LEARNING C++

http://Sstar5programs.blogfa.com

آموزش C++

سری مقالات الگوریتم نویسی :مقاله شماره 2

"Learning c++" Articles

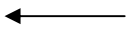
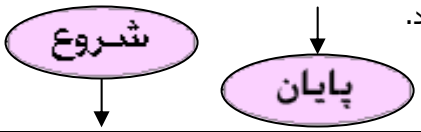
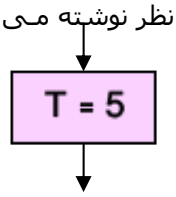
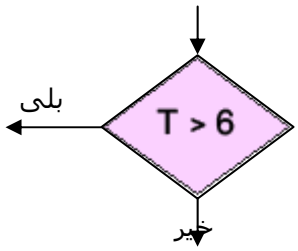
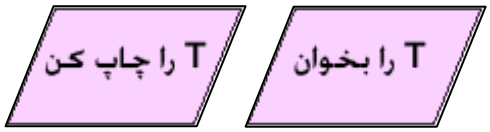
By S.Rezaei

مقاله های آموزشی سی پلاس پلاس
س.رضایی

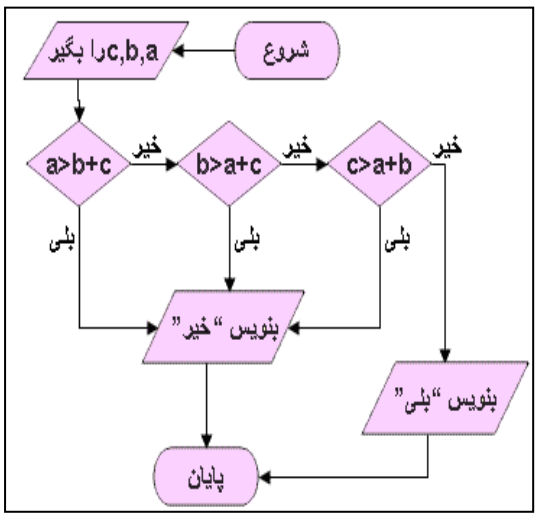
رسم فلوچارت

در عمل برای نمایش الگوریتم از یک **«فلوچارت»** (شمای جریان عملیات) استفاده می شود. در حقیقت فلوچارت روش تصویری و استاندارد نمایش الگوریتم است.

در رسم فلوچارت علائم و نمادهای استانداردی به کار می رود که هر کدام معنی خاص خود را دارند.

از علامت فلش (↑↓→ ←) برای نشان دادن روابط بین قسماتها استفاده می شود 
از شکل بیضی افقی برای شروع و پایان عملیات استفاده می شود. 
از مستطیل برای نمایش مراحل پردازشی استفاده می شود و در داخل آن عمل مورد نظر نوشته می شود. این نماد ممکن است چندین ورودی داشته باشد ولی تنها یک خروجی دارد. 
از نماد لوزی برای نشان دادن مراحل تصمیم گیری استفاده می گردد و شرط یا سؤال مورد نظر در داخل لوزی نوشته می شود. (مهم نیست که فلش های بلی و خیر از کدام سمت خارج شوند) 
از متوازی الاضلاع برای نشان دادن ورودی یا خروجی استفاده می شود. 

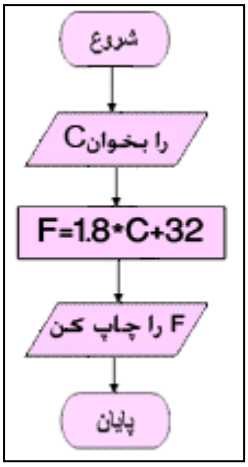
فرض کنید می خواهیم فلوچارت الگوریتم اضلاع مثلث که در مقاله قبلی داشتیم ، رسم کنیم.
این فلوچارت به این شکل خواهد بود :



