

# LEARNING C++

<http://IstarSprograms.blogfa.com>

آموزش C++

سری مقالات الگوریتم نویسی: مقاله شماره 1

"Learning c++" Articles

By S.Rezaei



## مقدمه

### انواع زبان های برنامه نویسی

برنامه نویسان دستورات برنامه نویسی را به زبانهای مختلفی می نویسند. بعضی دستورات مستقیماً برای کامپیوتر قابل فهم می باشند و بعضی نیاز به ترجمه دارند. به طور کلی زبانهای برنامه نویسی به سه دسته تقسیم می شوند:

- 1-زبانهای ماشین
- 2-زبانهای اسمبلی
- 3- زبانهای سطح بالا

هر کامپیوتر زبان ماشین مربوط به خود را دارد که این زبان بستگی به سخت افزار آن کامپیوتر دارد. زبان ماشین به سختی برای انسان قابل فهم می باشد.

با عمومی تر شدن کامپیوتر، زبان ماشین به دلیل مشکل بودن فهم دستورات آن و دشوار بودن خطایابی جای خود را به زبان دیگری به نام زبان اسمبلی داد. این زبان به جای استفاده از اعداد از کلمات انگلیسی که قابل فهمتر برای انسان می باشند استفاده می کند. برنامه مترجمی که دستورات اسمبلی را به زبان قابل فهم توسط کامپیوتر (زبان ماشین) تبدیل می کند، اسمبلر می گویند.

### الگوریتم نویسی

برای شروع برنامه نویسی بهتر است از نوشتن الگوریتم برنامه ها شروع کرد که در این مقاله به آموزش نوشتن الگوریتم می پردازیم.

**الگوریتم** مجموعه ای از دستورالعمل ها است که با زبان دقیق و قابل فهم به همراه جزئیات لازم و بصورت مرحله به مرحله (باید ترتیب مراحل مشخص باشد) به گونه ای اجرا می شود که هدف خاصی که همان حل مساله است را دنبال کرده و شروع و پایانش نیز معلوم باشد. به زبان ساده تر الگوریتم مجموعه مراحل ایست که برای حل یک مساله می پیماییم.

فرض کنید شما می خواهید برای خریدن ماست به مغازه بروید. الگوریتمی که شما دنبال می کنید اینگونه است:

- 1- شروع
- 2- پول را از خانه بردار و از خانه خارج شو
- 3- به سمت مغازه حرکت کن
- 4- اگر به مغازه رسیدی پول را بده ، ماست را بگیر و به سمت خانه برگرد، در غیر اینصورت برو به مرحله 3
- 5- پایان

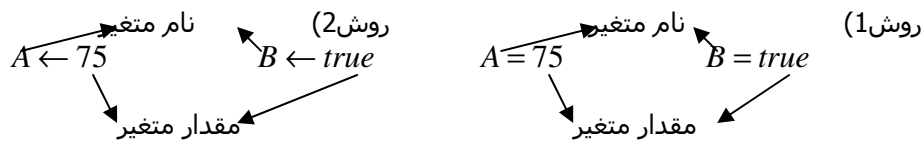
دقت کنید که در نوشتن الگوریتم رعایت **ترتیب مراحل** و **شرط شروع** و **خاتمه** ضروری است.

الگوریتم زیر برای جمع کردن دو عدد است.

- 1- شروع
- 2- دو عدد را از ورودی دریافت کن.
- 3- اعداد را با هم جمع کن.
- 4- حاصل اعداد چاپ کن.
- 5- پایان

این الگوریتم شامل 5 مرحله می باشد. مراحل 1 و 5 برای آماده شدن کامپیوتر برای اجرای برنامه و اتمام برنامه می باشند. مرحله دوم برای دریافت اعداد از ورودی است. در مرحله چهارم منظور از چاپ کردن درج حاصل جمع در خروجی است. همانطور که دیدید ، ما این الگوریتم را با کمک جملات فارسی بیان کردیم. اکنون می خواهیم آنرا به صورت ریاضی بیان کنیم. برای اینکار نیاز به دانستن مفهوم های بیشتری داریم:

**متغیر:** متغیر ها مکان هایی هستند که توانایی نگه داری و ذخیره سازی انواع داده را دارند. متغیر ها در الگوریتم همان کاربرد ریاضی خود را دارا هستند و علاوه بر اعداد می توانند متن یا مقادیر دیگری را در آنها نگهداری کرد. برای اینکه متغیری دارای مقدار شود ، باید به آن مقداردهی کنیم. در الگوریتم به اشکالی مانند اشکال زیر میتوان به متغیر ها مقدار دهی کرد. روش دوم متداول تر است ولی در این سری مقالات از روش اول استفاده شده است:



**عملگر:** عملگر یک یا مجموعه عملیاتی است که بصورت از پیش تعریف شده و برای استفاده از آن کافی است از علایمی که بصورت قراردادی تعریف شده است ، استفاده کنید. مانند عملگر + که حاصل جمع دو عدد یا دو متغیر را محاسبه و ارائه می کند. **نکته** در صورتی که به متغیر مقداردهی نکنیم ، کامپیوتر عددی که از قبل در آن فضا وجود داشته را به عنوان مقدار آن متغیر فرض می کند. **نکته** دقت کنید که نام متغیر در سمت چپ و مقدار آن در سمت راست نوشته می شود. **نکته** دقت کنید که متغیری که به نام a نام گذاری شده است با متغیری که به نام A نام گذاری شده است ، متفاوت است. **نکته** اگر بخواهیم به متغیری مقدار یک کاراکتر (مانند حروف) را مقداردهی کنیم ، به شیوه زیر عمل می کنیم:

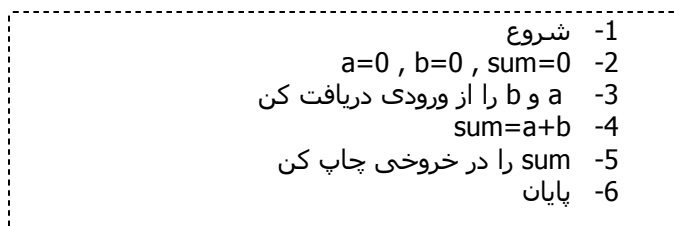
`Name="Reze" , S="A"`

**الگوریتم نویسی:**

الگوریتم ها اغلب شامل این مراحل می باشند:

- 1- خواندن ورودی ها
- 2- انجام محاسبات
- 3- چاپ نتیجه محاسبات

این الگوریتم را بدین ترتیب نیز می توان نوشت:



در مرحله دوم این الگوریتم کامپیوتر فضایی برای متغیر های a , b و sum در نظر میگیرد و در آنها مقدار صفر را قرار می دهد. در صورتی که به متغیر مقداردهی نکنیم ، کامپیوتر عددی که از قبل در آن فضا وجود داشته را به عنوان مقدار آن متغیر فرض می کند. از علامت = برای مقدار دهی به متغیر ها استفاده می شود. دقت کنید که نام متغیر در سمت چپ و مقدار آن در سمت راست نوشته می شود.

به این الگوریتم توجه کنید:

- 1- شروع
- 2-  $a=0$
- 3-  $a$  را از ورودی دریافت کن.
- 4-  $q=a/7$
- 5-  $r=a-(7*q)$
- 6- اگر  $r$  برابر 0 بود ،  $a$  را چاپ کن
- 7- پایان

همانطور که ملاحظه می کنید ، این برنامه عددی از ورودی دریافت می کند ، اگر آن عدد بر 7 بخش پذیر بود آنرا چاپ می کند.

**نکته** در مرحله پنج ، چون اعداد بصورت اعشاری تعریف نمی شوند،  $q \times 7 \neq a$  <=

**شرط:** همانطور که ملاحظه می کنید ، در مرحله ششم برنامه از «اگر» استفاده شده است. ساختار این شرط اینگونه است:  
اگر ... (شرط)... (عمل شرط)  
به عنوان مثال: اگر  $r=0$  ،  $a$  را چاپ کن.

**نکته** از علامت های + (جمع) ، - (تفریق) ، \* (ضرب) ، / (تقسیم) ، < > ( بزرگتر و کوچکتر) ، <= >= (کوچکتر مساوی) ، > < (بزرگتر مساوی) ، <> (نامساوی) و... در برنامه نویسی و الگوریتم نویسی استفاده می شود.

