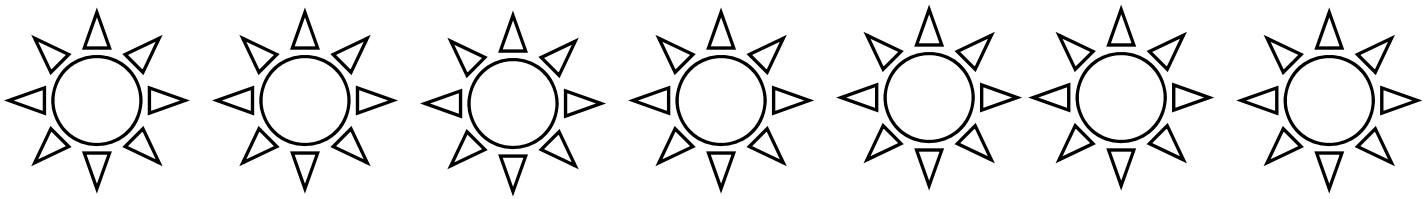
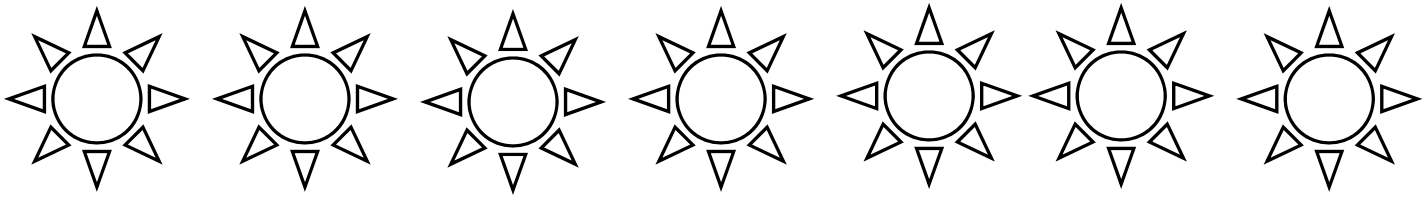


برنامه نویسی کامپیوتر



الگوریتم

الگوریتم: به مجموعه ای از دستورالعملها که مراحل حل یک مساله را با زبان دقیق و جزئیات کافی بیان کرده، دارای ترتیب مراحل و پایان پذیری مشخصی باشد، الگوریتم می گویند.

زبان دقیق: بدون ابهام.

جزئیات کافی: تمام دستورات قابل تفسیر باشند.

ترتیب مراحل: ترتیب دستورات از لحاظ اجرا مهم است.

پایان پذیری: در زمان معین الگوریتم خاتمه یابد.

ویژگیهای الگوریتم

- متناهی بودن: تعداد مراحل متناهی (محدود) باشند.
- روشن و فاقد ابهام بودن: مراحل اجرا دقیقا روشن باشند.
- مشخص بودن ورودیها: الگوریتم چند ورودی دارد.
- مشخص بودن خروجیها: الگوریتم چند خروجی دارد. چه رابطه ای با ورودیها دارد.
- موثر بودن: تاثیرگذاری دستورات در برنامه.

دستورالعملهای الگوریتم

- شروع
- پایان
- ورودی
- خروجی
- محاسباتی و جایگزینی
- شرطی
- حلقه

الگوریتم 1

الگوریتمی بنویسید که سه نمره یک دانشجو را دریافت کرده ، معدل آن را محاسبه کرده و نتیجه را نمایش دهد.

1- شروع

2- a ، b و c را از ورودی دریافت کن

3- avg←(a+b+c)/3

4- مقدار avg را نمایش بده

5-پایان

الگوریتم 2

الگوریتمی بنویسید که شعاع یک کره را دریافت کرده ، مساحت و حجم آن را محاسبه کرده و نمایش دهد.

1- شروع

2- r را از ورودی دریافت کن

3- s← 4*3.14*r*r

4- v← 4/3*3.14*r*r*r

5- مقدار S و V را نمایش بده

6- پایان

الگوریتم 3

الگوریتمی بنویسید که چهار عدد a ، b ، c ، و d را دریافت کرده ، حاصل عبارت $S=a^3+b^3+c^3+d^3$ را محاسبه کرده و نمایش دهد.

