $3 ! = 3 \times 2 !$	پاسخ = 6
2) $2 ! = 2 \times 1 !$	پاسخ = 2
3) $1 ! = 1$	پاسخ = 1

نتیجه محاسبه 3! **با استفاده از هر یک از دو روش بالا یکسان و برابر** 6 **است .در تعریف دوم فاکتوریـل ، از خـود تـابع فاکتوریل ، برای تعریف آن استفاده شده است بنابراین تعریف دوم را تعریـف بازگـشتی فاکتوریـل و تعریـف اول را تعریف غیر بازگشتی و یا تکراری (بدلیل تکرار دستور ضرب) فاکتوریل می نامند.**

مثال) تابع فاکتـوریل با استفاده از تعریف بازگشـتی

```
FUNCTION    RFactorial (n:Integer): LongInt;
BEGIN
     IF   N ≥  2   THEN
            RFactorial := N  *   RFactorial ( N – 1 )
     ELSE
                RFactorial := 1
END;
```

توجه داشته باشید که :

نحوه فراخوانی زیر برنامه های بازگشتی ، تفاوتی با نحوه فراخوانی زیر برنامه های غیر بازگشتی ندارد. بر فـرض اینکـه تابع **RFactorial** ، به صورت **I := RFactorial (3) ;** در یک برنامه فراخوانی شده است در هنگام اجرای این دستور رایانه عدد ۳ را در پارامتر n کپی می کند و سپس با اجرای دستورهای درون تابع حاصل را درون متغیر I می ریزد.

تست:

۱-تفاوت تابع و پروسیجر در چیست؟

الف) تابع مقدار بازگشتی دارد ولی پروسیجر ندارد ب) پروسیجر مقدار بازگشتی دارد ولی تابع ندارد

ج)تابع و پروسیجر هر دو مقدار بازگشتی دارند اما تابع همیشه یک خروجی را دارد

د) هیچ تفاوتی ندارند.

```
Function  test(n:byte):integer;
 Begin
     If  n=1  then
      Test:=6
     Else
        Test:=6+test(n-1);
End;
```

۲-خروجی تابع روبرو در ازای ورودی N=20 چند است؟

الف)120 ب)60 ج)100 د)140

۳-کدام گزینه صحیح است؟

الف) تابع و پروسیجر را می توان در عبارات ریاضی، مقایسه ایی و غیره به کار برد

ب)درون بدنه تابع نام تابع یک متغیر محلی است.

ج) در خارج از بدنهٔ تابع، می توان نام تابع را در سمت چپ عملگر انتساب(=:) به کار برد.

د) پردازه نمی تواند خودش را فراخوانی کند.

```
Function  F(x,y:integer):integer;
 Var  P,I:integer;
Begin
    P:=1;
    For  I:=1  to  Y  Do
      P:=P*X;
    F:=P;
End;
```

۴-تابع روبرو چه عملی را انجام می دهد؟

الف) مقدار X را در Y ضرب می کند.

ب)X را به توان Y می رساند

ج) Y را به توان X می رساند

د)X را به توان P می رساند.

۵-در برنامه روبرو چند خطا وجود دارد؟

الف) 2

ب)3

ج)4

د)5

```
Const ADDB=20;
Var Z:integer;
Function ZARB(a,b:Integer);
Begin
   H:=H+2;
   Zarb:=a*b;
End;
Var   H:real;
Begin
   Z:=15; H:=Z;
   ADDB:=ZARB(Z,H);
End.
```

۶-کدام گزینه در مورد زیربرنامه ها و توابع در زبان پاسکال صحیح است؟

الف)پارامترهای زیربرنامه ها و توابع فقط ورودی هستند ب) پارامترها توابع ورودی یا خروجی هستند

ج) تابع را می توان به عنوان پارامتر برای زیربرنامه یا تابع دیگر بکاربرد

د) زیربرنامه را می توان به عنوان پارامتر برای زیربرنامه یا تابع دیگر بکاربرد.

۷- خروجی زیربرنامه روبرو برای ورودی 5 کدام است؟

```
Function  KK(n:integer):integer;
Begin
   If  n=1  then  KK:=2
    Else
         KK:=2*n+KK(n-1);
End;
```

الف) 24 ب)28 ج)29 د)30

۸-کدامیک از گزینه های زیر نادرست است؟

الف) به زیربرنامه و تابع Module می گویند ب) پارامترهای زیربرنامه بعضی ورودی و بعضی خروجی هستند.

ج)تابع حداقل دارای یک خروجی است د) فراخوانی یک پروسیجر حتما باید در یک دستور انتساب یا خروجی صورت گیرد

۹-خروجی برنامه روبرو کدام است؟ از چپ به راست

الف) 13 13 13 ب) 3 13 3

ج) 3 13 3 د) 3 3 3

```
Var   a:integer;
Procedure  P1(x:integer);
  Var  a:integer;
Begin
     A:=10+X;
      Write(A);
End;
Begin
  A:=3;
  P1(A);
  Write(A);
End.
```

۱۰- خروجی قطعه برنامه روبرو کدام است؟

الف) 18 22 ب)22 22

ج)24 22 د)14 14

```
Procedure  T(Var X:byte ; I:byte);
Begin
    I:=X+4;
    X:=I+4;
End;
Const  I:byte=14;
Begin
   T(I,I);
    Writeln(I,I);
End.
```

۱۱-کدام گزینه صحیح نیست؟

الف) متغیرهای سراسری در همه جای برنامه قابل دسترسی هستند. ب) نام تابع حداقل یک بار باید در بدنه تابع مقدار دهی شود.

ج)تابع بازگشتی درون بلوک خودش فراخوانی می شود. د)متغیرهای محلی مربوط به یک تابع در برنامه اصلی قابل دسترسی هستند.

۱۲-خروجی برنامه روبرو به ترتیب از چپ به راست کدام است؟

الف)10,26,36,5,25

ب) 30,6,36,6,25

ج)10,26,36,5,26

د)30,6,36,5,26

```
Function  S(A:integer;VAR X:integer):integer;
  Begin
     A:=A+5;
     X:=X+1;
      Write(A,X);
     S:=X+A;
End;
Begin
    A:=25;   X:=5;
    Writeln(S(X,A),X,A));
End.
```

۱۳-تفاوت پارامتر Value (ارزشی) و variable (مرجع) در چیست؟

الف)پارامتر Value می تواند در مواقعی هم ورودی و هم خروجی باشد درصورتیکه پارامتر Variable فقط خروجی است.

ب)پارامتر Variable تنها می تواند خروجی باشد.

ج)پارامتر Variable در مواقعی هم ورودی و هم خروجی می باشد ولی Value همیشه ورودی است.

د)پارامتر Value و variable هم می تواند ورودی و هم خروجی باشند.

۱۴-تابع زیر چه عملی انجام می دهد. M,N>=1

الف)M+N ب)M*N ج)M^N د) M-N

```
Function  F(M,N:integer):integer;
  Begin
     If  N=1  then    F:=M
        Else
         F:=F(M,N-1)+M;
  End;
```

یونیـــت CRT :

در ادامه تعدادی از توابعی که در یونیت crt کاربرد بیشتری دارند بحث می شود. برای اینکه بتوانیم از ایـن توابـع اسـتفاده کنـیم بایـد **یونیت crt را در ابتدای برنامه Uses کنیم**.

پروسیجر GotoXY : از این تابع برای جابجا کردن مکان نما در صفحه خروجی استفاده می شود. صـفحه مـانیتور دارای ۲۵ سـطر و ۸۰ ستون است (در حالت متنی). مثلا دستور ;(74,15)GotoXy مکان نما را به ستون ۷۴ و سطر ۱۵ منتقل می کند.

نکته) دقت داشته باشید **که پارامتر اول ستون** است و **پارامتر دوم سطر** است. پارامتر اول **نمی تواند بیش از ۸۰** و پارامتر دوم نیز **نمی تواند بیش از ۲۵** باشد.

پروسیجر TEXTCOLOR : از این دستور **برای تغییر رنگ قلم در خروجی** استفاده می شـود. دارای یـک پـارامتر ورودی از نـوع صحیح است که شماره رنگ را مشخص می کند. مثلا **;TextColor(2)** باعث می شود از این دسـتور بـه بعـد نوشـته هـایی کـه در مانیتور چاپ می شوند رنگ سبز داشته باشند. **شماره رنگ عددی بین 0 تا 15 است**. هر رنگی دارای یک شماره است که جدول آن در کتاب وجود دارد. **برای بالا بردن خوانایی برنامه برای هر شماره ثابت آن نیز تعریف شده است. مثلا Textcolor(Green)** همان کار قبلی را انجام می دهد. مثلا **عدد ۱۴ برای رنگ زرد** است که می توانید از ثابت Yellow نیز بجای ۱۴ استفاده کنید.

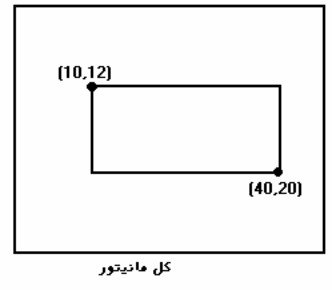
پروسیجر ClrScr : **این پروسیجر پارامتر ورودی ندارد و باعث می شود صفحه مانیتور پاک شود (مانند دستور Cls در Dos)**

پروسیجر TextBackGround :

این پروسیجر یک **پارامتر عددی صحیح بین 0 تا 7 دارد** که باعث می شود **رنگ پس زمینهٔ خروجی** تغییر کند. شماره رنگ عددی بین صفر تا هفت است. مثال;(Blue)TextBackGround یا ;(1)TextBackGround پس زمینه را آبی می کنند.

پروسیجر Window: از این پروسیجر برای **تعیین قسمتی از صفحه نمایش به عنوان پنجرهٔ فعال** استفاده می شود.

مثال: ;Window(10,12,40,20) باعث می شود پنجرهٔ فعال شما در یک مستطیل قرار گیرد.



تابع ReadKey نیز مربوط به یونیت Crt است که قبلا بررسی شد.

تعداد پروسیجرها وتوابع موجود در یونیت Crt زیاد است که تعدادی دیگر آن در پاسکال ۲ بحث خواهد شد.

تست: از پروسیجر ..... برای تعیین رنگ کاراکترها استفاده می شود.

الف) Color ب)TextBackGround ج)TextColor د)CharColor

تست :

۱- در محیط توربو پاسکال برای آنکه پنجره ی دیگری را به عنوان پنجره ی جاری انتخاب کنیم از کدام کلید استفاده می شود؟

الف) F4 ب)F6 ج)F5 د)F7

۲- جهت اجرای خط به خط برنامه در محیط پاسکال از کدام کلید استفاده می شود؟

الف)F7 ب)F6 ج)F5 د)F9

۳- کدام گزینه در مورد زیربرنامه استاندارد Upcase صحیح است؟

الف) یک رشته را به عنوان پارامتر ورودی دریافت و حروف آن را به بزرگ تبدیل می کند

ب) یک کاراکتر را به عنوان پارامتر ورودی دریافت و حرف بزرگ آن را برمی گرداند.

ج) اگر در رشته ی ورودی کاراکتری غیر از حروف الفبای کوچک باشد آنرا به رشته تهی تبدیل می کند.

د) یک کاراکتر یا رشته را به عنوان پارامتر ورودی دریافت کرده و آن را به حرف یا حروف بزرگ تبدیل می کند.

۴-برای درج توضیحات در زبان پاسکال از کدام گزینه استفاده می شود؟

الف) [ ] ب) ( ) ج) { } د) < >

۵-متغیر نوع word چه محدوده ای از اعداد را در خود ذخیره می کند و چند بایت فضا اشغال می کند؟

الف)32767+ تا 32768– و ۲ بایت ب)65535 تا 0 و ۴ بایت ج)65535 تا 0 و ۲ بایت د)128- تا127+ و۲ بایت

۶-خروجی دستور **Write(ODD($D) > ODD(16))** کدام است؟

الف) TRUE ب) 1 ج) 0 د)False

۷- ارزش بولی عبارت زیر به ازای X=3 و Y=4 چیست؟ **(X-Y<10) and (X+3*Y/4=5)**

الف) False ب)True ج)1 د)0

۸-کدام یونیت در پاسکال بصورت استاندارد تعریف می شود؟

الف)Dos ب) System ج)Crt د)Graph

```
Var
I,J:integer;
A,B,C : Char;
M,F,E:Real;
```

۹-از گزینه های زیر با توجه به تعاریف روبروکدام صحیح است؟

الف) For I:=A To B Do ج)For M:=F To E Do

ب)For A:=B To C Do د)For I:=10 To F Do

۱۰- حاصل عبارت روبرو چیست؟ 6+4 Mod 2*6-17

الف)-11 ب)-17 ج)1 د)13

۱۱-با توجه به دستور روبرو کدام گزینه صحیح است؟

```
For Y:=Trunc(X) To Round(X) Do
 Write('*');
```

الف)حلقه حداقل یک بار تکرار می شود ج) پیغام خطا صادر می شود

ب)حلقه دو بار تکرار می شود د) حلقه اصلا تکرار نمی شود

۱۲-اگر A=5 باشد خروجی دستورات روبرو کدام است؟

```
IF A>5 Then Else A:=A*2;
IF A>=5 Then A:=A*2 Else;
Write(A);
```

الف)5 ب)10 ج)20 د)40

۱۳-اگر کاراکتر های 1 و 2 و 3 را در ورودی با Enter از هم جدا کنیم خروجی کدام است؟(علامت ـ نشان دهنده محل کرسور درخروجی است)

```
Var A,B,C,D:Char;
Begin
 Readln(A,B,C,D);
Write(A,B,D);
End.
```

الف)1 ب)_2 ج)_123 د)123

-

-

۱۴-در زبان پاسکال IDE یعنی .....

الف) قسمتی از برنامه ب) ویرایش کردن برنامه ج)محیط برنامه د)هر سه مورد

۱۵-برای اجرای محیط برنامه نویسی در خط فرمان Dos کلمه ..... را می نویسیم.

الف)Turbo ب)Pascal ج)Tpascal د)Tp

۱۶-برنامه های پاسکال با پسوند ..... ذخیره می شوند.

الف)TPS ب)PAS ج)TPU د)TBO

۱۷-کمیت بولی چه مقادیری را می تواند به خود اختصاص دهد؟

الف)False-True ب)No-Yes ج)مقادیر عددی د)0 – 1

۱۸-خروجی برنامه مقابل چیست؟

```
Var A:Real;
Begin
A:=10.5;
Writeln(A);
End.
```

الف)1.05 ب)105

ج)10.5 د) 1.0500000000E+1

۱۹-تعاریف ..... باید قبل از شروع بلوک کد پروسیجر یا توابع قرار داده شوند.

الف)اصلی ب)فرعی ج)محلی د)سراسری

۲۰-برای دسترسی به پروسیجرها و توابع که صفحه کلید و صفحه نمایش را کنترل می کنند باید از کدام یونیت استاندارد پاسکال استفاده نمود.

الف)Crt ب)Dos ج)System د)Printer

۲۱-درپاسکال برنامه های بزرگ را می توان به تکه های کوچکتری تقسیم کرد. به این تکه برنامه ها ..... می گویند.