**نکته** از Mod برای محاسبه باقیمانده استفاده می شود.(در زبان سی پلاس پلاس از علامت % برای همین عمل استفاده می شود.) به عنوان مثال:

$$10 \% 5 = 10 \text{ Mod } 5 = 0 , 7 \% 5 = 7 \text{ Mod } 5 = 2 , 3 \% 2 = 3 \text{ Mod } 2 = 1$$

بنابراین الگوریتم فوق را بدین صورت نیز می توان نوشت:

- 1-شروع
- 2-  $a=0$
- 3-  $a$  را از ورودی دریافت کن
- 4-  $r=a \text{ Mod } 7$
- 5- اگر  $r$  برابر 0 بود ،  $a$  را چاپ کن
- 6-پایان

**مثال** الگوریتمی بنویسید که سه عدد از ورودی دریافت کند و بگوید که آیا این سه عدد می توانند اضلاع یک مثلث باشند یا نه.  
جواب:

- 1-شروع
- 2-  $a$  و  $b$  و  $c$  را از ورودی دریافت کن
- 3- اگر  $b+c < a$  یا  $a+c < b$  یا  $b+a < c$  برو به 5
- 4- چاپ کن "بله" و برو به 6
- 5- چاپ کن "خیر"
- 6-پایان

در این الگوریتم از «برو به ...» استفاده شده است. این در الگوریتم بدین معناست که پس از انجام مرحله ، به مرحله مشخص شده می رود. و همچنین چنانکه دیدید در زمان شرط گذاشتن در الگوریتم می توانیم چند شرط را با هم اعمال کنیم. کلمه « یا » در بین شرط ها بدین معناست که اگر حداقل یکی از آنها درست باشد ، شرط درست است . همچنین کلمه « و » نیز به این معناست که شرطها باید همزمان درست باشند.

به الگوریتم زیر توجه کنید:

```

1- شروع
2- i=1
3- min را از ورودی دریافت کن.
4- a را دریافت کن
5- اگر min < a ، min=a
6- i=i+1
7- اگر i <= 10 برو به 4
8- min را چاپ کن
9- پایان
    
```

این الگوریتم 10 عدد از ورودی خوانده و کوچکترین آنها را چاپ می کند. در این الگوریتم از یک «حلقه» استفاده شده است. این بدین معناست که مادامیکه 10 عدد از ورودی خوانده نشده اند ، قسمتی از الگوریتم تکرار می شود. الگوریتم بالا را به شکل زیر نیز می توان نوشت:

```

1- شروع
2- i=1
3- min را از ورودی دریافت کن.
4- تا زمانی که i <= 10 ، مراحل 4 تا 7 را انجام بده
5- a را دریافت کن
6- اگر min < a ، min=a
7- i=i+1
8- min را چاپ کن
9- پایان
    
```

همانطور که می بینید ، حلقه ها را به این دو صورت می توان نوشت:

- 1- n- تا زمانی که ..... (شرط حلقه) مراحل ... تا ... را انجام بده
- 2- اگر ..... (شرط حلقه) به مرحله n برو

حلقه ها در زمانی که نیاز به خواندن تعداد زیادی عدد یا انجام یک عمل به تعداد زیاد و متوالی داریم ، فوق العاده مهم هستند. مثلا فرض کنید می خواهیم ده عدد از ورودی خوانده و با هم جمع کنیم. برای اینکار بدین شکل عمل می کنیم:

```

1- شروع
2- i=0 , a=0 , sum=0
3- تا زمانی که i < 10 ، مراحل 4 تا 7 را انجام بده
4- a را از ورودی بخوان
5- sum=sum+a
6- i=i+1
7- a را چاپ کن
8- پایان
    
```

**مبناها (سیستم های اعداد):**

اطلاعاتی که وارد کامپیوتر می شوند ، ترکیبی از حروف A تا Z ، a تا z ، ارقام 0 تا 9 و علامت هایی مانند . ; : @ # \$ % ^ & \* و ... است. هر گروه از این اطلاعات به شیوه مخصوص به خود در حافظه نگهداری می شوند. یعنی شیوه نگهداری اعداد با شیوه نگهداری اسم یا یک کاراکتر در کامپیوتر متفاوت است. پس این اطلاعات به دو دسته تقسیم می شوند:

- 1- اطلاعات عددی
- 2- اطلاعات رشته ای

در اینجا به مطالعه طریقه نگهداری اطلاعات عددی می پردازیم:

در سیستم های عددی معمولی هر عدد به شکل زیر نمایش داده است:

$$N = (a_{n-1}a_{n-2}a_{n-3}...a_1a_0a_{-1}...a_m) B$$

$$N = a_{n-1} B^{n-1} + a_{n-2} B^{n-2} + a_{n-3} B^{n-3} + ... + a_0 B^0 + a_{-1} B^{-1} + ... + a_m B^m$$

$$N = \sum_{k=-m}^{n-1} a_k B^k$$

هر عدد در مبنای B بصورت  $(N)_B$  نمایش داده می شود. اگر مبنا مشخص نشده باشد ، ده فرض می شود.