1- شروع

2- a ، b ، c ، وd را از ورودی دریافت کن

3- $S \leftarrow a^3+b^3+c^3+d^3$

4- مقدار s را نمایش بده

5- پایان

جدول ردیابی

برای ردیابی الگوریتم باید از جدول ردیابی استفاده کرد.

در این جدول برای هر متغیر یک ستون در نظر گرفته می شود. ردیابی از ابتدای الگوریتم شروع شده و با در نظر گرفتن مقادیر اولیه متغیرها تا مرحله پایان ادامه داده می شود. مقادیر جدید متغیرها در زمان اجرای الگوریتم در ستون مربوطه نوشته می شود. سپس خروجی نهایی بررسی می شود که درست است یا خیر.

مثال

جدول ردیابی الگوریتم زیر را برای سه عدد 5، 8، 13 ترسیم کرده و خروجی را اعلام کنید.

1- شروع

2- a ، b و c را از ورودی دریافت کن

3- avg←(a+b+c)/3

4- مقدار avg را نمایش بده

5-پایان

a	b	c	avg
5	8	13	13

الگوریتم 4

الگوریتمی بنویسید که حقوق ناخالص یک کارمند را دریافت کرده ، 3% بیمه ، 4% حق مسکن از آن کم کرده ، حقوق خالص را به دست نمایش دهد.

1- شروع

2- w را از ورودی دریافت کن

3- b← 3*w/100

4- m← 4*w/100

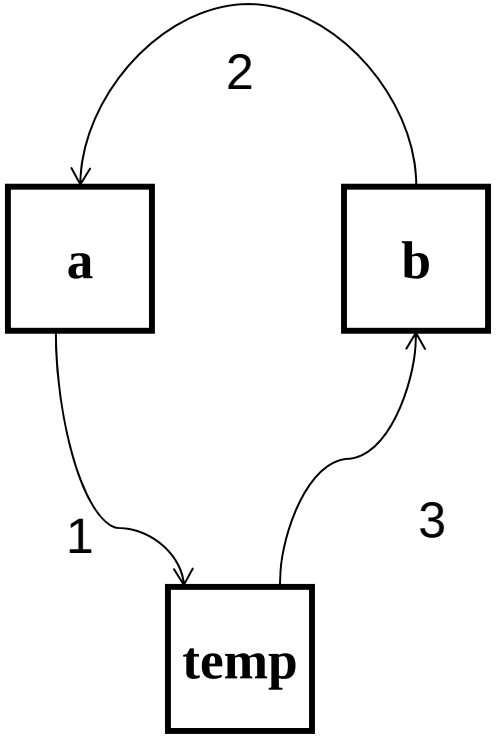
5- net← w-(b+m)

6- مقدار net را نمایش بده

7-پایان

الگوریتم 5

الگوریتمی بنویسید که دو عدد را دریافت کرده ، محتوای آن دو عدد را جابجا کرده و نتیجه را نمایش دهد.



1- شروع

2- a ، b را از ورودی دریافت کن

3- temp ← a

4- a← b

5- b ← temp

5- مقدار a و b را نمایش بده

6- پایان

ساختار کنترل (شرط)

هرگاه در طول الگوریتم نیاز به استفاده از شرط یا شروط داشته باشیم ، از اگر استفاده مي‌كنيم.

1- اگر شرط آنگاه دستورات

2- اگر شرط آنگاه دستورات1 درغیراینصورت دستورات2

الگوریتم 6

الگوریتمی بنویسید که یک عدد دریافت کرده ، نشان دهد ، مثبت ، منفی یا صفر است.