الف)روتین ها یا سابروتین ها ب) پروسیجرهایا توابع ج) توابع یا روتین ها د) پروسیجرها یا روتین ها

۲۲-از این تابع برای دریافت لگاریتم طبیعی عدد استفاده می شود؟

```
Var A,B,C:String[3];
Begin
A:='20';  B:='30';
C:=A+B;
Writeln(c);
End.
```

الف)Length ب)Lo ج)Ln د)INT

۲۳-خروجی برنامه مقابل چیست؟

الف)203 ب)20+30 ج)50 د)2030

۲۴-اگر A=5 باشد خروجی دستور روبرو کدام است؟

```
IF  A<5 Then  Else  A:=A*2;
IF A>=5  Then  A:=A*2 Else ;
Write(A);
```

الف) 5 ب)10 ج)30 د)40

۲۵-با توجه به ساختار زبان پاسکال اجرای کدام گزینه باعث خطا می شود؟

الف)
```
Begin


End;
```

ب)
```
Begin ;
End.
Var
X:integer;
Readln(x);
```

ج)
```
Begin

End.
```

د)
```
Begin ;

  Begin ; End;
End.
```

۲۶-خروجی برنامه ی روبروکدام است؟

```
A:=1;  B:=2;
For C:=A  To  B  Do
   Write(B,A,C);
```

الف) 111222 ج)211212

ب) 112212 د)211222

۲۷-اگر در برنامه های زیر برای A مقدار 12 وارد شود کدام خط در Loop می افتد؟

```
Readln(A);
Repeat ; Until  A>10;        {1}
For  B:=1  To  A  Do ;       {2}
While A>0  Do  ;  A:=A-1;  {3}
```

الف) 1 و 2 و 3 ب) 1

ج)2 د)3

انواع داده ها در پاسکال :

- ۱- ساده
  - ۱- اسکالر
    - صحیح به غیر از comp
    - کاراکتری(Char)
    - بولین(Boolean)
    - زیرقلمرو(Sub Range)
    - شمارشی(Enument)
  - ۲- اعشاری و Comp
- 2 - ساختاری
  - 1 - رشته ا ی
  - 2 - آرایه ای
  - 3- رکورد
  - 4 - اشاره گر (فایلها)
  - 5 - پردازه ای

نکته: نوع داده ساده نوعی است که قابل تقسیم به نوع داده دیگری نیست و خود به دو گروه **اسکالر و اعشاری** تقسیم می شود.

نکته: نوع اسکالر نوع داده ساده ای است که دارای ترتیب وشمارش پذیر باشد وهر عضو آن دارای مقدار قبلی وبعدی است.البته به جز اولین مقدار که قبلی ندارد وآخرین مقدار که بعدی ندارد.

نکته: اولین عضو هر مجموعه به جز مجموعه اعداد علامت دار دارای شماره ترتیب صفر است.(مربوط به تابع ORD)

نوع داده زیر قلمرو(زیربازه):

زیر قلمرو **محدوده ای از مقادیر یک نوع داده اسکالر** به نام میزبان می باشد.متغیرهایی از نوع زیرقلمرو ،تمام خصوصیات متغیری از نوع میزبان را دارند و به اندازه نوع میزبان نیز ، حافظه اشغال می کنند.

در تعریف نوع زیر قلمرو کافی است اولین(کوچکترین) و آخرین(بزرگترین) مقدار مجموعه را معین کنید تا به این طریق ، محدوده بین آن دو به عنوان نوع زیر قلمرو در نظر گرفته شود.

برای تعریف یک متغیر زیر قلمرو ۲ روش وجود دارد:

الف)در روش اول در قسمت VAR به روش زیر عمل می کنیم:

```
VAR
        نام متغیر : ثابت اول .. ثابت دوم ;
```

- **ثابت اول همواره باید کوچکتر یا مساوی ثابت دوم باشد.**
- **ثابت اول و ثابت دوم باید اسکالر باشد.**
- **ثابت اول ثابت دوم می توانند ثابت بدون نوع باشند.**

ب) در روش دوم تعریف زیر قلمرو ابتدا در قسمت Type نوع زیر قلمرو را تعریف می کنیم سپس در قسمت VAR متغیر را از همان نوع تعریف می کنیم. به صورت زیر:

```
Type
        نام نوع = ثابت اول .. ثابت دوم ;
Var
        نام متغیر : نام نوع ;
```

در پاسکال می **توان در قسمت تعاریف در Type نوع خاصی را تعریف کرد** مثلا می توان نوع داده های تعریف شده را تغییر نام داد.
مثال)

```
Type
        Myint=integer;
Var
        A: myint ;
```

در قسمت Type می توان نوع زیر قلمرو، شمارشی، آرایه رکورد نیز تعریف کرد.

میـــزبان:

هر نوع زیرقلمرویی دارای یک میزبان است این میزبان توسط مترجم پاسکال تعیین می شود به طوری که محدوده را شـامل شـود وکمترین فضا را اشغال کند و نیز اسکالر باشد.(اگر دو نوع فضای یکسانی دارند و محـدوده را نیـز شـامل مـی شـوند پاسـکال نـوعی را انتخاب می کند که منفی ها را نیز داشته باشد.)
مثال)

```
Const
  H=20;
Type
  Sub=0 .. H;
Var
  A: sub ;
  Z:integer;
Begin
  A:=25;          (constant out of range)  خطای زمان ترجمه
  Z:=40;
  A:=Z;
  A:=A+25;
  Readln(A);
  Writeln(A);
End.
```

برنامه نوشته شده رادر دوحالت بررسی می کنیم ابتدا در حالتی که برنامه,در حال **ترجمه** است یعنی دستورات اجرا **نمـی شـوند** بلکه ترجمه می شوند. خط A:=25; غلط است زیرا در حالت ترجمه نمی توان **یک ثابت** را که از محدودهٔ زیـر قلمـرو خـارج اسـت را داخل متغیر A ریخت. در این صورت پیغام خطای constant out of range ظاهر می شود. اما در **لحظـه ترجمـه** دسـتور A:=Z; خطا نیست **زیرا در لحظه ترجمه مقدار دهی انجام نمی شود**. حال برنامه را در حالت **اجرا** بررسی می کنیم: هنگامی که برنامه اجرا می شود متغیر های از نوع زیر قلمرو خصوصیات میزبانشان را پیدا می کنند مثلا متغیر A در برنامه ذکر شـده در حالـت اجـرا ماننـد یـک متغیر shortint عمل می کند بنابراین دستور A:=Z; خطا نیست و داخل متغیر A عـدد ۴۰ ریختـه مـی شـود و اگـر بـرای دسـتور readln(A) عددی خارج از محدوده وارد کنیم **بازهم خطا صورت نمی گیرد بلکه سرریزی رخ می دهد.(سرریزی خطا نیست)**
با این توضیحات هیچ محدوده ای برای متغیرهای زیرقلمرو کنترل نمی شود برای رفع این مشکل **از رهنمود {$R+}** استفاده می کنیم **این رهنمود باعث می شود محدودهٔ متغیرها کنترل شود**.یعنی اگر از محدودهٔ زیر قلمرو خارج شـدیم پیغـام خطـا صـادر مـی شود.**(در زمان اجرا)** حتی اگر سرریزی نیز رخ دهد باز هم خطا خواهد گرفت.
نکته) دوتابع استاندارد low و high کمترین وبیشترین مقدار یک نوع داده اسکالر را برمی گردانند.

کاربرد زیرقلمـــرو :

زیرقلمروها را بیشتر برای کنترل محدوده متغیرها به کار می برند و دارای مزایای زیر است:
۱-کنترل محدودهٔ متغیر ها وصدور پیغام خطا توسط مترجم
۲- چون نوع میزبان را پاسکال انتخاب می کند کمترین فضا را اشغال می کند ودر مصرف حافظه صرف جویی می شود.
۳- زیرقلمروها شناسه های با معنی هستند وخوانایی برنامه را بالا می برند.

نـوع داده شمارشی:

برنامه نویس می تواند با توجه به موضوع برنامه **تعدادی شناسه**(نه رشته ، نه عددو نه ثابت) را در یک نوع شمارشی قرار دهد.

مثال)

```
Type
  EnuA=(sat , sun , mon , tue);
Var
  A:EnuA;
Begin
  A:=sun;
End.
```

- نکته: شناسه هایی که برنامه نویس تعین می کند باید تابع نام گذاری شناسه های غیر استاندارد باشد.
- نکته: متغیر های نوع شمارشی **همیشه یک بایت فضا** اشغال می کنند.
- نکته: **تعداد** شناسه های نوع **داده شمارشی حداکثر ۲۵۶ تا** است.
- نکته: برای نامگذاری متغیرهای شمارشی معمولا پیشوند Enu به کار می رود.
- نکته: شناسه های موجود در نوع داده شمارشی نباید تکراری باشند

```
type
    enuDAY=(sat,sun,sat,mon);   خطاست
```

- شناسه های موجود در نوع شمارشی به ترتیب دارای رتبه های ۰ و ۱ و...و ۲۵۵ است.

مثال)

```
Type
  EnuB=(red,green,blue,black);
Var
  X,Z:EnuB;
  A:integer;
Begin
  z:=blue;     ž   صحیح-زیرا در متغیر نوع شمارشی فقط می توان شناسه های مربوط به خودش را قرار داد
  z:='black';  ž   غلط- زیرا رشته است
  z:=1;        ž   غلط
  a:=0;        ž   صحیح
  z:=a;        ž   غلط- عمل انتساب در صورتی صحیح است که هر دو متغیر هم نوع باشند
  a:=z;        ž   غلط
  z:=x;        ž   صحیح
  z:=pred(blue);  ž  صحیح
  a:=ord(z);   ž   صحیح
  writeln(z);  ž   غلط- متغیرهای شمارشی وشناسه های آنها را نه می توان چاپ کرد ونه میتوان دریافت کرد
  writeln(blue);  ž  غلط
  readln(z);   ž   غلط
  write(ord(z)); ž  صحیح- رتبه متغیر های شمارشی را می توان چاپ کرد
  writeln(succ(green)); ž  غلط
end.
```

کاربـرد ! نوع داده شمارشی :

داده های نوع شمارشی فقـط برای افزایش خوانایی برنامه استفاده می شوند.

مجموعـه ها (Sets) :

تعداد محدودی از داده ها که از نظر نوع یکسان هستند ، در قالب مجموعه نگهـداری مـی شـوند کـه مفهـومی شـبیه مجموعـه هـا در ریاضیات دارند.

مجموعه متغیری است که شامل لیستی از اعداد صحیح ، کاراکترها و بولین و بهتر است بگوییم کـلا داده هـای اسـکالر مـی باشـد کـه دارای تعداد عناصر محدود به حداکثر ۲۵۶ تا می باشد.

مجموعه از این جهت که شامل داده هایی از یک نوع می باشد شبیه به یک آرایه می باشد ولی آرایه دارای عناصر محدودی نیست و در ضمن مانند آرایه تعریف نمی شود.

تعـریف مجموعه :

برای تعریف یک مجموعه از کلمات کلیدی **Set of** استفاده می کنیم که در زیر نمونه هایی از تعـاریف مجموعـه هـای مختلـف آمـده است :

```
Var
S:= set of 0..10 ;
C:= set of char ;
B := set of boolean ;
G:= array [1..5] of set of 1..10 ;
```

1	2	3	4	5
مجموعه	مجموعه	مجموعه	مجموعه	مجموعه

حال با توجه به تعاریف بالا در داخل یک برنامه می توانیم **مقدار دهی** های زیر را داشته باشیم :

```
S := [2] ;
S := [2,1,4,5,2] ;
S := [1..4 , 8] ;
C := ['a' , 'b' , 'c' , 'd'] ;
G[1] := [1,2,3] ;
G[2] := [] ;  مجموعه تهی
```

نکته) مجموعه تهی مجموعه ای است که هیچ عضوی ندارد و با [] نشان داده می شود.

نکته) تکرار عضو در مجموعه ها تاثیری ندارد و عضوهای تکراری فقط یک عضو در نظر گرفته می شوند.

نکته) جابجایی عضوها در یک مجموعه تاثیری ندارد.یعنی دو مجموعه ای که اعضایشان یکـسان اسـت ولـی جابجـا هـستند مـساوی هستند با هم .

تعاریف زیر برای مجموعه ها درست هستند :

```
Var
Ch : set of char ;
Bool : set of boolean ;
Num : set of byte ;
X : set of 100..200 ;
Y : set of 'a'..'z' ;
Z : set of  '0'..'9'  ;
```

تعاریف زیر برای مجموعه ها نادرست هستند :

```
Var
R : set of real ;
I : set of integer ;
S : set of string ;
X : set of 6..0 ;
```

عملیات روی مجموعـه ها :

عمل عضویت متغیر در مجموعه در زبان پاسکال با کلمه ی کلیدی In صورت می گیرد. اگر عضویت صحیح باشد یعنی آن متغیر در مجموعه وجود داشته باشد جواب True و در غیر اینصورت False می باشد.

همچنین دو مجموعه را می توان با علامت های شرطی = **و** <> **و** => **و** =< مقایسه کرد که همگی دارای خروجـی درسـت یـا غلـط می باشند . **ولی علامت های < , > در مجموعه ها کاربردی ندارند**.

```
Var
    A: set of byte ;
Begin
   A := [1..3,4] ;
   If  3 in a   then
      Write('*')
   Else
        Write ('**');
   Write ( 3 in a ) ;                جواب: True
   Write ( 8 in a ) ;                جواب: False
   Write ( [1,2,3,4] = a ) ;         جواب: True
   Write ( [1,2] <=  [1,2,3] ) ;     جواب: True
   Write ( [] >=  [1] ) ;            جواب: False
End.
```

در ریاضیات می توانیم مجموعه ها را با هم **اجتماع ، اشتراک و تفاضل** کنیم که این عملیات در پاسکال به ترتیب با :

عملگرهای + و * و – می باشد.

اجتماع دو مجموعه : ترکیبی از همه عضوهای دو مجموعه

اشتراک دو مجموعه : عضوهایی که در هر دو مجموعه مشترک است.

تفاضل دو مجموعه : اعضایی که در مجموعه اول هست و در مجموعه دوم نیست.

```
Var
   a , c , d ,e : set of   byte ;
   b : set  of  0..10 ;
begin
    a:= [0,1,2,3] ;
    b:= [2,3,4,5] ;
    c:= a+b ;          c = [0,1,2,3,4,5]
    d:= a*b ;          d = [2,3]
    e := a – b ;       e = [0,1]
end .
```

نکته) پارامترهای ورودی پروسیجر می تواند مجموعه باشد بشرطی که قبلا در Type تعریـف شـده باشـند ، نـه اینکـه مـستقیما در پروسیجر بعنوان پارامتر بیاید.

نکته) خروجی یک تابع نمی تواند از نوع مجموعه باشد.

برای نوشتن یا خواندن مجموعه ها :

باید **عضو به عضو** عملیات صورت بگیرد و مستقیما Read و Write روی آنها کار نمی کند.

حال با توجه به نکاتی که گفتیم تعاریف زیر برای مجموعه ها غلط می باشد :

```
Function f1 ( var  ch1 : set of char ) : integer ;
Procedure   p1   ( ch2 : set of   0...9 ) ;

Type
Ch = set of char ;
Function f2  ( x : integer ) : ch ;
```

تعاریف زیر برای مجموعه ها درست می باشند :

```
Type
Ch = set of char ;
Function  f3  ( ch3 : ch) : char ;
Procedure p2 ( var  ch4 : ch ; ch5 : ch ) ;
```

آرایـــه ها :

آرایه تعدادی متغیر هم نوع است که تحت یک نام ذخیره می شوند.هرسلول آرایه یک متغیراندیس داراست. تعداد عناصرآرایه محدودوثابت است.عناصر آرایه از نظر فیزیکی مکان های متوالی درحافظه کامپیوتراشغال می کنند.