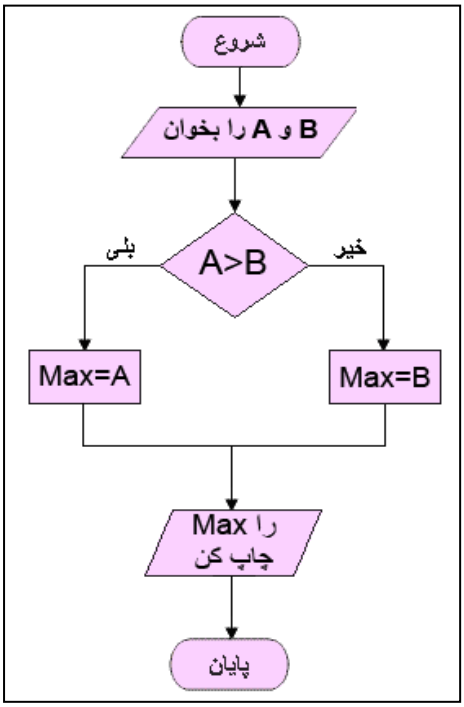
همانطور که می دانید از فرمول مقابل برای تبدیل درجه حرارت بر حسب سانتیگراد به فارنهایت استفاده می شود:

$$F = (1.8 \times C) + 32$$

حالا فلوچارت الگوریتم تبدیل درجه سانتیگراد به فارنهایت را رسم می کنیم:



فلوچارت برای مشخص کردن عدد بزرگتر بین دو عدد ورودی:



نکته) بهتر است قبل از رسم کرن فلوچارت ، الگوریتم مربوط به آن را بنویسیم.

آرایه ها

در زمانی که ما نیاز به نگهداری تعداد زیادی از مقدارها داریم ، به جای اینکه تعداد زیادی متغیر ایجاد کنیم ، یک **«آرایه»** ایجاد می کنیم. آرایه (یک بعدی) مانند ردیفی از اعداد یا همان مقادیر مختلف هستند.
(به آرایه **«لیست»** یا **«متغیر اندیس دار»** هم گفته می شود)
متغیرهای آرایه را **«عناصر»** آن آرایه می گویند.
برای اشاره به یک عنصر از آرایه ، ابتدا نام آرایه و بعد در کروشه ([ ]) یا پرانتز شماره عنصر مورد نظر را می نویسیم.
البته در الگوریتم نویسی بیشتر از پرانتز استفاده می شود ولی در کد های زبان سی پلاس پلاس ، از کروشه استفاده می شود.
به عنوان مثال آرایه ای به نام a دارم و می خواهیم اعداد 1 تا 7 را در خانه های آن ذخیره کنیم:
نکته) اندیس آرایه از 1 شروع می شود. یعنی عنصر های یک آرایه که دارای k خانه می باشد ، به شکل a[1] ، a[2] ، ... و a[k] مشخص می شوند.
عناصر آرایه ، در محل های متوالی حافظه و به شکل خاصی ذخیره می شوند.

```
1-  شروع
2-  آرایه a به طول 10 را ایجاد کن
3-  a[0]=1 , a[1]=2 , a[2]=3 , a[3]=4 , a[4]=5 , a[5]=6 , a[6]=7
4-  پایان
```

همانطور که در مقاله قبل آموختیم ، برنامه بالا را می توان بدین صورت نیز نوشت :

```
1-  شروع
2-  آرایه a به طول 10 را ایجاد کن
3-  I=0
4-  I=I+1
5-  a[I]=I+1
6-  اگر I<6 ، برو به 4
7-  پایان
```

البته نوشتن الگوریتم بصورت دوم برای تعداد زیاد داده بسیار بهتر است. فرض کنید اگر می خواستیم اعداد 1 تا 100 را داخل آرایه بریزیم ، باید در خط 3 روش اول ، 100 مقداردهی مختلف انجام می دادیم.