1- شروع

2- a را از ورودی دریافت کن

3- اگر a<0 آنگاه نمایش بده ‘a is negative’

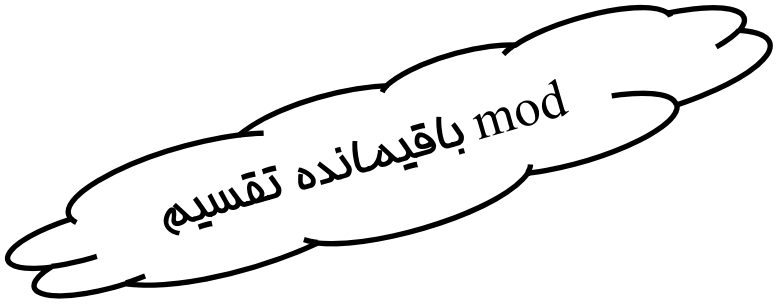
4- اگر a>0 آنگاه نمایش بده ‘a is positive’

5- اگر a=0 آنگاه نمایش بده ‘a is zero’

6- پایان

الگوریتم 7

الگوریتمی بنویسید که یک عدد دریافت کرده ، نشان دهد ، زوج است یا فرد.



1- شروع

2- a را از ورودی دریافت کن

3- اگر a mod 2 =0 آنگاه نمایش بده ‘Even’

در غیر اینصورت نمایش بده ‘Odd’

4- پایان

الگوریتم 8

الگوریتمی بنویسید که دو عدد دریافت کرده ، عدد بزرگتر را نمایش دهد.

1- شروع

2- a و b را از ورودی دریافت کن

3- max ← a

4- اگر b>max آنگاه max ← b

5- مقدار max را نمایش بده

6- پایان

الگوریتم 9

الگوریتمی بنویسید که سه عدد را دریافت کرده ، اگر a+c>b باشد مقدار a و در غیراینصورت مقدار b را نمایش دهد.

1- شروع

2- a و b و c را از ورودی دریافت کن

3- اگر a+c>b آنگاه مقدار a را نمایش بده

درغیراینصورت مقدار b را نمایش بده

4- پایان

الگوریتم 10

الگوریتمی بنویسید که ضرایب یک معادله دو مجهولی را دریافت کرده ، ریشه‌های آن را درصورت وجود نمایش دهد.

1- شروع

2- a و b و c را از ورودی دریافت کن

3- d ← b*b-4*a*c

4- اگر d>=0 آنگاه

x1 ←(-b-sqrt(d))/(2*a)

x2 ←(-b+sqrt(d))/(2*a)

x1 و x2 را نمایش بده

درغیر اینصورت نمایش بده ‘no root’

5- پایان

الگوریتم 11

الگوریتمی بنویسید که سه عدد را دریافت کرده ، نشان دهد ، آیا این سه عدد می‌توانند اضلاع یک مثلث باشند.

1- شروع

2- a و b و c را از ورودی دریافت کن

3- اگر $(a+b)>c$ و $(a+c)>b$ و $(b+c)>a$ آنگاه

نمایش بده ‘Yes’

درغیراینصورت نمایش بده ‘No’

4- پایان

الگوریتم 12

الگوریتمی بنویسید که دو عدد به همراه یک عملگر را از ورودي دریافت کرده ، عملیات را روی اعداد انجام داده ، نتیجه را نمایش دهد.

1- شروع
2- a و b و c را از ورودی دریافت کن
3- اگر c=‘*’ آنگاه s← a * b
4- اگر c=‘-’ آنگاه s← a - b
5- اگر c=‘/’ آنگاه s← a / b
6- اگر c=‘+’ آنگاه s← a + b
7- مقدار S را نمایش بده
4- پایان

حلقه

❖ هنگامیکه یک یا چند دستورالعمل باید چندین بار اجرا شوند ، از حلقه ها استفاده می شود.

❖ حلقه به دو صورت وجود دارد:

1- شمارشی (تعداد مراحل اجرا مشخص است)

2- غیرشمارشی (بر اساس یک شرط می باشد)

الگوریتم 13

الگوریتمی بنویسید که مجموع اعداد 1 تا 100 را محاسبه کرده و نمایش دهد.

1- شروع

2- $I \leftarrow 1$ ، $s \leftarrow 0$

3- $s \leftarrow s + I$

4- $I \leftarrow I + 1$

5- اگر $I<=100$ آنگاه برو به مرحله 3

6- مقدار S را نمایش بده

7- پایان

الگوریتم 14

الگوریتمی بنویسید که عدد صحیحي را از ورودي دریافت کرده و فاکتوریل آن را محاسبه کرده و نمایش دهد.