برای تعریف آرایه ابتدا طول (نوع اندیس) آرایه که در حقیقت تعداد خانه های آنرا مشخص می کند ، معین می گردد. سپس نوع خانه هایی که داده ها در آن قرار خواهند گرفت را تعیین می کنند و در نهایت ، عناصـر داخـل خانـه هـای آرایـه توسط اندیس آرایه که محل قرار گرفتن عنصر در آرایه را مشخص می کند ، قرار می گیرند.

می توانیم آرایه را بصورت زیر در نظر گرفت :

1	2	3	...	10
A[1]	A[2]	A[3]	...	A[10]

همان طور که ملاحظه می کنید داخل هر کدام از خانه های آرایه یک عنصر قرار می گیرد که نوع این عناصر باید با نوع آرایه یکسان باشد.

برای تعریف آرایه در پاسکال دو روش وجود دارد:

روش اول)

```
Var
  نام آرایه : array[نوع اندیس] OF  نوع محتوای آرایه  ;
```

روش دوم)

```
Type
  نام نوع = array[نوع اندیس] OF  نوع محتوای آرایه  ;
Var
  نام آرایه  :  نام نوع ;
```

در روش دوم ابتدا در قسمت Type نوع آرایه را تعریف می کنیم سپس در قسمت Var خود آرایه را تعریف می کنیم روش دوم بهتـر از روش اول است زیرا :

۱- خوانایی برنامه بالا می رود.

۲- اگر در جای دیگری از برنامه نیاز به آرایه ای از همین نوع داشته باشیم به سادگی می توانیم تعریف کیم.

نکته) **اندیسها** حداکثر می توانند **یک اسکالر** 2 **بایتی** باشند.

برای درک بهتر آرایه ها ، برای مثالهای زیر آرایــه را رسم کنید :

```
A:array[1..20]   of   integer;
```

```
B:array[-4..10]   of   real;
B[-4]:=200;        B[2]:=8+B[10];
```

```
c:array[byte] of  longint ;
```

ü نکته: نوع اندیس آرایه باید اسکالر باشد البته به غیر از **Longint** .

```
 D:array [ shortint ] of boolean;
```

```
E:array [ boolean ] of integer;
E[false]:=100;
E[E[5>8]<150]:=200;
```

E

False	True
100	200

```
H:array['A'..'Z'] of char;
H['B']:='c';        H[#65]:='M';       H['C']:=#65;
```

```
m:array[char] of 5..20;
m['!']:=12;         m[#35]:=14;        m[#2]:=3;
```
• خطا

```
Type
  EnuS=(red,blue,green,yellow,black);
  M=5..20;
Var
  n:array[m] of EnuS;
n[7]:=green;                n[7]:='green' ;
n[ord(yellow)+3]:=yellow ;      writeln( n[7] ) ;
```
• خطا

• خطا

```
TYPE
  COLOR=(RED,BLUE,YELLOW,GREEN);
  M=ARRAY[COLOR] OF INTEGER;
 VAR
  A:M;
  I:COLOR;
BEGIN
  FOR I:=RED  TO GREEN  DO
     READLN(A[I]);
end.
```

دسترسی به عناصر آرایه:

برای دسترسی به عناصر آرایه نام آرایه را نوشته سپس داخل کروشه اندیس را قرار می دهیم. **اندیس می تواند یک عبارت یا یک متغیر باشد به طوری که نوعش با نوع اندیس همخوانی داشته** باشد.

```
type
  myary=array[1..100] of real;
var
  a:myary;
  b:integer;
begin
  b:=105;
  a[-1]:=50;
  a[b]:=180;
  writeln( a[b] );
end.
```

اگر از محدوده اندیس یک آرایه خارج شویم برنامه خطا نمی گیرد بلکه حالت عادی ندارد. دراین حالت ممکن است دستگاه راه اندازی مجدد شود یا قفل کند **برای جلوگیری از چنین مشکلی باید از رهنمود {$R+} برای کنترل محدوده اندیس آرایه ها استفاده کرد.**

نوشتن رهنمود {$R+} باعث کندی اجرای برنامه می شود به همین دلیل **فقـط یک بار از آن استفاده کرده** ودر صورتی کـه برنامه خطا نداشت برای بار دوم آن را بردارید.

مثال 1) برنامه ای بنویسید که ابتدا یک آرایه 10 عنصری را با اعداد تصادفی بین صفر تا هزار تولید کند و آنـرا نمـایش دهـد ، سـپس ماکزیمم و موقعیت (اندیس سلول)آنرا نمایش دهد :

```
USES
   NEWDELAY,CRT;
TYPE
   VECTOR=ARRAY[1..10] OF INTEGER;
VAR
  B:VECTOR;
  I,MAX,POS:INTEGER;
BEGIN
  RANDOMIZE;
  CLRSCR;
  FOR I:=LOW(B)    TO     HIGH(B)   DO
    BEGIN
      B[I]:=RANDOM(1000);
      WRITE(B[I]:5);
    END;
  MAX:=B[1];
  POS:=1;
  FOR I:=2 TO  10 DO
    IF      B[I]  > MAX    THEN
      BEGIN
        MAX:=B[I];
        POS:=I;
      END;
  WRITELN;
  WRITELN(MAX:4 , POS:4);
END.
```

تولید یک آرایه 10 سلولی(عنصری) با اعداد تصادفی بین صفر تا هزار

پیدا کردن ماکزیمم و موقعیت آن

مثال 2) برنامه ای بنویسید که با استفاده از آرایه ابتدا 10 عدد از کاربر دریافت کند و ذخیره کند ، سپس فراوانی اعداد را با استفاده از آرایه ای دیگر چاپ کند :

```
USES
  NEWDELAY,CRT;
TYPE
    VECTOR=ARRAY[1..10] OF INTEGER;
VAR
    F,A:VECTOR;
    J,I:INTEGER;
BEGIN
  CLRSCR;
  FOR I:=1 TO 10 DO
    READLN(A[I]);
   CLRSCR;
  FOR I:=1  TO 10 DO
   BEGIN
     F[I]:=0;
     FOR J:=1  TO  10 DO
      IF A[I]=A[J] THEN
        F[I]:=F[I]+1;
   END;
  FOR I:=1 TO 10 DO
    WRITELN(A[I], '   ',F[I]);
END.
```

دریافت 10 عدد از کاربر و ذخیره بصورت یک آرایه

بدست آوردن فراوانی هر عدد و ذخیره آن در یک آرایه 10 عنصری دیگر

مثال 3) برنامه ای بنویسید که یک عدد دریافت کند و مبنای 16 آن عدد را درست چاپ کند :

```
TYPE
  VECTOR=ARRAY[1..10] OF CHAR;
VAR
 I,R,J,N:INTEGER;
 HEX:VECTOR;
BEGIN
  READLN(N);
  I:=1;
  WHILE N>0 DO
   BEGIN
     R:=N MOD 16;
     IF R<10 THEN
       HEX[I]:=CHR(48+R)
     ELSE
        HEX[I]:=CHR(55+R);
     I:=I+1;
     N:=N DIV 16;
   END;
   FOR  J:=I-1  DOWNTO 1 DO
    WRITE(HEX[J]);
END.
```

مثال 4) برنامه ای بنویسید که دو عدد 10 رقمی را رقم به رقم دریافت کند؛سپس این دو عدد را که در آرایه ای 10 تایی ذخیره شده اند، با هم جمع کند :

```
{$R+}
uses
   newdelay,crt;
var
  A,B:array[1..10] of 0..9;
  C:array[0..10] of 0..9;
  r,i,D:integer;
begin
   clrscr;
   writeln('Enter A');
   for i:=1  to 10 do
     readln(A[i]);
   writeln('Enter B');
   for i:=1  to  10  do
     Readln(B[i]);
   D:=0;
   for i:=10 downto  1 do
     begin
        r:=A[i]+B[i]+D;
        C[i]:=r mod 10;
        D:=r div 10;
     end;
    C[0]:=D;
    for I:=0 to  10 do
      Write(C[i]);
end.
```

مثال 5) یک آرایه مرتب حداکثر 10 عنصری را در نظر بگیرید؛برنامه ای بنویسید که این آرایه را دریافت کند و سپس با دریافت یک عنصر جدید Q از ورودی،اگر عنصر مورد نظر در آرایه موجود بود آنرا حذف کند در غیر اینصورت عنصر جدید را در محل مخصوص به خود در آرایه در ج کند :

توضیح :

برنامه بالا را با این تحلیل می نویسیم که :

```
uses
  newdelay,crt;
var
  A:array[1..10] of integer;
  tedad:byte;  i,z,m:integer;        c:char;
function search(m:integer):byte;
 var
    i:integer;
 begin
   search:=0;
   for i:=1  to  tedad do
     if A[i]=m then
       search:=i;
 end;
procedure hazf(z:integer);
 var
  i:integer;
 begin
   for i:=z to tedad do
     A[i]:=A[i+1];
   tedad:=tedad-1;
 end;
procedure add(m:integer);
  var
    i,j:integer;
   begin
      i:=1;
     if tedad=10 then exit;
     for i:=1  to tedad+1 do
       if A[i]>m then
          break;
    for j:=tedad+1  downto i do
      A[j]:=A[j-1];
    A[i]:=m;
    tedad:=tedad+1;
   end;
Begin
  tedad:=0;
  clrscr;
 repeat
    writeln('Enter a number');
    readln(m);
    z:=search(m);
    if z<>0 then
      hazf(z)
     else
      add(m);
    for i:=1 to tedad do
      write(A[i]:4);
    readln(c);
 until c='q';
end.
```

آرایه های دوبعدی:

برای تعریف ماتریس (آرایه دوبعدی) مانند تعریف آرایه های یک بعدی عمل می کنیم با این تفاوت که دوبعد به عنوان سطر(بعد اول) وستون(بعد دوم) تعریف می کنیم.نوع اندیس ها باید اسکالر باشد.

```
Type
  Mat = Array [1..5,boolean] of real;
Var
  a : mat ;
Begin
  a[3,true]:=100;     روش اول
  a[3][true]:=100;    روش دوم
  a[3,5>4]:=100;
  a[6,false]:=200;
end.
```

برای دسترسی به عناصر آرایه به دو روش می توانیم عمل کنیم:

در این سطر چون از محدوده اندیس ها خارج شد ایم برنامه حالت عادی ندارد.

مثال 6) برنامه ای بنویسید که دو ماتریس 3×4 را دریافت کند و حاصل جمع دو ماتریس را چاپ کند :

```
uses
   newdelay,crt;
var
  A,B,C:array[1..3 , 1..4]of integer;
  i,j:integer;
Begin
  Writeln('Enter A');
  for i:=1 to 3 do
    for j:=1  to 4 do
       Readln(A[i][j]);
  Writeln('Enter B');
  for i:=1 to 3 do
    for j:=1  to 4 do
       Readln(B[i][j]);
  for i:=1  to   3  do
    for j:=1   to 4  do
      C[i,j]:=A[i,j]+B[i,j];
  for i:=1 to 3 do
   begin
     writeln;
     for j:=1  to 4 do
        write(C[i,j]:5);
    end;
End.
```

مثال 7) برنامه ای بنویسید که یک ماتریس 4×4 را دریافت کند و ترانهاده آنرا چاپ کند :

```
uses
   newdelay,crt;
var
  A,B:array[1..4 , 1..4]of integer;
  i,j:integer;
```

ادامه برنامه در صفحه بعد

```
Begin
  CLRSCR;
  Writeln('Enter A');
  for i:=1 to 4 do
    for j:=1  to 4 do
       Readln(A[i][j]);
  for i:=1  to   4  do
    for j:=1   to 4  do
      B[J,I]:=A[i,j];
  for i:=1 to 4 do
   begin
     writeln;
     for j:=1  to 4 do
       write(B[i,j]:5);
   end;
End.
```

مثال 8) برنامه ای بنویسید که دو ماتریس 4×4 را دریافت کند؛سپس حاصلضرب این دو ماتریس را چاپ کند :

```
uses
   newdelay,crt;
var
  A,B,C:array[1..4 , 1..4]of integer;
  i,j,s,k:integer;
BEGIN
   clrscr;
   Writeln('Enter A');
   For i:=1  to   4 do
     for j:=1  to 4 do
       read(A[i][j]);
   Writeln('Enter B');
   For i:=1   to   4 do
     for  j:=1 to  4  do
       read(B[i,j]);
   for i:=1 to  4 do
    for j:=1  to 4 do
      begin
        s:=0;
        for k:=1  to  4 do
           S := S + A[i,k] * B[k,j] ;
        C[i,j]:=S ;
      end;
  for i:=1  to  4 do
    begin
      writeln;
      for j:=1 to 4 do
        write(C[i,j]:4);
    end;
END.
```

مهمترین قسمت برنامه : محاسبه ضرب دو ماتریس

آرایه های چند بعـدی :

می توان آرایه هایی با ابعاد بیشتر از دو نیز تعریف کرد.بطور کلی برای معرفی یک آرایه چنـد بعـدی مـی تـوان بـه دو روش زیر عمل کرد :

(روش اول) NAME : ARRAY [1..L] OF ARRAY [1..L2]… OF ARRAY [1..LN] OF TYPE ;

(روش دوم) NAME : ARRAY [1.. L1 , 1..L2 , … , 1.. LN] OF TYPE ;

مثال) آرایه سه بعدی به روش دوم بصورت زیر تعریف می کنیم :

```
VAR
A : ARRAY [ 1..2 , 1..4 , 1..3 ] OF REAL ;
```

برای دسترسی به اعضای این آرایه در حالت کلی بصورت A [I , J , K] عمل می کنند که در آن I,J,K ابعاد آرایه می باشند و در محدوده تعریف شده برای آرایه صدق می کنند.

نکته) آرایـــه های فشرده (Packed Array)

در پاسکال استاندارد هر کاراکتر داخل یک عنصر از آرایه قرار می گیرد . حال آنکه هر عنصر آرایه ظرفیت پـذیرش حداقل 4 کاراکتر و حداکثر 10 کاراکتر را دارا می باشد. که این ممکن است مشکل کمبود فضای حافظه را ایجاد کند لـذا برای جلوگیری از اشغال حجم بزرگی از حافظه ، آرایه های کاراکتری فشرده را تعریف می کنیم. در این حالـت ظرفیـت پذیرش هرعنصر از یک کاراکتر بیشتر نیست و بنابراین بدین صورت می توانیم از حافظه استفاده موثرتری داشته باشیم.
آرایه های فشرده بصورت زیر تعریف می شوند :

Name : packed Array [اندیس] of type ;

حسن آرایه های فشرده استفاده از حافظه کمتر است.

توجه : در توربو پاسکال بطور اتوماتیک آرایـه ها بصورت فشرده در نظر گرفته می شوند و نیاز به تعریف مجدد آنها بصورت فشرده نیست.