**سیستم اعداد دهدهی (مبنای 10):**

این سیستم اعداد یکی از سیستم های متداول است که همه در زندگی روزمره با آن سر و کار دارند. هر عدد در سیستم دهدهی ترکیبی از ارقام 0 تا 9 می باشد.

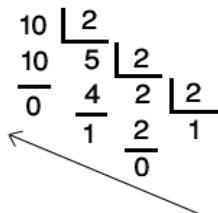
**سیستم اعداد دودویی (مبنای 2):**

در این سیستم مبنای اعداد 2 است پس اعداد می توانند ترکیبی از ارقام 0 و 1 باشند. به عنوان مثال اعداد یک رقمی را در مبنای 2 در جدول زیر ملاحظه می کنید:

دهدهی	0	1	2	3	4	5	6	7	8	9
دودویی	0	1	10	11	100	101	110	111	1000	1001

**تبدیل اعداد از مبنای 10 به مبنای 2:**

برای تبدیل اعداد از مبنای 10 به مبنای دو ، می توانید از روش تقسیم استفاده کنید. به عنوان مثال فرض کنید می خواهیم عدد 10 را به مبنای 2 تبدیل کنیم:



همانطور که مشاهده می کنید ، عدد 10 در مبنای دو برابر است با:

1010

سعی کنید الگوریتمی بنویسید که به همین ترتیب اعداد در مبنای ده را به مبنای 2 انتقال دهد.

**سیستم اعداد هشتایی (اکتال):**

در این سیستم مبنای اعداد 8 است. لذا هر عدد در این سیستم می تواند ترکیبی از ارقام 0 تا 8 باشد.

**سیستم اعداد شانزده تایی (هگزادسیمال):**

در این سیستم مبنای اعداد 16 است ، لذا می توان از 16 رقم در نوشتن این اعداد استفاده کرد. چون اعداد بزرگتر از 9 یک رقم نیستند ، از علامتهایی برای نمایش آنها در این مبنا استفاده می شود. این علامتها از این قرارند:

$$10=A, 11=B, 12=C, 13=F, 14=E, 15=F$$

در این قسمت از مقاله ، یک سری از الگوریتم ها برای مطالعه و درک بهتر قرار می گیرند که به ترتیب دشواری حل آنها ، از ساده به سخت مرتب شده اند. به دلیل جاگیر بودن الگوریتم ها ، تعدادی از آنها در مقالات و بقیه در وبلاگ بانک سورس قرار می گیرند.

### لیست الگوریتم های مقاله

الگوریتم های ساده

1- محاسبه حجم مخزنی به شکل مکعب مربع

- 1- شروع
- 2- a را از ورودی دریافت کن
- 3-  $a*a*a$  را چاپ کن
- 4- پایان

2- محاسبه معدل دانش آموزی در نمرات ریاضی ، فیزیک ، زبان و هندسه با توجه به جدول ضرایب

ضریب	درس
4	ریاضی
3	فیزیک
1	زبان
2	هندسه

- 1- شروع
- 2- r و h و z و p را از ورودی دریافت کن
- 3-  $((r*4)+(p*3+(h*2)+z))/10$  را چاپ کن
- 4- پایان

### الگوریتم های متوسط

1- خرد کردن پول به سکه های 50 و 20 و 10 و 5 و 2 و 1 ریالی

- 1- شروع
- 2-  $d=0, a=0$
- 3- a را از ورودی دریافت کن.
- 4-  $d=a/50$
- 5- اگر  $d>0$  ، آنگاه d و 50 را چاپ کن
- 6-  $a=a-(d*50)$
- 7-  $d=a/20$
- 8- اگر  $d>0$  ، آنگاه d و 20 را چاپ کن
- 9-  $a=a-(d*20)$
- 10-  $d=a/10$
- 11- اگر  $d>0$  ، آنگاه d و 10 را چاپ کن
- 12-  $a=a-(d*10)$
- 13-  $d=a/5$
- 14- اگر  $d>0$  ، آنگاه d و 5 را چاپ کن
- 15-  $a=a-(d*5)$
- 16-  $d=a/2$
- 17- اگر  $d>0$  ، آنگاه d و 2 را چاپ کن
- 18-  $a=a-(d*2)$
- 19- اگر  $a>0$  ، آنگاه a و 1 را چاپ کن
- 20- پایان