1- شروع

2- n را دریافت کن

3- I← 1 ، f ← 1

4- f ← f * I

5- I ← I + 1

6- اگر I<=n آنگاه برو به مرحله 4

7- مقدار f را نمایش بده

8- پایان

الگوریتم 15

الگوریتمی بنویسید که دو عدد صحیح مثبت را از ورودي دریافت کرده و حاصلضرب آن دو را بدون استفاده از عملگر ضرب محاسبه کرده و نمایش دهد.

1- شروع

2- a و b را دریافت کن

3- $I \leftarrow 1$ ، $s \leftarrow 0$

4- $s \leftarrow s + a$

5- $I \leftarrow I + 1$

6- اگر $I <= b$ آنگاه برو به مرحله 4

7- مقدار S را نمایش بده

8- پایان

الگوریتم 16

الگوریتمی بنویسید که عدد صحیح مثبت n را از ورودي دریافت کرده مجموع سری زیر را محاسبه کرده و نمایش دهد.

$$S = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}$$

1- شروع

2- n را دریافت کن

3- $s \leftarrow 0$

4- $I \leftarrow 1$

5- $s \leftarrow s + 1/I$

6- $I \leftarrow I + 1$

7- اگر $I <= n$ آنگاه برو به مرحله 5

8- مقدار s را نمایش بده

9- پایان

الگوریتم 17

الگوریتمی بنویسید که مجموع مضارب 5 کوچکتر از 100 را نمایش دهد.

1- شروع
2- $s \leftarrow 0$
3- $I \leftarrow 0$
4- $s \leftarrow s + I$
5- $I \leftarrow I + 5$
6- اگر $I<=100$ آنگاه برو به مرحله 4
7- مقدار S را نمایش بده
8- پایان

الگوریتم 18

الگوریتمی بنویسید که یک عدد را دریافت کرده ، مقلوب آن را نمایش دهد. مقلوب عدد 123 ، عدد 321 است.

1- شروع

2- n را دریافت کن

3- $m \leftarrow 0$

4- $b \leftarrow n \bmod 10$

5- $m \leftarrow m * 10 + b$

6- $n \leftarrow n \operatorname{div} 10$

7- اگر $n>0$ آنگاه برو به مرحله 4

8- مقدار m را نمایش بده

9- پایان

الگوریتم 19

الگوریتمی بنویسید که دو عدد را دریافت کرده ، ب.م.م آن دو را نمایش دهد.

1- شروع
2- n و m را دریافت کن
3- $r \leftarrow m \bmod n$
4- $m \leftarrow n$
5- $n \leftarrow r$
6- اگر $(m \bmod n) <> 0$ آنگاه برو به مرحله 3
7- مقدار n را نمایش بده
8- پایان

الگوریتم 20

الگوریتمی بنویسید که عدد صحیح مثبت n را از ورودي دریافت کرده مجموع سری زیر را محاسبه کرده و نمایش دهد.

$$S = \frac{1}{2} + \frac{1}{4} + ... + \frac{1}{2n}$$

1- شروع
2- n را دریافت کن
3- s ← 0
4- I ← 2
5- s← s + 1/I
6- I ← I + 2
7- اگر I<=2*n آنگاه برو به مرحله 5
8- مقدار s را نمایش بده
9- پایان

الگوریتم 21

الگوریتمی بنویسید که عدد صحیح مثبت n را از ورودي دریافت کرده مجموع سری زیر را محاسبه کرده و نمایش دهد.

$$S = 1 - \frac{1}{3} + \frac{1}{5} - .... + \frac{1}{n}$$

1- شروع

2- n را دریافت کن

3- s ← 0

4- k ← 1 ، I ← 1

5- s← s + k/I

6- I ← I + 2

7- k=k * (-1)

8- اگر I<=n آنگاه برو به مرحله 5

9- مقدار s را نمایش بده

10- پایان

الگوریتم 22

الگوریتمی بنویسید که دو عدد دریافت کرده ، خارج قسمت و باقیمانده صحیح تقسیم را بدون استفاده از عمل تقسیم نمایش دهید.