آرایه معمولی

			A
			L
			I

A
L
I

آرایه فشرده

ارسال آرایه به زیر برنامـه:

```
Type
  Ar=array [1..10]  of integer;
Var
  h:Ar;
function test (m:ar) :integer;
  var
        I,f:integer;
  Begin
        for i:=1 to 10 do
          f:=f+m[i];
        m[1]:=50;
        test:=f div 10;
  end;

var i:integer;
begin
  for i:=1 to 10 do
        readln( H[I] ):
  write(test(h));
  write(h[1]);
end.
```

الف) برنامه را در حالتی که M ارزش (مقدار) باشد بررسی می کنیم :

در این مثال آرایه H هنگام فراخوانی فانکشن test تحویل M داده می شود **و چون M یک پارامتر ارزش(مقدار) است از آرایه H یک کپی گرفته می شود** وکپی آن به اسم M شناخته می شود. دراین حالت **حافظه دو برابر اشغال** می شود همچنین اجرای زیـر برنامـه کند می شود زیرا باید تمام محتوای سلول های H در سلول های M کپی شود. **در عوض تغییر دادن آرایه M باعث تغییر آرایـه H نمی شود**.آرایه M **یک آرایه محلی است** که فقط در فانکشن test وجود دارد **وبعد از اتمام فانکشن test از بین می رود**.

ب) حال برنامه قبل را در حالتی که M مرجع (متغیر) باشد بررسی می کنیم :

در این حالت آدرس آرایه H به M تحویل داده می شود در واقع آرایه H و آرایه M یکی هستند به همین دلیل **حافظه اضافه اشـغال نشده است و سرعت نیز پایین نیامده است** اما **اگر آرایه M را تغییر دهید آرایه H نیز تغییر می کند**.**(امنیت ندارد)**

ج) مساله قبل را در حالتی بررسی می کنیم که پارامتر M ثابت(const) باشد :

در این حالت مانند حالت مرجع عمل می کنیم یعنی آدرس آرایه H تحویل M داده می شود **یعنی فضا اشغال نمی شود** زیـر برنامـه کند نمی شود اما **برنامه نویس نمی تواند آرایه M را تغییر دهد که در این مثال خط M[1]:=50 خطا خواهد بود**. **که در این صورت امنیت نیز به وجود می آید**.

نکته : هنگام فراخوانی یک زیر برنامه باید نوع پارامتر ها(دریافتی وارسالی) یکی باشد یعنی اگر یک آرایه را به زیر برنامـه ارسـال مـی کنیم باید با پارامتر مجازی آن هم نوع باشد در غیر این صورت خطاست.

نکته : اگر آرایه را به روش دوم (Type) تعریف کنیم هنگام ارسال آرایه به زیر برنامه ، برنامه نویس ساده تر می تواند پارامتر مجـازی را تعریف کند.

نکته : تابع **نمی تواند** با نام خود یک آرایه را به برنامه اصلی برگرداند به عبارت دیگر **خروجی تابع نمی تواند آرایه باشد**. برای اینکه **زیربرنامه خروجی اش آرایه باشد باید یک پارامتر مرجع به زیر برنامه ارسال کنیم** تا در زیر برنامه تغییر کند وهنگامی که به برنامه اصلی برگشتیم تغییرات اصلی را مشاهده کنیم.

پارامتـر باز :

پارامتر باز پارامتری است که هنگام فراخوانی تابع یا پروسیجر، **یک آرایه با هر تعداد سلول** دریافت می کند. نحوه تعریف پـارامتر باز به صورت زیر است:

```
Function   add(z : array of integer) : longint;
```

در پارامتر باز محدوده اندیس اعلام نمی شود به همین دلیل می توانیم هر آرایه ای با تعداد عناصرمختلف را به آن تحویل دهیم. **برای مشخص کردن کمترین وبیشترین اندیس از توابع استاندارد low و high استفاده می کنیم.**

مثال)

```
Var
  A:array [1..10] of integer;
  B:array[20..40] of integer;
  i:integer;
 function add(z : array of integer) : longint;
  var
        s:longint;
        i,l,h : integer;
  begin
        L:=low(z);
        H:=high(z);
        s:=0;
        for i:=L to H do
          s:=s+z[i];
        add:=s;
  end;
begin
  for i:=1 to 10 do
        readln( a[i] ) ;
  for i:=20 to 40 do
        readln( b[i] );
  writeln(add(a));
  writeln(add(b);
end.
```

در مثال ذکر شده دو آرایه A و B از نوع integer تعریف شده اند اما با محدوده اندیس های متفاوت. فانکشن add هر دو آرایه را می تواند تحویل بگیرد. این فانکشن مجموع عناصر آرایه را برمی گرداند.

رشتـــه ها :

رشته شبیه آرایه ای از نوع **char** است ؛ یعنی می توانیم خط عنوان string را به صورت زیر تصور کنیم:

```
Type
  String = array  [0..255]  of  char ;
```

مثال)

```
S:='ALIREZA';
Writeln(s[4]); ž R
  S[3]:='M';
  Writeln(s); ž ALMREZA
  S[8]:='B';
  Writeln(s); ž ALMREZA
  S[0]:=#8;
  Writeln(s) ž ALMREZAB
  S[0]:=#0;
  writeln(s); ž چیزی چاپ نمی شود
  s[0]:=#3;
  writeln(s); ž ALM
  s[0]:=#10 ;
  writeln(s); ž ALMREZAB    معلوم نیست که دو کاراکتر آخر چه چیزی باشد
```

می توانیم به کاراکتر های داخل رشته دسترسی مستقیم داشته باشیم و کاراکترها را تغییردهیم یا چاپ کنیم.**اگر دستور انتـساب بر روی کل رشته انجام شود(خط اول در مثال بالا) سلول شماره صفر، تغییر می کند** اما **اگر بر روی یکی از خانه های رشته باشد سلول شماره صفر تغییر نمی کند (خط پنجم در مثال بالا)** برنامه نویس می تواند **خانه شماره صفر را تغییر دهد که در این صورت طول رشته تغییر کرده است.**

ü نکته: دستور روبرو خطاست. • **S[0]:=4**

مقایسه رشتـــه ها :

برای مقایسه رشته ها **حرف اول رشته اول** با **حرف اول رشته دوم** مقایسه می شود اگر نتیجه گرفته نشد حرف دوم ها را مقایسه می کنیم و الی آخر. **معیار مقایسه کد اسکی کاراکترهاست. از هر حرفی که نتیجه گرفته شد مابقی حروف مقایسه نمی شود.**

مثال 1) برنامه ای بنویسید که یک رشته را دریافت کند و اعداد داخل آنرا استخراج کند :

بعنوان مثال رشته A5B82C را دریافت کند و عدد 582 را در خروجی چاپ کند .

```
var
  s:string[50];
  N:longint;
  i:integer;
Begin
   readln(S);
   N:=0;
   for i:=1  to  ord(S[0]) do
     if (S[i]>='0') and (S[i]<='9') then
      N := N * 10 + Ord(s[i]) – 48 ;
   writeln(N);
End.
```

مثال 2) برنامه ای بنویسید که یک رشته را دریافت کند و معکوس آنرا در خروجی چاپ کند :

```
var
  s,m:string;
  i:integer;
Begin
   readln(S);
   m:='';
   for i:=1  to  ord(S[0]) do
     m:= s[i] + m;
   m[0]:=s[0];
   writeln(m);
End.
```

بعنوان مثال رشته pascal را دریافت کند و رشته lacsap را چاپ کند.

مثال 3) برنامه ای بنویسید که یک رشته از کاربر دریافت کند و فراوانی هر یک از حروف الفبای انگلیسی را چاپ کند و همچنین فراوانی کاراکترهای دیگر را بوسیله متغیری بنام other چاپ نماید :

```
var
  S:string;
  f:array['A'..'Z']of byte;
  other,i:integer;
  C:char;
begin
  readln(S);
  for  C:='A' to 'Z' do
    f[C]:=0;
   other:=0;
  for i:=1  to ord(s[0]) do
    if (S[i]>='A') and (S[i]<='Z') then
      F[S[i]]:=F[S[i]]+1
     else
       other:=other+1;
   for c:='A' to 'Z' do
     if F[C]>0 then
       Writeln(C ,'  ', F[C]);
    writeln('Other=',other);
end.
```

بعنوان مثال رشته WWW)987HHKKPHH را دریافت مـی کنـد و خروجی آن بصورت زیر می باشد :

```
W  3
H  4
K  2
P  1
OTHER    4
```

الحاق رشتـــه ها :

برای **الحاق رشته ها از دو روش** زیر استفاده می کنیم :

1- **استفاده از عملگر** +

2- **استفاده از تابع concat**

الحاق در پاسکال خاصیت جابجایی ندارد.

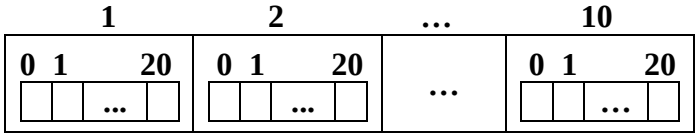
ارسال رشتـــه به زیر برنامه:

رشته را می توان به سه روش **مقدار(ارزشی) ومتغیر(مرجع) یا ثابت** به زیر برنامه ارسال کرد.**خروجی تابع می تواند رشته ای باشد در صورتی که آرایه نمی تواند باشد**.

آرایه ای از رشتــه ها :

آرایه ای از رشته ها ، آرایه ای است که **هر عنصر آن می تواند یک رشته** را در خود نگه دارد.

```
Var
   S : array[1..10] of  string [20] ;
   I : Integer ;
Begin
   For  I := 1    to    10    do
            Readln(S [I] ) ;
   s[2]:='abcdef';
   s[3]:='xyz';
   writeln( s[3][2] , s[2,4] );
   For  I := 1    to    10    do
            writeln( S [I] ) ;
  Readln;
End.
```



سوال) دستور **Writeln(s[3][2] , s[2,4]);** چه چیـزی در خروجـی چاپ می کند ؟

آرایه ای از رشته ها شبیه یک ماتریس دوبعدی از نوع **char** است.

ثابت نــوع آرایه ای:

```
Type
  Ar =  array [1..5]    of    string ;
Const
  Name : Ar =  (  'ali'   ,   'reza'    ,   ''   ,   'amin'    ,   'akbar'    ) ;
```

آرایه Name از نوع رشته است و یک ثابت نوع دار است که در لحظه تعریف مقدار دهی اولیه شده است و می توان مقادیر داخـل آن را در هنگام اجرای برنامه تغییر داد.

زیر برنامه های استاندارد رشتــه ای :

تابع length : طول یک رشته را برمی گرداند.

```
Ord( s[0] )=length(s);
```

تابع concat : این تابع **تعداد نامشخصی از رشته ها را دریافت کرده وآنها را الحاق کرده** وبر می گرداند این **تابع حاصیت جابجایی ندارد.**

Concat (رشته۱ , رشته۲ , … , رشتهn) • رشته

```
S:=concat('ali' , 'reza' , 'akbar' );   ž   alirezaakbar
```

پروسیجر str : این پروسیجر**یک عددریافت کرده وآن را تبدیل به رشته می کند و رشته را درپارامتردوم می ریزد.**

str (متغیر رشته ای , عبارت یا متغیر یا عدد)

```
مثال)    str (14,s1);
  str (2.5,s2);
  writeln(s1,s2);   ž   142.5
مثال)    str($12+3,s);
  Writeln(s);   ž   21
```

مثال ۴) برنامه ای بنویسید که پروسیجر str را شبیه سازی کند :

```
procedure mystr(n:integer ;var s:string);
var
  r,i:integer;      m:string;
begin
  i:=1;
  while n>0 do
   begin
     r:=n mod 10;
     n:=n div 10;
     s[i]:=chr(r+48);
     i:=i+1;
   end;
  s[0]:=chr(i-1);
  m:='';
  for i:=1  to length(s) do
    m:=s[i]+m;
  m[0]:=s[0];
  s:=m;
end;
 var
   z:string;
begin
    mystr(348,z);
    writeln(z);
end.
```

تمرین) خروجی برنامه چیست ؟

```
var
  s:string;
begin
   str(-3.758:0:2,s);
   writeln(s);
   writeln(length(s));
end.
```

تابع upcase : این تابع یک کاراکتر را دریافت کرده وحرف بزرگ آن را بر می گرداند.

Upcase(کاراکتر) • کاراکتر

Upcase ('b'); ž *B*
Upcase ('A'); ž *A*
Upcase ('+'); ž +

مثال ۵) برنامه ای بنویسید که یک رشته از کاربر دریافت کند و همه ی حروف آنرا به کوچک تبدیل کند :

```
var
  s:string;      i:integer;
begin
  readln(s);
  for  i:=1   to   length(s)   do
    if  (s[i]>='A')  and  (s[i]<='Z')  then
       s[i] := chr ( ord (s[i]) + 32 );
   writeln(s);
end.
```

پروسیجـر val : این پروسیجر یک رشته را دریافت می کند و در صورتی که بتواند آن را تبدیل به عدد کند **عدد را در پارامتر دوم** و **عدد صفر را در پارامتر سوم** می ریزد **ولی** اگر نتواند تبدیل کند **آنگاه محل خطا(محل کاراکتر غیر مجاز) را در پارامتر سوم** می ریزد و در **پارامتر دوم صفر** می ریزد.

Val(رشته , متغیر عددی , متغیر صحیح)

مثال) val ('128' , a , c); در این مثال c=0 وa=128

مثال) val ('12a45' , a, c); دراین مثال c=3 و a=0

مثال) val ('12.56' , rlA , intC); در این مثال intC=0 و rlA=12.56

مثال) val ('14.76' , intA , intC); در این مثال intC=3 و intA=0

تابع pos : این تابع **رشته ۱ را در رشته ۲ جست وجو می کند درصورت پیداشدن محل اولین وقوع** آن رابرمی گرداند.

Pos(رشته ۲ , رشته ۱) • عدد صحیح

- Pos ('amh' , 'amzhbamhdqamh') • 6
- Pos ('abc' , 'aBCdABcd') • 0
- pos ('ABCDEF' , 'CD') • 0

تابع copy : این تابع **قسمتی از یک رشته را استخراج می کند.**

Copy(رشته , عدد صحیح شروع , عدد صحیح طول) • رشته

- Copy ('ABCDEFG', 2,3) • *BCD*
- copy ('ABCD' , 4,3) • *D*

مثال ۶) برنامه ای بنویسید که تابع کپی را شبیه سازی کند :

```
function mycopy(s:string;  k,n:integer):string;
var
  m:string;     i:integer;
 begin
  m:='';
  If   n+k > length(s)     then
    n := length(s) – k + 1;
  for     i:=k     to     k+n     do
     m[i-k+1] := s[i];
  m[0] := chr(n);
  mycopy := m;
 end;
begin
 writeln(mycopy( 'abcdef' , 2 , 4) );
end.
```

پروسیجر insert : این پروسیجر **یک رشته را از محل مشخصی در رشته دیگر درج** می کند.

Insert (۱ رشته , متغیر رشته ای , عدد صحیح)

مثال) s:='ABCDEFG';
Insert ('XYZ',s,5);
Writeln(s); ž *'ABCDXYZEFG'*

نکته : دستور insert(s1,s2,length(s2)+1) برابر است با s2:=s2+s1

نکته : دستور insert را می توان با ترکیب دستورهای copy و الحاق انجام داد:

Insert(s2,s1,x) برابر است با خط زیر

S1:= copy (s1,1,x-1) + s2 + copy (s1, x , length(s1) – x + 1)

پروسیجر delete : این پروسیجر **قسمتی از یک رشته را حذف** می کند.