مثال) الگوریتمی بنویسید که 10 عدد خوانده و در آرایه نوشته و آرایه را بصورت معکوس (از آخرین عنصر به اولین عنصر) در خروجی چاپ کند.

```
1-  شروع
2-  آرایه L را به طول 10 ایجاد کن
3-  X=0 ، i=1
4-  تا زمانی که i<=10 ، اعمال شماره 5 تا 7 را انجام بده
5-  X را بخوان
6-  L(i)=X
7-  i=i+1
8-  i=10
9-  تا زمانی که i>=1 ، اعمال 10 و 11 را انجام بده
10- L(i) را چاپ کن
11- i=i-1
12- پایان
```

مثال) الگوریتمی بنویسید که 5 عدد از ورودی خوانده ، در آرایه ای قرار دهد و سپس آنها را به طور معکوس در آرایه 5 عنصری دیگری قرار می دهد.

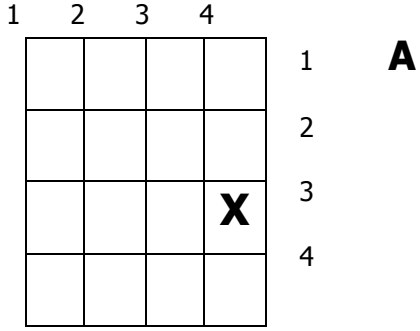
```
1-  شروع
2-  آرایه های A و B را به طول 10 ایجاد کن
3-  I=0 , J=1
4-  I=I+1
5-  A(I) ار بخوان
6-  اگر I<5 برو به 4
7-  B(J)=A(I)
8-  J=J+1 , I=I-1
9-  اگر I>0 برو به 7
10- پایان
```

مثال) الگوریتمی بنویسید که 10 عدد از ورودی خوانده و در آرایه قرار می دهد و سپس عدد دیگری می خواند و اعلام می کند که عدد خوانده شده چند بار در آرایه ظاهر شده است.

```
1-  شروع
2-  آرایه A را به طول 10 ایجاد کن
3-  I=1 , C=0
4-  تا زمانی که I<=10 مراحل 5 و 6 را انجام بده
5-  A(I) را بخوان
6-  I=I+1
7-  X را بخوان
8-  I=1
9-  تا زمانی که I<=10 مراحل 10  و 11 را انجام بده
10- اگر A(I)==X ، C=C+1
11- I=I+1
12- C را چاپ کن
13- پایان
```

آرایه های چند بعدی

همانطور که دیدید ، آرایه ها در برنامه نویسی کاربرد فراوانی دارد.آرایه هایی که در قبا مطالعه کردیم ، دارای یک اندیسی بودند. (همانند دنباله ها در یاضیات)
اما آرایه هایی نیز وجود دارند که دارای بیش از یک بعد هستند.به این آرایه ها **«آرایه های چند بعدی»** گفته می شود.
وقتی شما نیاز داشته باشید اعداد یک جدول را در آرایه نگهداری کنید ، می توانید از **«آرایه های دو بعدی»** استفاده کنید. این آرایه ها دارای دو اندیس می باشند. (مانند ماتریس ها در ریاضیات)
آرایه های دو بعدی مانند مستطیلی از متغیر ها هستند . از آرایه های دو بعدی می توان صفحه شطرنج یا صفحه بازی مار و پله را نام برد.
همانطور که گفتیم آرایه های دو بعدی دارای دو اندیس می باشند.یکی نشان دهنده شماره ردیف و دیگری نشان دهنده شماره ستونی است که عنصر مورد نظر در آن قرار دارد. در الگوریتم نویسی عنصری که در ردیف I و ستون J قرار دارد را با A(I,J) نشان می دهند که مانند آرایه های یک بعدی ، A نام آرایه ، I شماره ردیف و J شماره ستون است.
به عنوان مثال در آرایه 4×4 زیر ، خانه مشخص شده با A(3,4) مشخص می شود:



مثال) الگوریتمی بنویسید که جدول ضرب اعداد 1 تا 10 را در آرایه ای دو بعدی قرار دهد و آن را چاپ کند:

```
1-  شروع
2-  آرایه دوبعدی A با 10 سطر و 10 ستون ایجاد کن
3-  I=1 , J=1
4-  J=1
5-  A(I,J)=I*J
6-  A(I,J) را چاپ کن
7-  J=J+1
8-  اگر J<=10 ، برو به 6
9-  I=I+1
10- اگر I<=10 ، برو به 4
11- پایان
```

مثال) الگوریتمی بنویسید که جدولی 5×5 ایجاد کند و در عنصر سطر I و ستون J عدد I+J را قرار دهد:

```
1-  شروع
2-  آرایه دوبعدی A با 5 سطر و 5 ستون ایجاد کن
3-  I=1 , J=1
4-  A(I,J)=I+J
5-  J=J+1
6-  اگر J<=10 برو به 4
7-  I=I+1
8-  اگر I<=10 برو به 4
9-  پایان
```

مثال) الگوریتمی بنویسید که دو متغیر m و n را از ورودی بخواند و ماتریس همانی $m\times n$ را ایجاد کند:
(ماتریس همانی ماتریسی است که عناصری که بصورت A(I,I) هستند (شماره سطر و ستون آنها برابر است و بقیه عناصر صفر هستند.)
به عنوان مثال ماتریس همانی 3×3 به اینصورت است:

1	0	0
0	1	0
0	0	1

```
1-  شروع
2-  M=0 , N=0 , I=1 , J=1
3-  M و N را از ورودی دریافت کن
4-  ماتریس A را با M سطر و N ستون ایجاد کن
5-  اگر I برابر با J بود A(I,J)=1 ، در غیر اینصورت A(I,J)=0
6-  J=J+1
7-  اگر J<=10 برو به 5
8-  I=I+1
9-  اگر I<=10 برو به 5
10- پایان
```

مثال) الگوریتمی بنویسید 4 آرایه یک بعدی 4 عنصری را به ترتیب از ورودی بخواند و آنها را در یک جدول 5×5 قرار دهد:

```
1-  شروع
2-  آرایه های A و B و C و D را به طول 10 ایجاد کن
3-  I=1
4-  تا زمانی که I<=4 ، دستورات 5 و 6 را انجام بده
5-  A(I) را بخوان
6-  I=I+1
7-  I=1
8-  تا زمانی که I<=4 ، دستورات 9 و 10 را انجام بده
9-  B(I) را بخوان
10- I=I+1
11- I=1
12- تا زمانی که I<=4 ، دستورات 13 و 14 را انجام بده
13- C(I) را بخوان
14- I=I+1
15- I=1
16- تا زمانی که I<=4 ، دستورات 17 و 18 را انجام بده
17- D(I)  را بخوان
18- I=I+1
19- I=1
20- آرایه دو بعدی E با 4 سطر و 4 ستون ایجاد کن
21- J=1
22- اگر I برابر 1 بود ، تا زمانی که J<=4 E(I,J)=A(I) را بخوان و J=J+1
    در غیر اینصورت اگر I برابر 2 بود ، تا زمانی که J<=4 E(I,J)=B(I)  و J=J+1
    در غیر اینصورت اگر I برابر 3 بود ، تا زمانی که J<=4 E(I,J)=C(I)  و J=J+1
    در غیر اینصورت اگر I برابر 4 بود ، تا زمانی که J<=4 E(I,J)=D(I)  و J=J+1

23- I=I+1
24- اگر I<4 برو به 22
25- پایان
```

زیر الگوریتم ها

در الگوریتم هایی که تا کنون می نوشتیم ، در بعضی مواقع نیاز داشتیم تا از یک عمل در چند جای مختلف الگوریتم استفاده کنیم و یا یک یا چند عمل به دفعات زیادی تکرار می شدند. (مانند مثال قبل که نیاز داشتیم 4 بار از عمل خواندن آرایه استفاده کنیم)
برای استفاده کردن از یک عمل در یک الگوریتم که چندین بار در آن الگوریتم تکرار شده است ، می توان یک الگوریتم دیگر نوشت و آنرا با الگوریتم اصلی ترکیب کرد. به این الگوریتم ها که برای حل بخشی از مساله نوشته می شوند ، **«زیر الگوریتم»** گفته می شود.