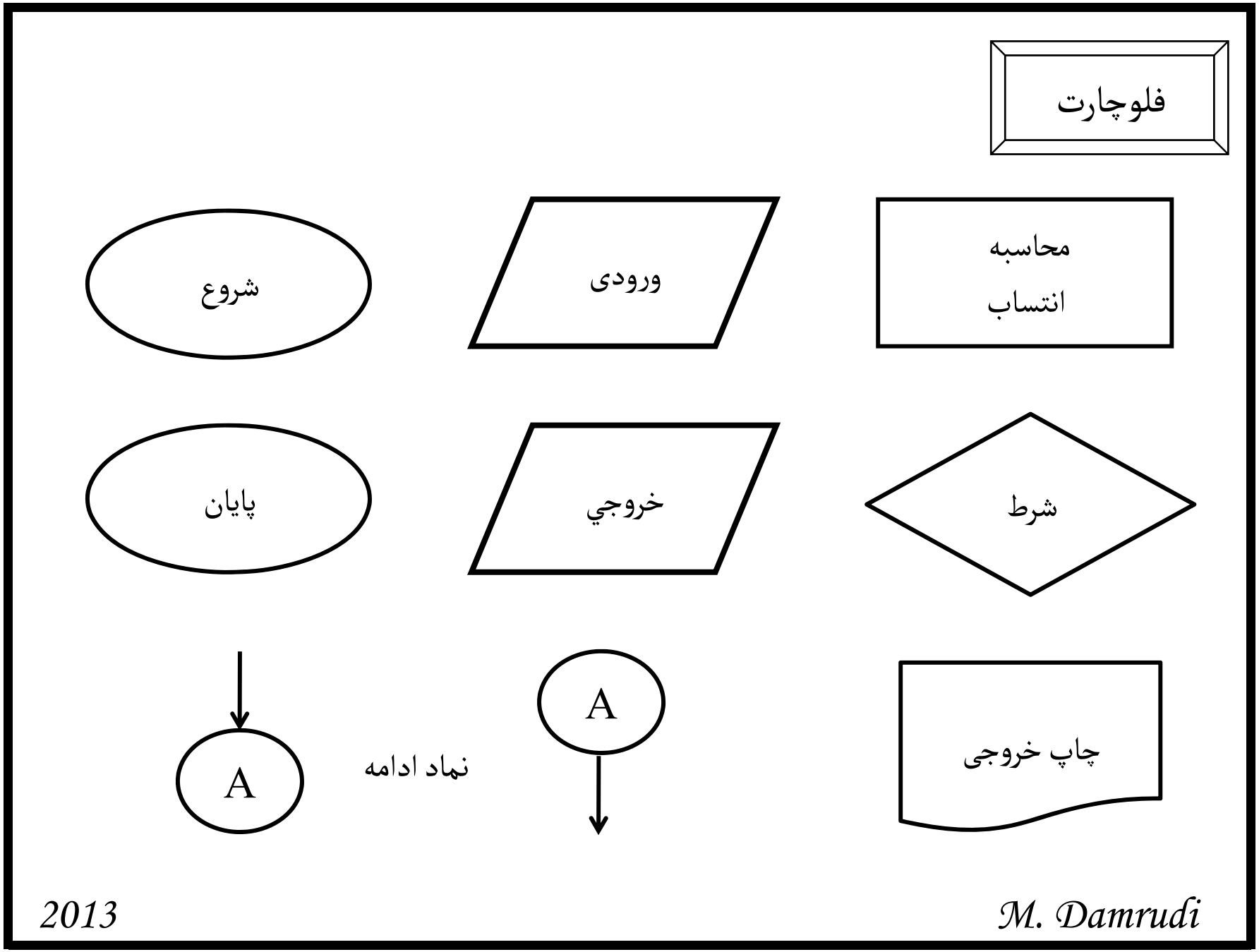
1- شروع
2- b و a را دریافت کن
3- $q \leftarrow 0$
4- $a \leftarrow a - b$
5- $q \leftarrow q + 1$
6- اگر $a>b$ آنگاه برو به مرحله 4
7- a و q را نمایش بده
8- پایان

فلوچارت

فلوچارت: فلوچارت ، ساده ترین و واضح ترین روش تصویری بیان الگوریتم است.