Delete(متغیر رشته , عدد صحیح شروع , عدد صحیح طول)

S := 'ABCDEFG' ;
Delete(s,3,2);
Write(s); ž *'ABEFG'*

به وسیله تابع copy وعمل الحاق پروسیجر delete را شبیه سازی کنید.

مثال ۷) برنامه ای بنویسید که رشته های Z3 , Z2 , Z1 را دریافت کند؛سپس رشته Z2 را از Z1 حذف کند و بجای آن Z3 را قرار دهد :

```
procedure replace(var s1:string ; s2,s3:string);
var
  p,L1:integer;
begin
   L1:=length(s2);
   p:=pos(s2,s1);
   while(p<>0) do
   begin
     delete(s1,p,L1);
     insert(s3,s1,p);
     p:=pos(s2,s1);
   end;
end;
var
  z1,z2,z3:string;
begin
  readln(z1);
  readln(z2);
  readln(z3);
  replace(z1,z2,z3);
  writeln(z1);
end.
```

مثلا رشته Z1 = abcdefcdghcd و Z2 = cd و Z3 = pascal را وارد می کنیم ، cd را که همان Z2 است از Z1 که همان abcdefcdghcd است حذف می کند و بجای cd های حذف شده ، pascal را قرار میدهد ؛ یعنی در نهایت رشته abpascalefpascalghpascal را که همان Z1 است در خروجی چاپ می نماید .

مثال ۸) برنامه ایی بنویسید که رشته ایی از کاراکترها که بیانگر یک عدد رومی است را از صفحه کلید بخواند و سپس عدد معـادل آن را چاپ کند. حروف رومی به صورت جدول زیر هستند.

عدد معادل	حرف رومی
1000	M
500	D
100	C
50	L
10	X
5	V
1	I

مثلا رشته MCX معادل عدد 1110 است یا رشتهٔ DLXVI معادل عدد 566 است. یا رشتهٔ CXXV معادل عدد125 است.

```
function rumi (s:string):longint;
var
  i:integer;
  n:longint;
begin
   n:=0;
   for i:=1 to length(s) do
      case s[i] of
        'M': n:=n+1000;
        'D': n:=n+500;
        'C': n:=n+100;
        'L': n:=n+50;
        'X': n:=n+10;
        'V': n:=n+5;
        'I'  : n:=n+1;
      end;
   rumi:=n;
end;
var
  z:string;
begin
   readln(z);
   writeln(rumi(z));
end.
```

رکـــورد :

رکورد مجموعه ای از **فیلد ها** است که هر کدام می توانند **یک نوع خاصی** داشته باشند و **هر کدام نیز دارای یک نـام هـستند** و **تحت یک نام ذخیره می شوند**. مثلا برای ذخیره کردن اطلاعات یک دانش آموز شامل نام فامیلی ، شماره دانش آموزی و معدل رکورد زیر را تعریف می کنیم .

	Name	Family	StdId	AVG
Student				

تفـاوت آرایـه و رکـــورد :

۱- **نوع سلول های آرایه یکسان است در صورتی که فیلد های رکورد می توانند هم نوع نباشند.**

۲- **آرایه دارای اندیس است اما رکورد هر فیلدش دارای یک نام است .**

نحوه ی تعـــریف رکورد :

برای تعریف رکورد ابتدا در قسمت Type نوع آن را تعریف می کنیم سپس **در قسمت var متغیر رکوردی را تعریف می کنیم برای تعریف نوع رکورد از کلمه رزرو شده record به روش زیر استفاده می کنیم** :

```
Type
 Rec1=record
   Name : string [10];
   Family: string [15];
   AVG: real;
   STDid : longint;
 End;
Var
    Student : rec1;
```

مجموع فضای اشغالی فیلد ها ، حافظه اشغال شده ی رکورد را تعیین می کند .

برای مثال فضای اشغالی رکورد بالا از محاسبه روبرو به دست می آید : 11+16+6+4=37

هنگامیکه می خواهیم اطلاعات مختلفی از شی یا فرد یا ... را ذخیره کنیم بهتر است که اطلا عات آن را کنار هم داشته باشـیم بـه همین دلیل از رکورد استفاده می کنیم .

نحوه ی دسترسی به فیلدهای رکـــورد :

برای دسترسی به فیلدهای یک رکورد **ابتدا نام رکورد را نوشته ،سپس یک نقطه می گذاریم و بعد از آن نام فیلد** را می گذاریم.

نام فیلد . نام رکـــورد

```
Begin
  Student . name := ‘ali’;
  Readln (student . family);
  Student . AVG := student . AVG+10;
  Writeln (student . STDid);
  Student . name[0] := #3;
End.
```

برای دسترسی به فیلد های یک رکورد می توانیم از دستورWith استفاده کنیم،تکرار نام رکورد وقت گیر و خسته کننده می باشد.

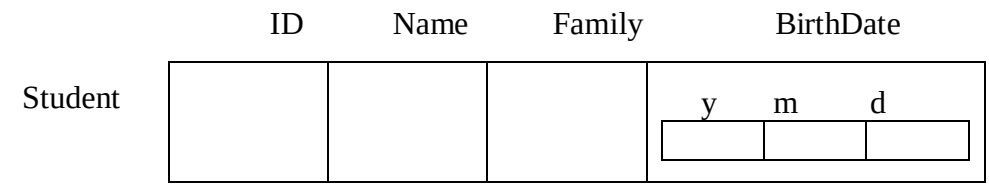
مثال ۱) برای پی بردن به چگونگی استفاده از دستور With به برنامه ی زیر دقت کنید :

```
TYPE
 STD=RECORD
   NAME:STRING;
   FAMILY:STRING;
   AVG:REAL;
  END;
VAR
  S1 , S2 : STD ;
BEGIN
   S1.NAME := 'ALI' ;
   WITH   S1    DO
    BEGIN
      FAMILY := 'HASANI' ;
      AVG := 14 ;
    END;
  S2 := S1;
END.
```

- **رشته ها و آرایه ها را نمی توان یکجا دریافت یا چاپ کرد** .
- اگر دو رکورد **هم نوع** داشته باشیم با یک دستور انتساب می توانیم محتوای فیلد های یکی را در دیگری کپی کنیم.

رکـورد های تو در تو (رکورد حاوی رکورد) :

فیلدهای یک رکورد می توانند از نوع رکورد دیگری باشند به عنوان مثال می خواهیم اطلاعات یک دانش آموز شامل شماره دانش آموزی، نام، فامیلی و تاریخ تولد را داشته باشیم که تاریخ تولد خود از سه قسمت سال،ماه،روز تشکیل می شود .




```
Type
     Rec1= record
       Y: integer;
       M: 1 .. 12;
       D: 1 .. 31;
      End;
    Rec2= record
        Id: longint;
        Name: string[10];
        Family: string [15];
        BirthDate : rec1;
    End;
Var
   Student : rec2;
```

متغیر student چند بایت فضا اشغال می کند؟

آرایه ای از رکـورد ها :

برای ذخیره اطلا عات ۱۰۰ دانش آموزکه هر دانش آموز دارای نام،فامیلی و معدل است از ساختار زیر استفاده می کنیم:

	1	2	…	100
Students	Name Family AVG	Name Family AVG	…	Name Family AVG

نحـوه تعریف :

```
Type
   Rec = record
        Name : string [10];
        Family: string [15];
        AVG : real;
   End;
Var
   Students : array [1..100] of Rec ;
```

نحـوه دستـرسی :

آرایه نام [اندیس] فیلد.[مقـدار := ;

مثال) Students [2].AVG := 11 ;

برای دسترسی به تمام سلولهای آرایه از حلقه **FOR** استفاده می کنیم.

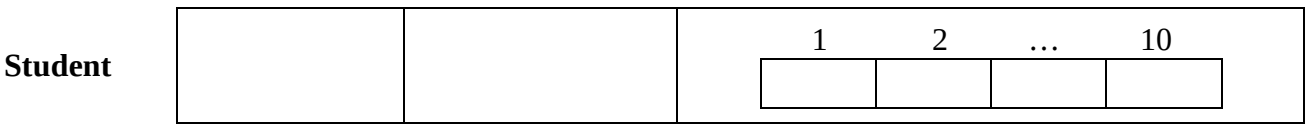
مثال ۲) اطلاعات روبرو مربوط به مشترکین تلفن می باشد : شماره تلفن ، نام ، آدرس و هزینه .برنامه ای بنویسید که با آن بتوانیم اطلاعات مربوط به ۵۰ مشترک را ذخیره کنیم . همچنین بوسیله ی زیربرنامه ای مشترکین پر مصرف را چاپ نمایید.(مشترکینی را پر مصرف فرض می کنیم که هزینه شان از میانگین هزینه تمام ۵۰ مشترک بیشتر باشد).

```
TYPE
 REC=record
   number:integer;
   name:string[10];
   address:string[20];
   cost:longint;
 end;
var
  M:array[1..50] of REC;
  i:integer;
procedure Hazine;
 var
   s:longint;
begin
  s:=0;
  for i:=1 to 50 do                 } محاسبه میانگین
    s := s + M[i].cost ;            } هزینه مشترکین
  s:=s div 50;
  for i:=1 to 50 do                 } چاپ مشترکین پر مصرف
    if M[i].cost > s then
      writeln(M[i].number,M[i].name,M[i].cost);
end;
```

```
begin
  for i:=1 to 50 do
    with M[i] do
      begin
        readln(number);
        readln(name);
        readln(address);
        readln(cost);
      end;
   Hazine;
end.
```

رکـوردی از آرایـه :

برای ذخیره کردن اطلاعات یک دانش آموز شامل نام و فامیلی و نمرات ۱۰ درس آن از ساختار زیر استفاده می کنیم :



نحـوه ی تعـریف :

```
Type
    Rec = record
        Name : string[10];
        Family: string[15];
        Grades : Array [1..10] of  real ;
    End;
Var
   Student : Rec;
```

نحـوه دستـرسی :

```
مقـدار  :=  [ اندیس ] نام آرایه . نام رکورد ;
Student . Grades[2] := 5.5 ;
```

ثابت نـوع دار رکـوردی :

می توانیم یک ثابت نوع دار از نوع رکورد تعریف کنیم و مقدار دهی اولیه کنیم.این رکورد در طول اجرای برنامه قابل تغییر است .
نحوه ی تعریف به صورت زیر است :

```
Type
    Rec = record
        Name : string[10];
        ID : longint ;
        AVG : real ;
    End;
const
   Student : Rec = (  Name :  'ali'  ;  ID : 57438   ;    AVG : 13.25  ) ;
```

ارسال رکـورد به زیر برنامه :

یک رکورد را می توان به صورت یک پارامتر به **زیر برنامه** ارسال کرد. برای این کار باید پارامتر تحویـل داده شـده(آرگومـان)بـا پارامترمجازی هم نوع باشد.**پارامتر مجازی می تواند از نوع مرجع یا مقدار یا ثابت باشد.**

- خروجی تابع(مقدار برگشتی آن) نمی تواند از نوع آرایه یا رکورد باشد.در صورت نیاز به برگشت یک رکورد از یک تابع باید آن را به صورت مرجع(متغیر) تعریف کنیم .

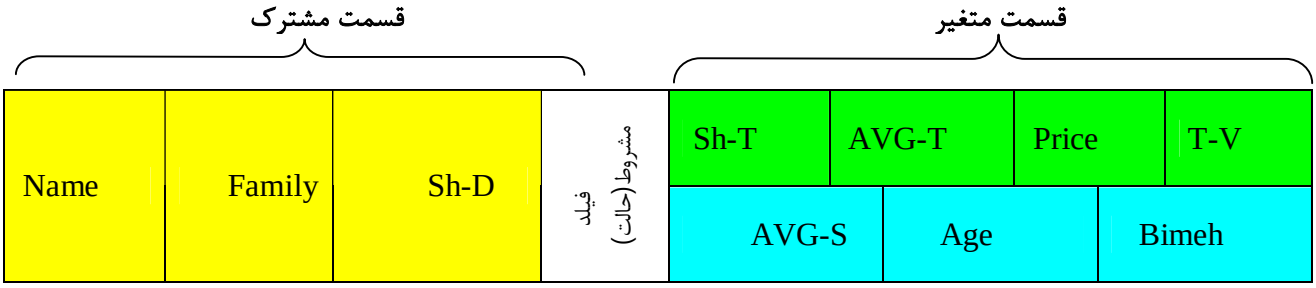
رکـورد متغیـر :

می خواهیم اطلاعات دانش آموزان یک مدرسه را ذخیره کنیم. **در مدرسه دو نوع دانش آموز وجود دارد: دانش آموز ترمی و دانش آموز سالی واحدی**. برای ذخیره آنها می خواهیم یک رکورد با مشخصات زیرتعریف کنیم.

ترمی (نام- فامیلی- شماره دانش آموزی- شماره ترم- معدل ترم- مبلغ- تعداد واحد)
سالی واحدی (نام- فامیلی- شماره دانش آموزی- معدل سال- سن- بیمه)

این دو نوع دانش آموز دارای **سه فیلد مشترک می باشند(نام- فامیلی- شماره دانش آموزی)** . اما بقیه اطلاعات آنها با هم **متفـاوت** است.

برای ذخیره کردن آنها یک رکورد به صـورت زیر تعریف می کنیم :



قسمت متغیر رکورد ، یک فضای مشترک است بین دو حالت ترمی و سالی واحدی- یعنی می توان این قسمت را که یک فضا است **در یک حالت با چهار فیلد در نظر گرفت(ترمی)** و **در حالت دیگر همین فضا را سه فیلد در نظر گرفت(سالی واحدی)**.قسمت متغیر آن به صورت زیر است :

Sh-T *Avg-s*		AVG-T *age*		Price *bimeh*	T-V

رکورد متغیر دارای یک فیلد به نام فیلد مشروط(حالت) است که نشان دهنده ی تفسیر برنامه نویس از قسمت متغیر است .
فیلد مشروط (حالت) باید از نوع اسکالر باشد .

چگونگی تعریف رکـورد متغیـر :

```
Type
  Rec = record
     Name : string [10];
     Family : string [15];
     Shstd : longint;
      Case  ts : byte  of
           1: ( sh-t:byte;   avg-s:real;   price:longint;   T-V:byte;);
           2: ( avg-t:real;  age:byte;    bimeh:boolean;);
    End;
Var
     Student : rec;
```

مقدار فضـای اشغالی رکـورد های متغیر :

برای محاسبه مقدار فضای اشغالی یک رکورد متغیر به روش زیر عمل می کنیم :

ماکزیمم حالت متغیر + (فیلد تعیین گونه) اندازه ی فیلد مشروط + اندازه ی قسمت ثابت = فضای اشغالی

نحوه دسترسی به فیلد های رکورد متغیر :

نحوه دسترسی به فیلد های رکورد متغیر تفاوتی با رکورد های قبلی ندارد ؛ فقط با یک دستور if یا case قسمت متغیر را حالت بندی می کنیم.

مثال ۳) با استفاده از رکورد متغیر ، دو نوع کارگر رسمی و قراردادی را تعریف می کنیم و برنامه ی دریافت اطلاعات این دو نوع کارگر را می نویسیم :

در این مثال ۳ فیلد مشترک (نام ، فامیلی و شماره شناسایی) داریم که برای هر دو نوع کارگر می باشد.