مزایای استفاده از زیر الگوریتم ها:

- می توان یک الگوریتم را به چند بخش تقسیم کرد و این امکان بوجود می آید که بصورت گروهی الگوریتم را نوشت.یعنی هر فرد قسمت مخصوص به خود را بنویسد.
- از نوشتن قسمت های تکراری در الگوریتم جلوگیری می کند.
- اشکال زدایی و عیب یابی الگوریتم را آسان تر می کند.
- می توان از یک زیر الگوریتم در چند الگوریتم استفاده کرد.

زیر الگوریتم ها مانند متغیر ها و آرایه ها دارای یک اسم می باشند. هر زیر الگوریتم دارای چند ورودی است که آنها را از الگوریتم اصلی در یافت می کند. همچنین دارای یک یا چند خروجی است که در زمان فراخوانی باز می گرداند. همچنین یک زیر الگوریتم می تواند هیچ مقداری را بر نگرداند.به زیر الگوریتم هایی که فقط یک مقدار را بر می گردانند **«زیر الگوریتم های تابع» (function)** و به زیر الگوریتم هایی که مقداری بر نمی گردانند یا بیش از یک مقدار به الگوریتم اصلی بر می گردانند ، **«زیر الگوریتم های زیرروال» (subroutine)** گفته می شود.

به عنوان مثال می خواهیم زیر الگوریتمی بنویسیم که دو عدد را از الگوریتم اصلی گرفته و اولی را به توان دومی برساند:

زیر الگوریتم Pow(a,b)
1- P=1;
2- تا زمانی که b>0 دستورات 3 و 4 را انجام بده
3- P=P*a
4- b=b+1
5- Pow=P
6- برگشت

در زیر الگوریتم فوق ، a و b ورودی های الگوریتم هستند و مقدار a^b خروجی زیرالگوریتم است.

نکته) زیر الگوریتم ها نیازی به دستورات «شروع» و «پایان» ندارند. فقط در انتهای آنها باید دستور برگشت را نوشت.

مثال) می خواهیم زیر الگوریتمی بنویسیم که فاکتوریل عددی را محاسبه کند. این زیر الگوریتم شامل یک ورودی و یک خروجی است:

(فاکتوریل یک عدد از فرمول مقابل محاسبه می شود: $N!=1\times2\times3\times4\times..\times(N-1)\times N$)

زیر الگوریتم Fact(A)
1- I=1 , F=1
2- تا زمانی که I<=X ، دستورات 3 و 4 را انجام بده
3- F=F*I
4- I=I+1
5- Fact=F
6- برگشت

حال با کمک زیر الگوریتم بالا می خواهیم الگوریتمی بنویسیم که مقدار P(n,r) را از فرمول مقابل محاسبه کند:

$$P(n,r)=\frac{n!}{(n-r)!}$$

زیر الگوریتم Fact(A)
1- I=1 , F=1
2- تا زمانی که I<=X ، دستورات 3 و 4 را انجام بده
3- F=F*I
4- I=I+1
5- Fact=F
6- برگشت

الگوریتم
1- شروع
2- N و R را از ورودی بخوان
3- P=Fact(N)/Fact(N-R)
4- P را چاپ کن
5- پایان

تفاوت بین دو شکل نوشتاری حلقه ها

همانطور که در مقاله قبل آموختیم ، به دو شکل می توان یک حلقه را ایجاد کرد. یکی اینکه شرط حلقه را در ابتدای حلقه بنویسیم و دیگری اینکه شرط حلقه را در انتهای آن بنویسیم.
فرق این دوشکل نوشتاری اینست که در حالت دوم ، حتی اگر شرط حلقه درست نباشد ، اعمال حلقه یکبار انجام می شوند.

تمارین:

*) برای حل کردن بعضی از تمارین ، می توانید از الگوریتم های تمارین مقاله قبل استفاده کنید.