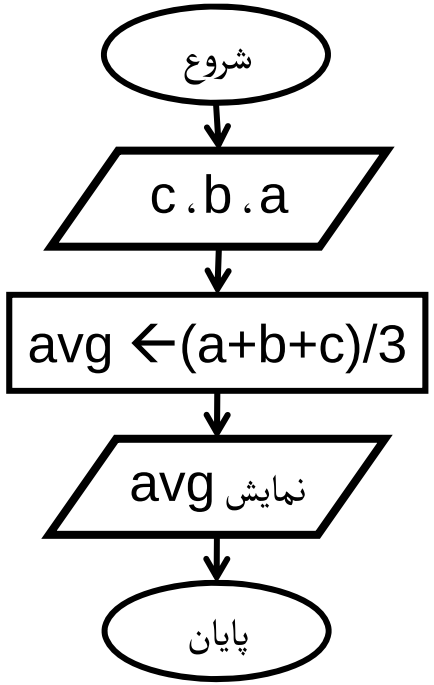
از نمادهای خاصی(اشکال هندسی) برای نمایش مراحل اجرای الگوریتم استفاده می شود.

هر عملی با یک نماد نشان داده می شود.


فلوچارت
محاسبه
انتساب
ورودی
شروع
شرط
خروجي
پایان
چاپ خروجی
A
نماد ادامه
A
2013
M. Damrudi

فلوچارت 1

فلوچارتی بنویسید که سه نمره یک دانشجو را دریافت کرده ، معدل آن را محاسبه کرده و نتیجه را نمایش دهد.



1- شروع

2- a ، b و c را از ورودی دریافت کن

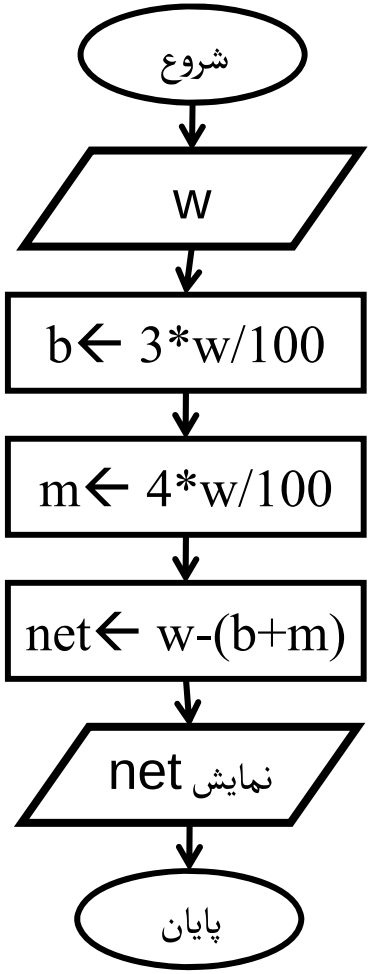
3- avg←(a+b+c)/3

4- مقدار avg را نمایش بده

5-پایان

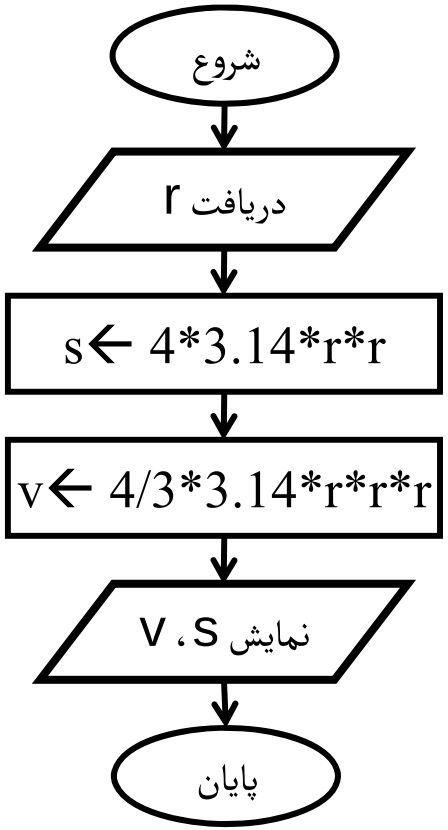
فلوچارت 2

فلوچارتی بنویسید که حقوق ناخالص یک کارمند را دریافت کرده ، 3% بیمه ، 4% حق مسکن از آن کم کرده ، حقوق خالص را به دست آورید.