اما برای کارگر رسمی علاوه بر ۳ فیلد مشترک فیلدهای بیمه ، حقوق ثابت و پاداش در نظر مـی گیـریم و بـرای کـارگر قـراردادی هـم علاوه بر ۳ فیلد مشترک ، فیلدهای مدت قرارداد و حقوق را در نظر می گیریم.

```
type
  Worker = record
     name:string[10] ;
     family:string[20] ;
     id:integer ;

  case  t :boolean of
      true : ( modat : integer;    salary : longint );
      false : (bime : integer;     sabet : longint;   padash : integer   );
 end;
var
   W :worker;       c:char;
begin
  readln(w.name);
  readln(w.family);
  readln(w.id);

  readln(c);
  if   c = 'r'   then
    w.t := false
  else
    w.t := true;

  if   w.t = false    then
    readln( w.bime   ,  w.sabet   ,  w.padash )
   else
     readln( w.modat  ,  w.salary );
end.
```

فیلد های مشترک

فیلد مشروط (حالت)

دریافت یک کاراکتر از ورودی :
اگر کاراکتر r (کارگر رسمی) باشد فیلد حالـت را false مـی کنـیم.
در غیر اینصورت فیلد حالت را True می کنیم .

حال اگر w.t که همان فیلد مشروط (حالت) اسـت False بـود ، اطلاعات مربوط به کارگر رسمی(بیمه ، حقوق ثابت و پاداش) را از ورودی دریافت می کنیم در غیر اینصورت (اگـر ; w.t = true) اطلاعات مربوط به کـارگر قـراردادی (مـدت قـرارداد و حقـوق) را دریافت می کنیم.

روشهـای جستجو و مرتب سازی

تعریف لیست :

لیست از تعدادی داده تشکیل شده است که با یکدیگر در خاصیتی مشترک هستندو تعریف لیـست بـسیار شـبیه تعریـف مجموعـه در ریاضی است با این تفاوت که :

- در مجموعه(Set) عضو تکراری وجود ندارد اما در لیست ممکن است داده تکراری وجود داشته باشد.
- در مجموعه ترتیب اعضا اهمیت ندارد اما در لیست ترتیب داده ها اهمیت دارد.

سئوال) آیا نمرات درس پاسکال می تواند مجموعه باشد؟ **جواب** : اگر نمره تکراری در آنها نباشد می تواند مجموعه باشد

سئوال) آیا لیست روبرو می تواند مجموعه باشد؟ {4 , 2 , 3 , 8 , 7 , 2} **جواب** : خیر زیرا عضوتکراری دارد.

نکته) یک **مجموعه می تواند حتما لیست باشد** اما **یک لیست همیشه یک مجموعه نمی تواند باشد**.

نکته) داده های موجود در یک لیست می تواند از چند نوع داده متفاوت باشد مثلا : { ali , 14 , 15.5 , 18}

دقت شود که داده های یک لیست در خاصیتی باید مشترک باشند.

نکته) رکورد و آرایه می توانند یک لیست باشند.

5	8.5	'A'	ALI

5	25	19	14	6	13	9	8

جستجودر لیست :

در مواردی می خواهیم یک **داده خاصی** را در یک لیست پیدا کنیم ؛ به این داده **کلیـد** می گویند. روش های مختلفی برای جستجو در لیست ها وجود دارد که در اینجا به دو مورد آن اشاره می کنیم :

1 - روش جستجوی خطـی (ترتیبی)

در این روش کلید(داده ایی که قرار است جستجو شود) با اولین دادهٔ لیست مقایسه می شود **اگر پیدا شد** کار به **پایان** می رسد در غیر این صورت با دومین دادهٔ لیست مقایسه می شود و این کار تا پایان لیست ادامه می یابد.

مثال) اگر کلید 12 باشد در لیست زیر **چند عمل مقایسه** انجام می شود تا کلید پیدا شود؟ (به روش خطی)

10	13	14	11	7	12	19	17	12	2

نکته) اگر در لیست چند مورد از کلید باشد و به روش خطی جستجو کنیم **اولین کلید** را پیدا می کند و اندیس (فیلد) لیست را برمی گرداند و **کار جستجو تمام می شود**. در مثال بالا دو عدد 12 وجود دارد اما اولین 12 که در خانه ششم است پیدا می شود و کار تمام می شود.

نکته) در روش جستجوی خطی لازم نیست که لیست ما مرتب شده باشد.(یعنی داده ها در لیست صعودی یا نزولی باشند).

```
Var   A: Array[1..100] of integer;
      Key , F:integer;  B:Boolean;
Begin
  For   F:=1  to  100 do
     Readln( A[F]);
  Readln(Key);
  B:=False;
  For  F:=1   to   100  do
     IF  A[F] = Key  then
           Begin
                B:=True;
                 Break;
           End;
  If   B=False   then
       Writeln(' Not Found ')
  Else
       Wrietln(' Find ' , F);
End.
```

نکته) در روش جستجوی خطی عمل مقایسه یکبار انجام میشود.در **بهترین** حالتِ جستجوی خطی، به تعداد خانه های لیست، عمل مقایسه انجام میشود ؛ بنابراین **اگر لیست N تا عنصر داشته باشد N عملِ مقایسه** انجام می شود. در **حالت متوسط N / 2 عملِ مقایسه** انجام می شود.

سئوال) اگر لیستی دارای ۲۰ عنصر باشد در روش جستجوی خطی در بدترین حالت و حالت متوسط چند عمل مقایسه انجام می شود؟

نکته) در روش جستجوی خطی لیست می تواند نامرتب باشد.

ولی **تعداد مقایسه هازیاد است و جستجو به کندی صورت می گیرد.**

2 - روش جستجوی دودویـی (باینری) :

در این روش باید لیست از قبل مرتب باشد. پس این روش مخصوص لیست هایی است که از قبل مرتب(صعودی یا نزولی) شده اند.
فرض کنید که یک آرایه مرتب صعودی (از کوچک به بزرگ) داریم و می خواهیم کلیدی را در آن به روش باینری جستجو کنیم :
آرایه را از وسط نصف می کنیم و کلید را با عنصر وسط آرایه مقایسه می کنیم که سه حالت ممکن است رخ دهد:

1) **کلید از عنصر وسط بزرگتر باشد(Key > A[C])** : در این حالت با توجه به اینکه لیست مرتب (صعودی) است حتما کلید در نیمه سمت راست است و ما باید نیمه سمت راست را جستجو کنیم و نیمه سمت چپ را جستجو نخواهیم کرد.

	نیمه سمت راست	وسط	نیمه سمت چپ	

2) **کلید از عنصر وسط لیست کوچکتر است** : در این حالت با توجه به اینکه لیست به صورت صعودی مرتب است حتمـا کلیـد در نیمه سمت چپ است و باید نیمه سمت چپ را جستجو کرد و نیمه سمت را جستجو نخواهیم کرد.

3) **در بهترین حالت کلید مساوی عنصر وسط لیست باشد** : در این حالت کلید پیدا شده است و عمل جستجو متوقف می شود.
اگر هر کدام از **حالت های یک یا دو رخ دهد؛** عمل جستجو را در نیمه سمت راست یا چپ ادامه می دهیم. به عبارت دیگر نیمه سمت راست یا نیمه سمت چپ را دوباره از وسط نصف می کنیم و دوباره عمل بالا را روی آن انجام می دهیم.

نکته) در **بهترین حالت** جستجوی دودویی یک عمل مقایسه انجام می شود و در **بدترین حالت** جستجوی دودویی یک لیستی که دارای N عنصر باشد ؛ تعداد عملِ جستجو $[\text{Log}_2(N)] + 1$ است.

مثال)
لیستی دارای ۱۸ عنصر است؛ در **بدترین و بهترین حالت** جستجوی یک کلید به **روش دودویی** چند عمل مقایسه انجام می شود؟ **به روش خطی چطور** ؟

نکته)
جستجوی اطلاعات غیر عددی(رشته یا کاراکتر) دقیقا مانند اطلاعات عددی است.

نکته)
معمولا در حالتهایی که تعداد **عناصر لیست زیاد** باشد **جستجوی دودویی** بسیار بهترعمل می کند.

سئوال) در چه حالتی **جستجوی خطی از جستجوی دودویی** سریعتر است؟

سئوال) درجستجوی **باینری** اگر کلید در لیست نباشد، چگونه متوجه می شویم که کلید در لیست نیست؟

```
Var   A:array[1..20] of integer;
  J, Key , C , E , S : integer;
F:Boolean;
Begin
  For   J:= 1  to 20  do
        Readln( A[ J ] )   ;
   Sort(A); { یک زیربرنامه جهت مرتب سازی }
S:=1 ;    E : =20;
Readln(Key);
F:=False;
While  (S ≤ E)  and ( Not  F)  Do
  Begin
      C:= ( S+E) Div 2;
      IF  A[ C]  < Key  then
           S:=C+1
     Else
      IF  A[C] > Key  then
           E:=C-1
       Else
      IF      A[C]=Key then
          F:=True;
  End;
IF  F=False then
  Writeln(' Not Found')
Else
   Writeln(' Find in index' , C);
End.
```

سئوال) اگر **عضو تکراری در لیست باشد** و به روش **باینری** جستجو کنیم **کدام عضو** پیدا خواهد شد؟

سئوال) آرایه روبرو را برای کلید 34 به دو روش خطی و باینری جستجو کنید و تعداد مقایسه ها را در هر دو روش بدست آورید:

8	15	32	17	19	34	21	12	2	38	34	13	9	20

مرتب سـازی لیست :

مرتب سازی : منظور از مرتب سازی داده ها **تغییر دادن موقعیت آنها در یک لیست است** تا داده ها در لیست **به طور منظم صعودی یا نزولی قرار گیرند**.

صعودی ç از کمتر به بیشتر **2 , 7 , 8 , 9 , 15 , 18**

نزولی ç از بیشتر به کمتر **18 , 15 , 9 , 8 , 7 , 2**

درهمه مثالهایی که در زیر آمده است می خواهیم لیست را صعودی مرتب کنیم.

1 - روش مرتب سازی حبابـی: (Bubble Sort)

مرحله یا گذر ۱) در این روش **اولین عدد لیست با دومین عدد لیست مقایسه می شود** و اگر ترتیب آنها درست نبود،(یعنی اینکه عدد اولی از دومی بزرگتر باشد) جای آنها را عوض می کنیم.

سپس عدد دومی را با عدد سومی مقایسه می کنیم و اگر ترتیب آنها درست نبود جای آنها را عوض می کنیم. و ایـن کـار را تـا انتهـای لیست(فرض بر این است که لیست N تا عنصر دارد) ادامه می دهیم. **عدد بزرگتر به خانه N ام می رود**.(مانندیک حباب)

مرحله ۲) دوباره از خانه اول شروع می کنیم و خانه اول را با خانه دوم مقایسه می کنیم و اگر ترتیب آنهـا درسـت نبـود جـای آنهـا را عوض می کنیم و سپس عدد دوم وعدد سوم مقایسه می شوند و ..... و **این کار تا خانه N-1 ادامه پیدا می کند.**

مرحله ۳) مرحله سوم نیز مانند مرحله دوم است **با این تفاوت که تا خانه N-2 انجام می شود.**

نکته) اگر لیست داری N تا سلول باشد N-1 مرحله خواهیم داشت.

مثال) آرایه روبرو را به **روش حبابی، صعودی مرتب** کنید :

12	18	10	17	13	9

1	2	3	4	5	6
12	18	10	17	13	9
12	18	10	17	13	9
12	10	18	17	13	9
12	10	17	18	13	9
12	10	17	13	18	9
12	10	17	13	9	18

مرحله ۱ : ۵ عمل مقایسه و چهار جابجایی رخ داده است.

1	2	3	4	5	6
12	10	17	13	9	18
10	12	17	13	9	18
10	12	17	13	9	18
10	12	13	17	9	18
10	12	13	9	17	18

مرحله ۲ : چهار عمل مقایسه و سه جابجایی رخ داده است.

1	2	3	4	5	6
10	12	13	9	17	18
10	12	13	9	17	18
10	12	13	9	17	18
10	12	9	13	17	18

مرحله ۳ : سه عمل مقایسه و یک عمل جابجایی رخ داده است

1	2	3	4	5	6
10	12	9	13	17	18
10	12	9	13	17	18
10	9	12	13	17	18

مرحله ۴ : دو عمل مقایسه و یک جابجایی رخ داده است.

1	2	3	4	5	6
10	9	12	13	17	18
9	10	12	13	17	18

مرحله ۵ : یک مقایسه و یک جابجایی رخ داده است.

در این مثال مجموعا ۱۵ مقایسه و ۱۰ جابجایی صورت گرفته است.

نکته) بطور کلی در روش **مرتب سازی حبابی** :

اگر لیست دارای **N تا عنصر**باشد به **تعداد N(N-1) / 2 تا مقایسه** انجام می شود.

نکته) تعداد جابجایی ها در روش حبابی را **دقیقا نمی توان** محاسبه کرد و بستگی به بی نظمی آرایه دارد ولـی مـی تـوان گفـت کـه **حداکثر تعدادِ جابجایی ها به اندازه تعداد مقایسه ها است** یعنی هر مقایسه ایی که انجام می شود یک جابجایی نیز رخ دهد.
اگر آرایه ایی **نزولی** باشد و **بخواهیم آن را به روش حبابی صعودی کنیم** تعداد **جابجایی** ها **برابر** تعداد **مقایسه** ها است.

سئوال) اگر بخواهیم یک لیست ۱۰ عنصری را به روش **حبابی** مرتب کنیم چند **مقایسه** انجام می شود؟
الف) ۹ ب) ۱۵ ج) ۵۰ د) ۴۵

سئوال) آرایه زیر را به روش حبابی مرتب نزولی کنید، شکل آرایه در گذر چهارم چگونه است؟

12	18	9	13	15	7	10	14

مثال) برنامه ایی بنویسید که یک آرایه ۱۰ عنصری را دریافت کند و به روش حبابی، صعودی مرتب کند و چاپ کند :

```
Var
   A:Array[1..10] of integer;
  G, B , C:Integer;
Begin
  For  B:=1  To  10 do
     Readln(A[B]);
 For  G:=9   Downto  1  do
   For   B:=1   To   G  Do
     If  A[ B]   > A[B+1]  then
          Begin   { جابجایی }
               C:=A[B];
               A[B]:=A[B+1];
               A[B+1]:=C;
           End;
For  B:=1  to  10  do
    Writeln(A[B]);
End.
```

مثال) در مثال ۲ فصل رکورد در صفحه ۹۳ مربوط به مشترکین تلفن ، پروسیجری بنویسید که مشترکین را بر اساس شماره تلفن مرتب کند :

```
procedure sort;
  var
    i,j:integer;
    temp:REC;
begin
  for i:=49 downto 1 do
    for j:=1 to i do
        if M[j].number>M[j+1].number then
          begin
             temp:=M[j];
             M[j]:=M[j+1];
             M[j+1]:=temp;
          end;
end;
```

2 - روش مرتب سازی انتخـابی : (Selection)

فرض کنید می خواهیم یک لیست شامل N خانه را به روش انتخابی مرتب کنیم(صعودی):

مرحله اول) از ابتدا تا انتهای لیست را جستجو کرده و کوچکترین عدد را پیدا کرده و محل آن را با اولین عنصر لیست عوض می کنیم.