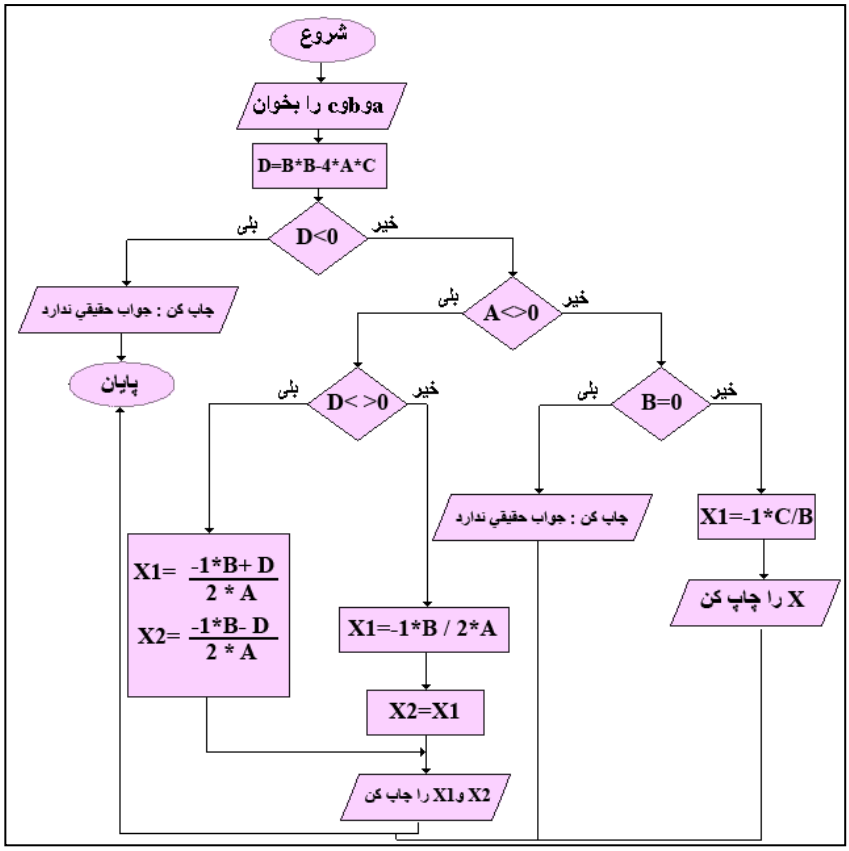
*1- فلوچارت الگوریتمی را رسم کنید که جوابهای حقیقی معادله $ax^2+b^x+c=0$ را محاسبه کند.
2- فلوچارت الگوریتمی را رسم کنید که 5 عدد از ورودی خوانده و اعداد زوج را دز خروجی چاپ می کند.
*3- فلوچارت الگوریتمی را رسم کنید که N عدد از ورودی خوانده و تعداد اعداد منفی ، صفر و مثبت را چاپ می کند.
*4- فلوچارت الگوریتمی را رسم کنید که تعداد ارقام عدد خوانده شده از ورودی را چاپ کند.
**5- فلوچارت الگوریتمی را رسم کنید که مقلوب یک عدد را چاپ کند(مقلوب به معنا است که مکان اعداد عدد را جابجا کنیم ، طوری که اولین رقم بجای آخرین رقم و دومین رقم بجای رقم یکی مانده به آخر و ... قرار گیرند. به عنوان مثال : مقلوب 153=351 ، مقلوب 1447=7441 ، مقلوب 9=9 .برای این سوال از سوال 1 کمک بگیرید.)
**6- فلوچارت الگوریتمی را رسم کنید که عددی از ورودی خوانده و اگر آن عدد ، عدد مقلوب بود چاپ کند بلی و در غیر اینصورت چاپ کند خیر.(برای این سوال از سوال 5 کمک بگیرید.)
**7- فلوچارت الگوریتمی را رسم کنید که حاصل جمع ارقام یک عدد را چاپ کند.(برای این سوال از الگوریتم سوال 7 استفاده کنید.)
**8- فلوچارت الگوریتمی را رسم کنید که معلوم کند یک عدد اول است یا نه(برای این کار آنرا به تمام اعداد کوچکتر از خودش به جز یک تقسیم کنید)
9- فلوچارت الگوریتم محاسبه فاکتوریل یک عدد را رسم کنید.
**10- فلوچارت الگوریتمی را رسم کنید که بدون استفاده از عمل ضرب ، حاصلضرب دو عدد خوانده شده از ورودی را چاپ کند.
***11- فلوچارت زیر چه عملی انجام می دهد ؟