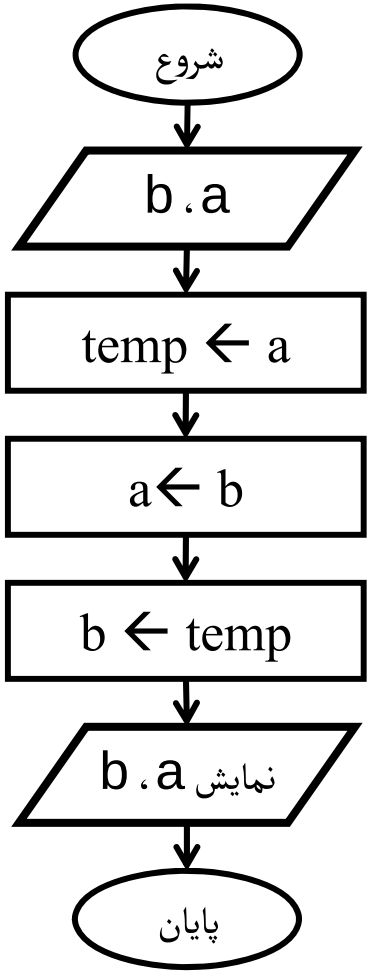
فلوچارت 3

فلوچارتی بنویسید که شعاع یک کره را دریافت کرده ، مساحت و حجم آن را محاسبه کرده و نمایش دهد.



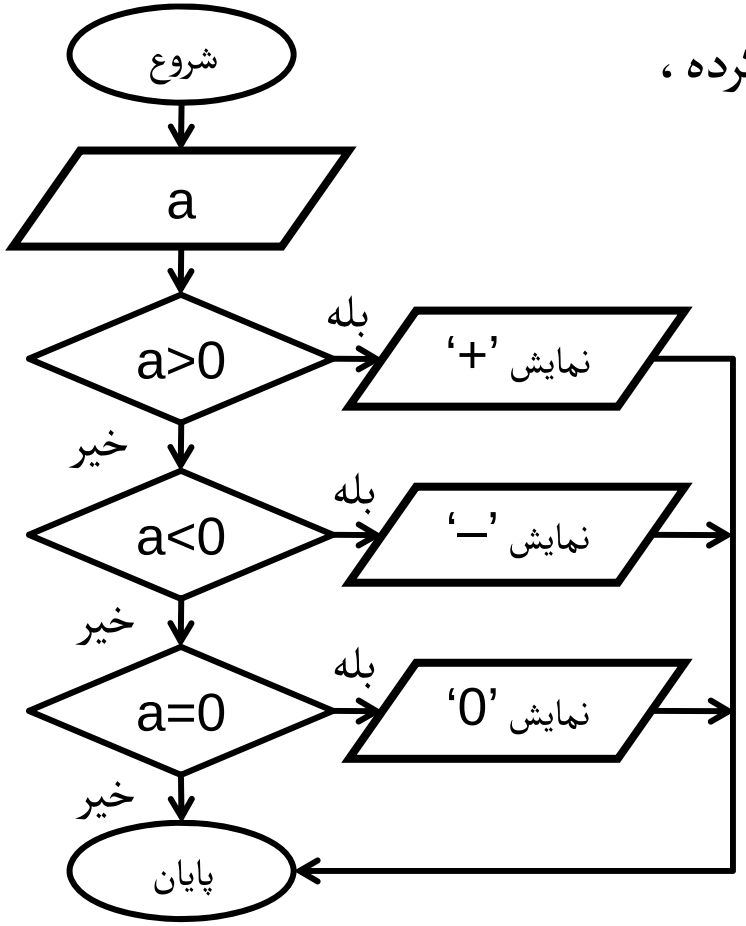
فلوچارت 4

فلوچارتی بنویسید که دو عدد را دریافت کرده ، محتوای آن دو عدد را جابجا کرده و نتیجه را نمایش دهد.