مرحله دوم) از خانه دوم تا انتهای لیست را جستجو کرده و کوچکترین عدد را پیدا کرده و محل آن را با خانه دوم عوض می کنیم.

دقت کنید که در هر مرحله محل واقعی یک از اعداد مشخص می شود.

مرحله سوم) از خانه سوم تا انتهای لیست را جستجوکرده و کوچکترین عدد را پیدا کرده و محل آن را با سومین عنـصر لیـست عـوض می کنیم.

این کار را تا انتهای لیست انجام می دهیم .

اگر **لیست دارای N تا عنصر** باشد **N-1 گذر یا فاز** خواهیم داشت و **در هر مرحله فقط یک جابجایی** رخ می دهد.

سئوال) آرایه روبرو را به **روش انتخابی**، صعودی مرتب کنید:

12	18	10	17	13	9

9	18	10	17	13	12

گذر ۱ : ۵ مقایسه و یک جابجایی ⟵

9	10	18	17	13	12

گذر ۲ : ۴ مقایسه و یک جابجایی

9	10	12	17	13	18

گذر ۳ : ۳ مقایسه و یک جابجایی ⟵

9	10	12	13	17	18

گذر ۴ : ۲ مقایسه و یک جابجایی

9	10	12	13	17	18

گذر ۵ : ۱ مقایسه و یک جابجایی

در این مثال ۱۵ عمل مقایسه و پنج جابجایی رخ داده است.

همانطور که مشاهده می کنید همین آرایه **به روش حبابی ۱۰ جابجایی** داشت ولی **به روش انتخابی پنج جابجایی** دارد.

نکته) در روش **مرتب سازی انتخابی تعداد مقایسه ها** برابراست با **N (N-1) / 2**

$$(N-1) + (N-2) + (N-3) + \ldots + 2 + 1 = N \times (N-1) / 2$$

در روش انتخابی تعداد جابجایی ها همیشه برابر N-1 تا است.

نکته) تعداد مقایسه ها در روش انتخابی و حبابی یکسان است.

نکته) تعداد **جابجایی** ها **در روش انتخابی کمتر از روش حبابی** است به همین دلیل روش انتخابی سریعتر از حبابی است.

سئوال) آرایه روبرو را به روش انتخابی مرتب کنید نزولی کنید و گذر چهارم آن را بدست آورید؟

12	18	9	13	15	7	10	14

نکته) مرتب سازی لیست های غیر عددی نظیر رشته ها **دقیقا مانند لیست های عددی** است.

مثال) برنامه ایی بنویسید که یک آرایه ۱۰ عنصری از اعداد را دریافت کندو به روش انتخابی مرتب صعودی کند و چاپ کند.

```
Var  A:Array [1 .. 10 ] of integer;
   K , L , T,Min , C:  integer;
Begin
  For  K:=1   to   10  do
       Readln( A[K]);
  For K:=1  to  9  do
     Begin
        Min:=A[K];
         C:=K;  { موقعیت عدد کوچک}
         For  L:=K+1  To 10 Do
              IF  A[L] < Min then
                   Begin
                       Min:=A[L];
                        C:=L;
                   End;
            T:=A[K];
            A[K]:=Min;   } جابجـایی
            A[C]:=T;
       End;
For L:=1 To 10 do
  Writeln(A[L);
End.
```

تست۱) اگر آرایه A یک آریه N عنصری مرتب شده نزولی باشد حداکثر چند عمـل مقایـسه و جابجـایی لازم اسـت تـا مرتـب سـازی انتخابی به صورت صعودی در بیاید؟

الف) $N(N-1)/2$ ب) $N(N+1)/2$ ج) $Log_2(N)+1$ د) N^2

تست۲) اگر آرایه S یک آرایه N عنصری باشد در این صورت حداکثر چند عمل جابجایی لازم است تا بـا مرتـب سـازی حبـابی بتـوان آرایه را به صورت صعودی مرتب نمود؟

الف) $N+ N-1 + N-2 +\ldots+2+1$ ب) $N(N-1)/2$ ج) $N(N+1)/2$ د) نامشخص است

سئوال) اگر یک آرایه 10 عنصری نامرتب داشته باشیم و بخواهیم در را به روش باینری عمل جستجو را انجام دهیم چند عمل مقایـسه انجام می شود؟

تست۳) اگرعملیات مرتب سازی حبابی را برای مرتب کردن صعودی آرایه زیر بکار ببریم پس از چند گذر آرایه مرتب می شود؟

5 , 3 , -4 , 204 , 190 , 180

الف) 3 ب) 4 ج) 5 د) 2

تست۴) در جستجوی دودویی کدام گزینه صحیح است؟

الف) سرعت جستجو نسبت به خطی کمتر است. ج) عنصر مورد جستجو با تک تک عناصر مقایسه می شود

ب) لیست مرتب شده بهتر است د) لیست حتما باید مرتب باشد

تست ۵)لیستی دارای ۱۰۰۰ عضو است حداکثر چند مقایسه برای پیدا کردن یک آیتم در لیست مورد نظر با روش دودویی لازم است؟

الف) ۱۰ ب) ۱۱ ج) ۵۰۰ د)۱۰۰۰

فایـلها (FILES)

فایل چیست ؟

فایل ، مجموعه ای از اطلاعات درباره یک موضوع است که روی **حافظه های جانبی** نظیر دیسکت و یا هارددیسک نگه داری می گـردد. **بنابراین برای ذخیره دائمی داده ها و اطلاعات از ساختاری بنام فایل استفاده می کنیم**. پس تمام برنامه هایی که تا کنون می نوشتیم و همچنین عملیات مربوط به برنامه ها در حافظه اصلی (RAM) کامپیوتر انجام می گرفت که گـذرا و فقـط بـه زمـان اجـرای برنامـه و روشن بودن کامپیوتر بستگی داشت.ولی در فایل ها چنین نیست، بلکه داده ها در فایلهایی قرار دارند که بعدا و حتـی بعـد از خـاموش کردن کامپیوتر، باز هم داده ها قابل دسترسی هستند.

در نظر بگیرید که در حال اجرای برنامه ای هستیم که ۱۰۰ عدد از ورودی دریافت می کند و ۲۰۰ عـدد در خروجـی تولیـد مـی کنـد. حال با هر بار اجرای برنامه، باید تعداد ۱۰۰ عدد بعنوان داده های ورودی به برنامه، وارد کنیم تا در خروجی ۲۰۰ عدد ببینیم کـه ایـن بسیار خسته کننده است. لذا بهتر است که این ۱۰۰ عدد از قبل یکبار در فایلی ذخیره شوند و در هنگام اجرای برنامـه فایـل از ورودی خوانده شود و همچنین اطلاعات خروجی نیز در فایل تولید شود و بدین ترتیب داده ها ماندگار شوند و از بین نیز نروند.

بیشتر بدانیم ...

هر فایل دارای یک نام می باشد . در سیستم عامل DOS نام یک فایل حداکثر از ۸ حرف تشکیل شده و ممکن است دارای پـسوند بـا حداکثر ۳ حرف باشد .

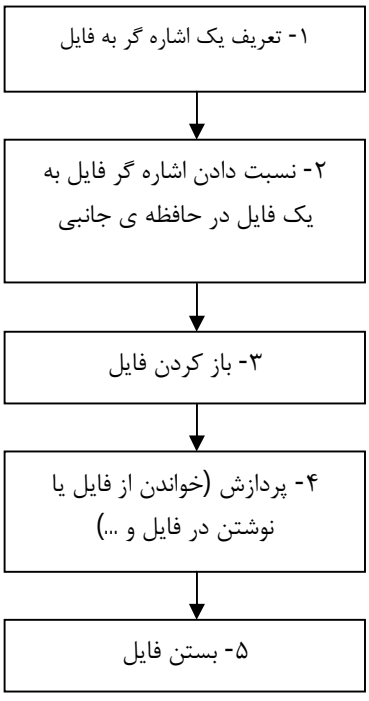
در زبان پاسکال **سـه نوع فایل** زیر وجود دارند که در این فصل دستورات مربوط به هر یک را بررسی می کنیم :

1 - **فایلهای متنی** - Text - (**ترتیبی**)

2 - **فایلهای نوع دار** - Typed File - (**باینری یا دودویی یا تصادفی**)

3 - **فایلهای بدون نوع** - Untyped File

برای کار با هر نوع فایل در توربـو پاسکال باید روند زیر انجام گیرد :



۱- برای کار با هر نوع فایل ابتدا باید متغیری از نوع فایل موردنظر ایجاد نماییم.
بعنوان مثال برای فایلهای متنی، تعریف متغیر(یا اشاره گر) چنین است :

```
Var
F:Text ;
```

نکته) شناسه استاندارد Text نوع فایلهای متنی را مشخص می کند.

۲ - بعد از تعریف اشاره گر، باید آنرا با نام یک فایل در هارد دیسک مرتبط کنـیم، یـا بعبارت دیگر باید آنرا به یک فایل در حافظه جانبی نسبت دهیم. این کـار بـا دسـتور Assign انجام می شود.بعنوان مثال برای نسبت دادن اشاره گـر F بـه فایـل متنـی Student.txt در درایو C کامپیوترمان به صورت زیر عمل می کنیم :

```
VAR
  F : TEXT;
BEGIN
  ASSIGN( F, ' C:\Student.TXT ' );
```

نکته) دستور Assign فقط کافی است یکبار در برنامه نوشته شود.

بعد از اجرای دستور Assign ،برای انجام عمل خواندن یا نوشتن و یا تغییـر دادن در فایل از اشاره گر F استفاده می کنیم.بعبارت دیگر هر عملی که بر روی F انجام شـود در واقع بر روی فایل Student.txt انجام می شود.

مراحل ۳ و ۴ و۵ برای فایلها دقیقا مانند این است که بخواهیم یک پرونده یا کتابی را ابتدا باز کرده و سپس مطالبی را از آن خوانـده و یا در آن اضافه، حذف و یا تغییر داده ودر انتها پرونده یا کتاب را ببندیم.

1- فایلهای متنـــی :

فایل متنی، فایلی است که از تعدادی **خط** شامل **کاراکترهایی** با کد اسکی تشکیل شده است. **هر خط به کاراکترهای CR و LF ختم می گردد**.ترکیب CR و LF بعنوان **جدا کننده خطوط** مورد استفاده قرار می گیرد. **فایلهای متنی معمولا دارای پسوند TXT مـی باشند**.

هر اطلاعاتی که داخل فایلهای متنی ریخته شود به رشته تبدیل می شود ؛ به همین دلیل فایلهای متنی نسبت به دو نوع دیگـر کنـدتر هستند.

عمل باز کردن فایل **متنـی** برای یکی از ۳ هدف زیر انجام می شود :

۱ - ایجاد کردن یک فایل جدید

۲ - خواندن محتوای یک فایل موجود

۳ - اضافه کردن داده به انتهای یک فایل متنی

در تورپو پاسکال برای هر یک از عملیات باز کردن فایل **متنی** ، یک دستور وجود دارد. **برای ایجاد کردن یک فایـل جدیـد از دسـتور Rewrite وبرای خواندن از یک فایل موجـود از دستور Reset و برای اضافه کردن داده به انتهای یـک فایـل متنـی از دسـتور Append استفاده می کنیم**.

1 - **دستور Rewrite** : اگر به این روش فایل متنی را باز کنیم ، **فایل متنی ساخته می شود و فقط می توانیم داخل آن اطلاعات بریزیم**،در صورتی که فایل از قبل موجود باشد و آن را با دستور Rewrite باز کنیم ، فایل را پاک کرده و از نو می سازد.

2 – **دستور Reset** : اگر فایل متنی را به این روش باز کنیم **فایل را فقط می توانیم بخوانیم** و نمی تـوانیم در آن اطلاعـات بریـزیم. نکته مهمی که باید به آن توجه داشته باشیم این است که **در این روش باید از قبل فایل موجود باشد**.اگر فایـل موجـود نباشـد و مـا بخواهیم **فایل وجود نداشته ! را باز کنیم** قطعا برنامه Error میدهد.

3 – **دستور Append** : با باز کردن فایل متنی به این **روش فقط می توانیم به انتهای فایل از قبل موجود ، اطلاعات اضافه کنیم** . یعنی **در این روش نیز فایل باید از قبل وجود داشته باشد**. همچنین اطلاعات را نیز **نمی توان خواند**.

نکته) در هر لحظه فقط باید از یکی از ۳ روش بـالا اسـتفاده کـرد.یعنـی نمـی تـوانیم فایـل را هـم Reset کنـیم و هـم Rewrite کنیم؛بلکه باید فایل را بست و دوباره به روش دیگر باز کرد.

بوسیله دستورات readln , read می توان اطلاعات را از فایل خواند.وبرای نوشتن اطلاعـات در فایـل از Write , Writeln اسـتفاده می کنیم.(البته به شرط باز بودن فایل می توانیم از فایل بخوانیم و یا در فایل بنویسیم).

نکته) تفاوت Read با Readln و هم چنین Write با Writeln مانند قبل است.

هنگامیکه فایلی را به روش Reset باز می کنیم یک مکان نمای فرضی در ابتدای فایل قرار می گیرد ؛ با هر بار خواندن اطلاعات فایـل ، مکان نمای فرضی یکی به جلو حرکت می کند.

اگر فایلی را به روش Append باز کنیم مکان نمای فرضی در انتهای فایل قرار می گیرد که با هر بار نوشتن اطلاعات در فایل ، مکـان نما یکی به جلو حرکت می کند.

برای بستن فایل از دستور Close استفاده می کنیم.

دستور close باعث save شدن فایل می شود ؛ **البته اگر فایل را ببندیم ارتباط اشاره گر با آن فایل قطع نمی شود و برای باز کردن مجدد آن فایل نیازی به استفاده از دستور Assign نیست**.

حال به مثالی که در زیر نوشته شده است توجه کنید و به نحوه ی تعریف اشاره گر و بازکردن فایل و **پردازش فایل** و بـستن آن دقـت کنید :

مثال ۱) برنامه ای بنویسید که تعدادی عدد از کاربر دریافت کند و سپس داخل یک فایل متنی بریزد،پایان بخش داده ها صفر است و همچنین بوسیله یک پروسیجر اعداد داخل فایل را خوانده و جمع و تعداد اعداد را در **مانیتور** چاپ کند :

```
var
   f:text;
   n:integer;
procedure sum;
  var
    x,s,i:integer;
begin
   assign(f,'c:\m.txt');
   reset(f);
   i:=0; s:=0;
   while not eof(f) do
     begin
       Readln ( f , x );
       s:=s+x;
       i:=i+1;
     end;
   close(f);
   writeln(s);
   writeln(i);
end;
begin
  assign(f,'c:\m.txt');
  rewrite(f);
  repeat
     readln(n);
     if n=0 then break;
     writeln ( F , n );
  until false;
 close(f);
 sum;
end.
```

سوال) حال اگر بخواهیم در برنامه ی بالا در انتهای فایل دو متغیر S(جمع اعداد) و I (تعداد اعداد) اضافه شود؛ چه تغییـری بایـد در برنامه بالا بدهیم یا بعبارت دیگر چه دستورهایی باید به قسمت پروسیجر برنامه بالا اضافه کنیم ؟

مثال ۲) برنامه ای بنویسید که محتوای فایل m.txt را در فایل p.txt کپی کند :

```
var
  f1,f2:text;
  c:char;
begin
  assign(f1,'c:\m.txt');
  assign(f2,'c:\p.txt');
  reset(f1);
  rewrite(f2);
  while not eof(f1) do
   begin
     read(f1,c);
     write(f2,c);
   end;
  close(f1);
  close(f2);
end.
```

سوال) اگر بجای دستور Read(f1,c) دستور Readln(f1,c) نوشته شده بود ؛ محتوای فایل F2 چه تغییری می کرد ؟

تابع بولی Eof (End of file) :

برای اطلاع از رسیدن به انتهای فایل استفاده میشود. اگر به انتهای فایل برسیم یا بعبارتی تمام خطوط برنامه خوانده شده باشد ایـن تابع مقدار True وگرنه مقدار False را بر می گرداند.