*12- با کمک یک زیر الگوریتم الگوریتمی بنویسید که عدد صحیح و مثبت N را از ورودی خوانده و کلیه اعداد اول کوچکتر از N را چاپ کند.
**13- با کمک زیر الگوریتم ها الگوریتمی بنویسید که N عدد از ورودی خوانده و مشخص کند چند تا از این اعداد کامل هستند. (عدد کامل عددی است که مجموع مقسوم علیه هایش به جز خودش برابر با آن عدد باشد. مانند عدد 6 و عدد 28)

:هر ستاره نشان دهنده یک درجه سختی سوال می باشد.

حل تمرین ها:

-1



با توجه به اینکه الگوریتم های سوالات 2 تا 10 در مقاله قبلی در اختیار شما قرار گرفتند و همچنین به دلیل جاگیر و وقت گیر بودن رسم فلوچارت ، فلوچارت این تمرین ها در این مقاله رسم نمی شود.

به علت بزرگ بودن اندازه فلوچارت سوال 11 ، پاسخ این سوال بعد از پاسخ سوال های دیگر قرار دارد.

-12

زیر الگوریتم prime(I,P,K,W)

1- J=1 , W=0
2- تا زمانی که J<=K و W=0 مراحل 3 و 4 را انجام بده
3- اگر P(J)=0*(I/P(J))-I ، W=1
4- J=J+1
5- بازگشت

الگوریتم

1- N را بخوان
2- P(1)=2 , K=1 , I=3
3- تا زمانی که I<=N مراحل 4 تا 6 را انجام بده
4- زیر الگوریتم prime(I,P,K,W) را اجرا کن
5- اگر W=1 P(K)=I و K=K+1
6- I=I+1
7- تمامی K عنصر P را چاپ کن
8- پایان

در زیرالگوریتم این سوال ، I عددی است که اول بودن آن مورد بررسی قرار می گیرد. P آرایه ای از اعداد اول، K تعداد اعداد اول موجود در آرایه و W متغیر کمکی است که قبل از بررسی برابر صفر است و اگر I اول باشد ، W برابر 1 می شود.
در پاسخ این سوال ، عمل ایجاد آرایه انجام نشده است. چون تعداد اعداد اول کوچکتر از N مشخص نیست.
برای حل این مشکل می توان P را با N عنصر ایجاد کرد و بعد K عنصر اول آن را چاپ کرد.