**2- محاسبه ب.م.م و ک.م.م**

- 1- شروع
- 2-  $a=0, b=0, \max=0, \text{BMM}=1, i=1, \text{KMM}=0$
- 3-  $a$  و  $b$  را از ورودی دریافت کن
- 4- اگر  $a > b$  ،  $\max=a$  ، در غیر اینصورت  $\max=b$
- 5- اگر  $0 = a - (a/i) * i$  و  $\text{BMM} = i - (b/i) * i = 0$
- 6-  $i = i + 1$
- 7- اگر  $i \leq \max$  ، برو به 5
- 8-  $\text{KMM} = (a * b) / \text{BMM}$
- 9-  $\text{BMM}$  و  $\text{KMM}$  را در خروجی نمایش بده
- 10- پایان

الگوریتم های سخت

**1- فاکتوریل یک عدد**

- 1- شروع
- 2-  $a=0, \text{fact}=1$
- 3-  $a$  را از ورودی دریافت کن
- 4- تا زمانی که  $0 < a$  ، مرحله 5 را انجام بده
- 5-  $\text{fact} = a * \text{fact}$
- 6-  $a = a - 1$
- 7-  $\text{fact}$  را چاپ کن
- 8- پایان

**2- سینوس یک زاویه بر حسب رادیان**

همانطور که میدانید سینوس زاویه بر حسب رادیان از فرمول مقابل محاسبه می شود:

$$\sin X = X - (X^3 \div 3!) + (X^5 \div 5!) - (X^7 \div 7) + \dots$$

فرض کنید تعداد جملات  $2n+1$  باشد.

- 1- شروع
- 2-  $x$  و  $n$  را از ورودی دریافت کن
- 3-  $i=1, k=1, \sin=0$
- 4-  $\text{fact}=1, z=1$
- 5-  $\text{fact} = \text{fact} * z$
- 6-  $z = z + 1$
- 7- اگر  $z \leq I$  برو به 5
- 8-  $\sin = \sin + k * (x^i / \text{fact})$
- 9-  $i = i + 2$
- 10-  $k = k * (-1)$
- 11- اگر  $i \leq 2n+1$  برو به 4
- 12-  $\sin$  را چاپ کن
- 13- پایان

( به جای  $x^i$  می توان از یک حلقه استفاده کرد. ولی برای طولانی نشدن الگوریتم از این شکل نمایش استفاده شده است.)



لیست الگوریتم های وبلاگ (مربوط به این مقاله) <http://sstarsourcebank.blogfa.com>

- |                          |                                                       |
|--------------------------|-------------------------------------------------------|
| <a href="#">Download</a> | 1- تشخیص زوج یا فرد بودن عدد                          |
| <a href="#">Download</a> | 2- تبدیل مبنای 2 به مبنای 10                          |
| <a href="#">Download</a> | 3- تبدیل مبنای 16 به مبنای 10                         |
| <a href="#">Download</a> | 4- تبدیل مبنای 10 به مبنای 2                          |
| <a href="#">Download</a> | 5- تبدیل مبنای 10 به مبنای 16                         |
| <a href="#">Download</a> | 6- محاسبه حجم کره                                     |
| <a href="#">Download</a> | 7- محاسبه حاصل تقسیم دو عدد بدون استفاده از عمل تقسیم |
| <a href="#">Download</a> | 8- بررسی بخشپذیری دو عدد بر یکدیگر                    |
| <a href="#">Download</a> | 9- تشخیص مثبت ، منفی یا صفر بودن عدد                  |
| <a href="#">Download</a> | 10- محاسبه عدد e (عدد نپر)                            |
| <a href="#">Download</a> | 11- محاسبه جملات دنباله فیبوناتچی                     |
| <a href="#">Download</a> | 12- پیدا کردن بزرگترین عدد در بین 10 عدد              |

تمارین:

- 1\* - الگوریتمی بنویسید که تعداد ارقام عدد خوانده شده از ورودی را چاپ کند.
- 2\*\* - الگوریتمی بنویسید 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.)
- 3\*\* - الگوریتمی بنویسید 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 2 کمک بگیرید. )
- 4\*\* - الگوریتمی بنویسید که حاصل جمع ارقام یک عدد را چاپ کند. (برای این سوال از الگوریتم سوال 1 استفاده کنید.)
- 5\* - الگوریتمی بنویسید که عددی از ورودی خوانده و مجذور ، مکعب و نصف آنرا در خروجی چاپ کند.
- 6 - الگوریتمی بنویسید که طول و عرض مستطیلی را از ورودی دریافت کند و مساحت و محیط آنرا در خروجی چاپ کند.
- 7 - الگوریتمی بنویسید که شعاع دایره ای را از ورودی بخواند و مساحت و محیط آنرا در خروجی چاپ کند.
- 8\*\* - این الگوریتم چه عملی انجام می دهد؟

- 1- شروع  
2- a و b و c و d را از ورودی دریافت کن  
3-  $ave = (a+b+c+d)/4$   
4- ave را چاپ کن  
5- پایان

- 9\*\* - الگوریتمی بنویسید که معلوم کند یک عدد اول است یا نه (برای این کار آنرا به تمام اعداد کوچکتر از خودش به جز یک تقسیم کنید)
- 10\*\* - پس از انجام الگوریتم زیر خروجی برابر 55 بوده است. ورودی را مشخص کنید.

- 1- شروع  
2- a را از ورودی دریافت کن  
3-  $i=0$   
4- اگر  $a>0$  ،  $i=i+1$  و  $a=a-1$   
5- اگر  $a>0$  برو به 4  
6- i را چاپ کن  
7- پایان

- 11\*\* - الگوریتمی بنویسید که بدون استفاده از عمل ضرب ، حاصلضرب دو عدد خوانده شده از ورودی را چاپ کند

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

**حل تمرین ها:**

```

-1- شروع
-2- a=0 , q=1 , c=0
-3- a را از ورودی دریافت کن
-4- q=a/10
-5- c++
-6- a=q
-7- اگر q برابر 0 نبود ، برو به 4
-8- c را چاپ کن
-9- پایان
    
```

```

-2- شروع
-1- شروع
-2- a=0 , b=0 , s=0 , r=0
-3- a را از ورودی دریافت کن
-4- s=a/10
-5- r=a%10
-6- b=r+(b*10)
-7- a=s
-8- اگر a/10 برابر 0 نبود برو به 4
-9- b را چاپ کن
-10- پایان
    
```

3- برای حل این سوال کافیسست در الگوریتم سوال قبل اگر  $a=b$  بود چاپ شود بلی ، در غیر اینصورت چاپ شود خیر.

4- برای حل این سوال کافیسست در الگوریتم سوال اول به جای مرحله 5 ، نوشته شود:  $c=c+(a\%10)$

5 و 6 و 7 - حل این سه سوال بر عهده خود شماست!

8- این الگوریتم چهار عدد از ورودی خوانده و میانگین آنها را چاپ می کند.

```

-9- شروع
-1- شروع
-2- a=0 , i=2
-3- تا زمانی که  $i < a$  مرحله 4 و 5 را انجام بده
-4- اگر  $a\%i=0$  چاپ کن خیر و برو به
-5-  $i=i+1$ 
-6- چاپ کن بلی و برو به 8
-7- چاپ کن خیر
-8- پایان
    
```

10- در هر مرحله از حلقه الگوریتم از عدد خوانده شده یکی کم می شود و به  $i$  یک واحد اضافه میشود. پس این الگوریتم خود ورودی را چاپ می کند. پس ورودی 55 بوده است.

-11

- 1- شروع
- 2- m و n را از ورودی بخوان
- 3- s=0
- 4- s=s+n
- 5- m=m-1
- 6- اگر m بزرگتر تر از صفر بود ، برو به 4
- 7- s را چاپ کن
- 8- پایان

**آنچه در مقاله بعد می خوانیم:**

- 1- طریقه رسم فلوجارت
- 2- آرایه ها و طرز کار با آنها
- 3- مفهوم زیرالگوریتم ها و طرزکار با آنها

**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](mailto:silver_star_senator@yahoo.com)  
 وبلاگ برنامه نویسی : <http://sstarsprograms.blogfa.com>  
 وبلاگ دانلود بازی : <http://sstarsgames.blogfa.com>  
 وبلاگ بانک سورس و کد : <http://sstarsourcebank.blogfa.com>  
 و این هم وبلاگ جدید: <http://cppcenter.blogfa.com>