نکته) در فایلهای متنی برای اینکه متوجه بشویم که **به انتهای یک خط رسیده ایم یا نه از تابع EOLN** استفاده می کنیم .

نکته) در فایلها ی متنی هنگام خواندن اطلاعات فایل برای اینکه متوجه بشویم که به انتهـای فایـل رسـیده ایـم یـا نـه از تـابع Eof استفاده می کنیم. مثلا در برنامه ی بالا برای خواندن تمام اطلاعات از فایل m.txt و ریختن آن به فایل p.txt ابتدا باید تمام محتـوای فایل اولی خوانده شود ،که برای اینکار از حلقه ی while ای استفاده کرده ایم که شرط آن not eof(f1) است ؛ یعنی تا زمانی که بـه انتهای فایل نرسیده ایم اطلاعات را از فایل بخوان.که مسلما وقتی به انتهای فایل برسیم شـرط حلقـهTrue میـشود و از حلقـه خـارج میشویم.

مثال ۳) با استفاده از فایلها برنامه ای بنویسید که :

- با زدن **کلید A** یک دانش آموز با اطلاعات : نام ، فامیلی و معدل ، به فایل اضافه شود.
- با زدن **کلید L** لیست دانش آموزان داخل فایل بر روی **مانیتور** نمایش داده شود.
- با زدن **کلید S** یک نام و فامیلی دریافت کند و آنرا در فایل جستجو کند(جستجو به روش خطی) ؛ اگر در فایل موجود بود اطلاعـات کامل آن دانش آموز را چاپ کند و اگر موجود نبود پیغام Not Found را در مانیتور چاپ کند.
- با زدن **کلید Q** از برنامه خارج شویم.

همچنین دانش آموزانی را که معدلشان کمتر از ۱۰ است در فایلی بنام Mardud.txt بریزد.

```
var
  c:char;
  F:text;
procedure add;
 var
   n,m:string[20];
   avg:integer;
begin
   assign(F,'c:\std.txt');
   writeln('Enter name & Family & avg:');
   readln(n);
   readln(m);
   readln(avg);
   append(F);
   writeln(F,n);
   writeln(F,m);
   writeln(F,avg);
   close(f);
end;
{++++++++++++++}
procedure list;
 var
  n,m:string[20]; avg:integer;
 begin
   assign(F,'c:\std.txt');
   writeln(' List File ');
   reset(F);
   while not eof(f) do
     begin
       readln(F,n);
       readln(F,m);
       readln(F,avg);
       writeln(n ,'  ',m,'   ',avg);
     end;
   close(F);
end;
{-----------------}
procedure search;
 var
   x,p,n,m:string[20];
   avg:integer;    find:boolean;
begin
   assign(F,'c:\std.txt');
   writeln('name & family ');
   readln(x);
   readln(p);
   find:=false;
   reset(F);
```

{ادامه برنامه در صفحه بعد}

```
  while not eof(f) do
      begin
        readln(F,n);
        readln(F,m);
        readln(F,avg);
        if (x=n) and(p=m) then
           begin
             writeln(x , '   ', p,'   ',avg);
             find:=true;
             break;
           end;
      end;
    close(F);
    if find=false then
      writeln(' not found');
end;
 {============}
procedure mardud;
  var
    n,m:string[20];
    avg:integer; f1:text;
begin
   assign(f1,'c:\mardud.txt');
   assign(f,'c:\std.txt');
   reset(f);
   rewrite(F1);
   while not eof(F) do
     begin
       readln(F,n);
       readln(F,m);
       readln(F,avg);
       if avg<10 then
         begin
           writeln(F1,n);
           writeln(F1,m);
           writeln(F1,avg);
         end;
     end;
     close(F); close(F1);
end;
begin
   repeat
      writeln(' A or S or L or Q ?');
      readln(c);
      if c='A' then add;
      if c='L' then list;
      if c='S' then search;
   until c='Q';
  mardud;
end.
```

فایل های نـوع دار :

فایل نوع دار حاوی اطلاعاتی از یک نوع داده می باشد. مثلا فایلی از نوع Integer ، فقط حاوی اعـداد صـحیح و یـا فـایلی از نـوع Real ، فقط حاوی اعداد اعشاری است. نوع فایل می تواند هر نوع داده ، حتی انواع داده ساختاری نظیر رکورد نیز باشـد.ویژگـی دیگـر فایلهای نوع دار این است که همهی اطلاعات در یک خط ذخیره می شوند. عملیات خواندن و نوشتن در فایلهای نوع دار بسیار سـریعتر از فایلهای متنی انجام می شود. زیرا در فایلهای نوع دار ، اطلاعات با همان الگویی که در حافظه قرار دارند ، بدون هیچ گونه تبـدیلی در فایل ذخیره می شوند. بعنوان مثال ، یک عدد صحیح نوع Integer در هنگام ذخیره در فایل نوع دار ، فقط دو بایت را اشغال می کنـد یعنی به همان صورت که در حافظه اصلی قرار دارد اما در هنگام ذخیره در فایل متنی به ازای هر رقم یک بایت را اشغال می نماید زیرا هر رقم آن باید به یک کد اسکی تبدیل شود. بنابراین یک تا شش بایت را اشغال می کند.

روش کار کردن با فایلهای نوع دار مانند فایلهای متنی است یعنی باید ابتدا متغیری به عنـوان اشـاره گـر فایـل نـوع دار تعریـف و سپس آنرا با دستور Assign به یک نام فایل در حافظه جانبی نسبت دهیم.

نکته) معمولا پسوند فایلهای نوع دار را Dat قرار می دهیم که با فایلهای متنی اشتباه نشوند.

بعنوان مثال برای ذخیره تعدادی عدد اعشاری در یک فایل نوع دار اشاره گر فایل اینگونه خواهد بود :

```
Var
F : FILE OF Real ;
```

یا برای ذخیره نام و نام خانوادگی و نمره دانش آموزان اشاره گر فایل و دستور Assign بصورت زیر تعریف می شوند :

```
TYPE
  Student = RECORD
     Name : STRING[20] ;
     Family : STRING[20] ;
     Grade : Real ;
   END ;
VAR
    F : FILE OF   student ;
BEGIN
  Assign ( F, 'c:\school info.dat' );
END.
```

در فایلهای نوع دار از **دستور Rewrite به منظور باز کردن یا ایجاد یک فایل جدید و خالی از اطلاعات (مانند فایلهای متنی)** و از دستور **Reset به منظور خواندن و یا نوشتن** در یک **فایل موجود** استفاده می شود.

نکته) دستور Append برای فایلهای نوع دار قابل استفاده نمی باشد.

پس از باز کردن فایل نوع دار ، که با یکی از دو دستور Rewrite یا Reset انجام می شود جهت نوشـتن و خوانـدن اطلاعـات در ایـن نوع فایلها از دستورات Write و Read استفاده می گردد. ودر نهایت برای بستن فایل نیز ، از Close استفاده می کنیم.

نکته) در فایلهای نوع دار نمی توانیم از دستورات Readln و Writeln استفاده کنیم.

مکان نمای فایل :

در هر لحظه تنها یکی از رکوردهای فایل نوع دار قابل دستیابی است. هنگامیکه یک فایل را باز می کنیم اولین رکـورد و بـا اجـرای هـر دستور Read یا Write ، رکورد بعدی قابل دسترسی خواهد بود. برای مـشخص شـدن ایـن کـه در حـال حاضـر کـدام رکـورد قابـل دستیابی است شماره رکورد مزبور در داخل متغیری بنام مکان نمای فایل که فیلدی از متغیر فایل است نگـه داری میگـردد. بـا اجـرای دستور Reset یا Rewrite در مکان نمای فایل عدد صفر قرار میگیرد و اجرای دسـتورات Read یـا Write سـبب افـزایش مقـدار مکان نمای فایل نیز می شود که اصطلاحا گفته می شود مکان نمای فایل جلو می رود.

برای اینکه بتوان یک رکورد دلخواه از فایل نوع دار را پردازش کرد لازم است مکان نمای فایل را به ابتدای رکورد دلخواه ببریم ؛ برای اینکار از پردازه استاندارد Seek استفاده می کنیم :

Seek (f , n) ;

منظور از f مکان نمای فایل است و n شماره رکورد مورد نظر است. شماره اولین رکورد صفـر است.

مثال) دستورات لازم برای دسترسی به عنصر دهم فایل school info.dat را بنویسید :

```
Assign (f , 'c:\school info.dat' ) ;
Reset (f) ;
Seek ( f , 9 );
```

نکته) برای اطلاع از تعداد رکورد ها یا **تعداد عناصر یک فایل نوع دار از تابع استاندارد FileSize** استفاده می کنیم.

مثال) دستورات لازم برای دسترسی به آخرین عنصر فایل school info.dat را بنویسید :

```
Assign (f , 'c:\school info.dat' ) ;
Reset (f) ;
Seek ( f , FileSize (f) - 1 );
```

مثال) دستورات لازم برای اضافه کردن یک عنصر به انتهای فایل school info.dat را بنویسید :

```
Assign (f , 'c:\school info.dat' ) ;
Reset (f) ;
Seek ( f , FileSize (f) );
```

برای اضافه کردن یک عنصر به انتهای فایل ، کافی است مکان نمای فایل را بعد از آخرین عنصر قرار دهیم و سپس **دسـتور Write** را اجرا کنیم.

نکته) اگر با دستور seek اشاره گر فایل را از محدوده فایل خارج کنیم پیغام خطای شماره ۱۰۰ یعنی disk read error صادر می شود.

Seek (f, filesize(f) +1) ž **غلط**

Seek (f,-1) ž **غلط**

نکته) تابع استاندارد filepos شماره رکورد جاری را بر می گرداند.

```
Reset(f);
Write(filepos(f)); ž 0
Read(f,a);
Write(filepos(f)); ž 1
```

مثال ۴) با فرض وجود یک فایل نوع دار از نوع Integer برنامه ای بنویسید که :

الف) یک عدد به انتهای فایل اضافه کند .

ب) اعداد را از آخر به اول در مانیتور چاپ کند .

```
Var
  F: file of integer ;
  n , i , x : integer ;
Begin
    Assign ( F, 'c:\mydigit.dat' ) ;
    Rest (f) ;
    Seek (f , filesize(f) );
    Readln (x);
    Write (f , x );
    Close (f);
```

{ قسمت ب برنامه }

```
Reset (f) ;
n:= filesize(f) ;
for   i:= n – 1      downto      0      do
      begin
           seek ( f , i ) ;
           read ( f , x ) ;
           writeln (x) ;
      end ;
 close(f) ;
End .
```

مثال ۵) برنامه ای بنویسید که تعدادی عدد از کاربر دریافت کند و آنرا در فایل بریزد.(پایان بخش دادهـا (آخـرین عـدد ورودی) عـدد صفر است).همچنین با استفاده از پروسیجر Sum اعداد موجود در فایل را با هم جمع کند و در متغیری بنام S بریزد و همچنین تعـداد اعداد را در متغیری بنام I بریزد و سپس S و I را در مانیتور چاپ کند.

```
Var
 F: file of integer ;          X: integer ;
Procedure   Sum ;
Var
   s , i , x : integer ;
Begin
 Assign ( F, 'c:\mydigit.dat' ) ;
 Reset (f) ;
 s:= 0 ;    i:= 0 ;
 while   not eof(f)   do
    begin
       read( f , x ) ;
       s:= s + x ;        i :=  i + 1 ;
    end ;
close (f) ;
writeln ( s ) ;     writeln ( i ) ;
end;
Begin
   Assign ( F, 'c:\mydigit.dat' ) ;
   Rewrite (f) ;
   Repeat
        Read (x) ;
        Write (f , x) ;
    Until x = 0 ;
    Close (f) ;
    Sum ;
End.
```

فایل بدون نـوع:

فایل بدون نوع فایلی است که به برنامه نویس اجازه می دهد **در مورد داده های درون فایل هرگونه تفسیر دلخواهی** داشته باشد در این نوع فایل ها برنامه نویس می تواند با یک دستور خواندن و نوشتن حجم دلخواهی از داده ها را بخواند یا بنویسد. ایـن نـوع فایـل انعطاف زیاد وسرعت بسیار بالایی دارد.

نحوه تعـریف اشاره گر فایل:

```
Var
      F : file ;
```

بعد از کلمه file ، نوع داده ای نباید معین گردد.

با دستور assign اشاره گر فایل را نسبت می دهیم.

برای باز کردن فایل بدون نوع از دو پردازه rewrite و reset مانند فایل های نوع دار استفاده کرده با این تفـاوت کـه پـارامتر دیگـری بعنوان اندازه ی داده باید ذکر کنیم.

دستور **reset(f,10);** فایل f را برای خواندن ونوشتن باز می کند با این فرض که فایل مزبور از داده هایی به اندازه ۱۰ بایت تـشکیل شده است.

دستور **rewrite(f,1);** فایل جدیدی ایجاد می کند که هر داده ای که داخل آن می ریزیم ۱ بایت فضا اشغال می کند.

نحوه خواندن و نوشتن در فایل بدون نـوع :

برای **نوشتن و خواندن** در فایلهای **بدون نوع** از دستورهای **BlockRead و BlockWrite** با پارامترهایی بصورت زیر استفاده می شود :

```
BlockWrite ( F ,  B  ,  N , W ) ;
BlockRead ( F , B , N , R ) ;
```

پارامترهای دستـور BlockWrite :

F: اشاره گر فایل

B: بافر که معمولا آرایه ای از نوع char است.

N: یک عدد صحیح است که نشان دهنده تعداد داده هایی است که قرار است از بافر (B) در فایل ریخته شود.

W: متغیری صحیح است که بعد از عملیات نوشتن در فایل مشخص میکند چه مقدار از داده ها با موفقیت درفایل ریخته شده اند.

- اگر بعد از اجرای این دستور N=W باشد یعنی همه داده ها با موفقیت در فایل ریخته شده اند.

پارامترهای دستـور BlockRead :

F: اشاره گر فایل

B: بافر که معمولا آرایه ای از نوع char است.

N: یک عدد صحیح است که نشان دهنده تعداد داده هایی است که قرار است از فایل خوانده و دربافر (B) ریخته شود.

R: متغیری صحیح است که بعد ازعملیات خواندن از فایل مشخص می کند چه مقدار از داده ها با موفقیت ازفایل خوانده شده اند.

- اگر بعد از اجرای این دستور N=R باشد یعنی همه داده ها خوانده شده است واگر R<N باشد یعنی به انتهـای فایـل رسـیده ایم.

با دستور **Close** فایل **بسته** خواهد شد.

معمولا **فایل های بدون نوع** جهت **کپی برداری یا ایجاد نسخه پشتیبان از فایل** به کار می رود.