-13

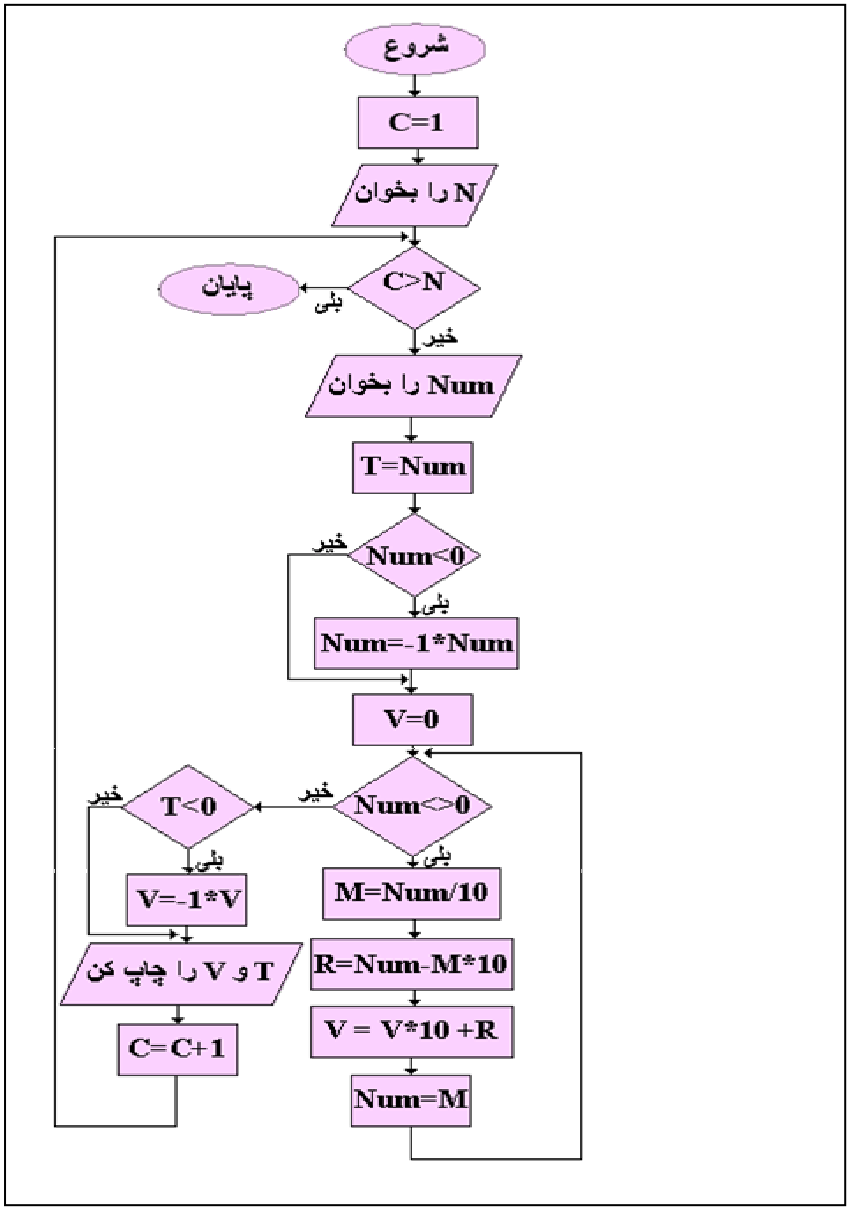
زیر الگوریتم Com(A)

1- ST=0 ، K=1
2- تا زمانی که K<A ، مراحل 3 تا 5 را انجام بده
3- R=A-K*(A/K)
4- اگر R=0 ، آنگاه ST=ST+K
5- K=K+1
6- Com=ST
7- بازگشت

الگوریتم

1- N را بخوان
2- I=1
3- تا زمانی که I <= N مراحل 4 تا 6 را انجام بده
4- A را بخوان
5- اگر Com(A) برابر A بود ، A را چاپ کن
6- پایان

-11



آنچه در مقاله بعد می خوانیم:

1- اثبات درستی الگوریتم
2- برنامه نویسی به زبان سی پلاس پلاس
3- آرایه ها در سی پلاس پلاس

تماس با ما:

شما می توانید از طریق پست الکترونیک و یا با مراجعه به وبلاگ های ما با ما تماس حاصل کنید.

پست الکترونیک : silver_star_senator@yahoo.com
وبلاگ برنامه نویسی به زبان سی پلاس پلاس : http://cppcenter.blogfa.com
وبلاگ برنامه نویسی به زبان سی پلاس پلاس : http://sstarsprograms.blogfa.com
وبلاگ برنامه نویسی به زبان سی پلاس پلاس : http://cppprograms.blogfa.com
وبلاگ دانلود بازی : http://sstarsgames.blogfa.com
وبلاگ بانک سورس و کد : http://sstarssourcebank.blogfa.com