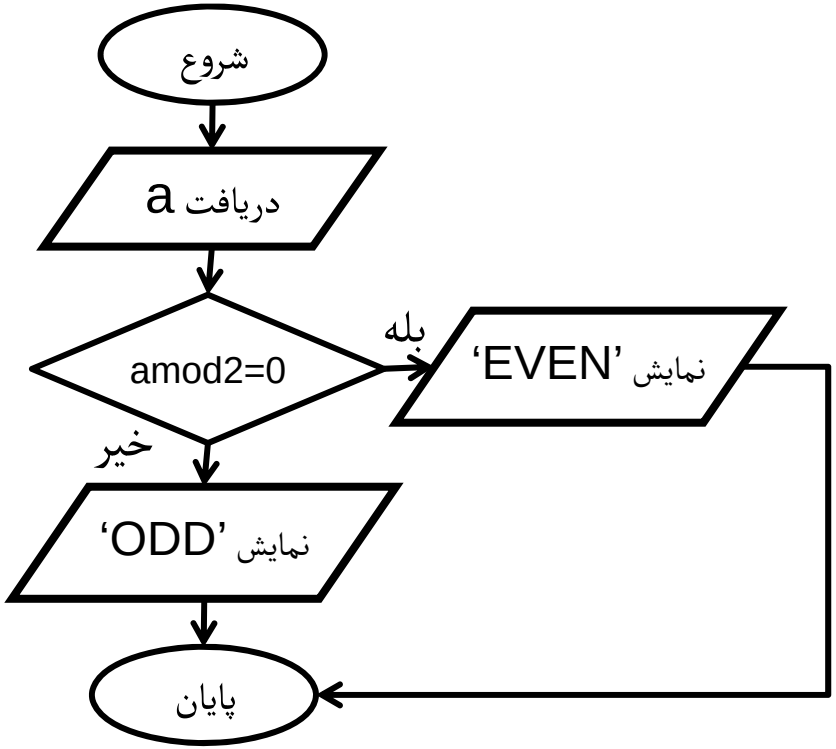
فلوچارت 5

فلوچارتی بنویسید که یک عدد دریافت کرده ،
نشان دهد ، مثبت ، منفی یا صفر است.



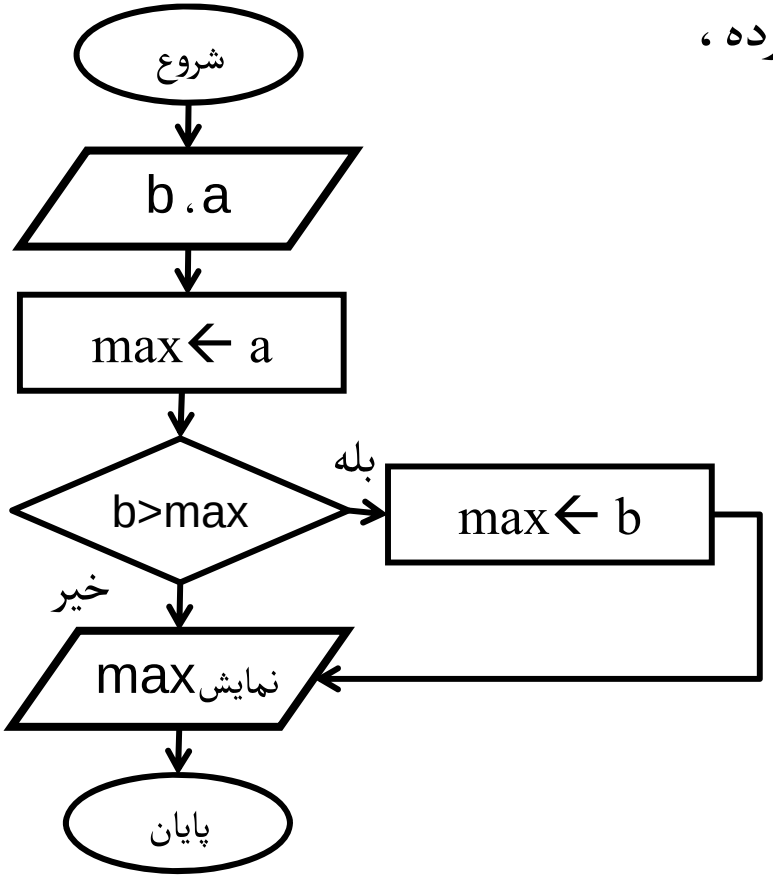
فلوچارت 6

فلوچارتی بنویسید که یک عدد دریافت کرده ، نشان دهد ، زوج است یا فرد.



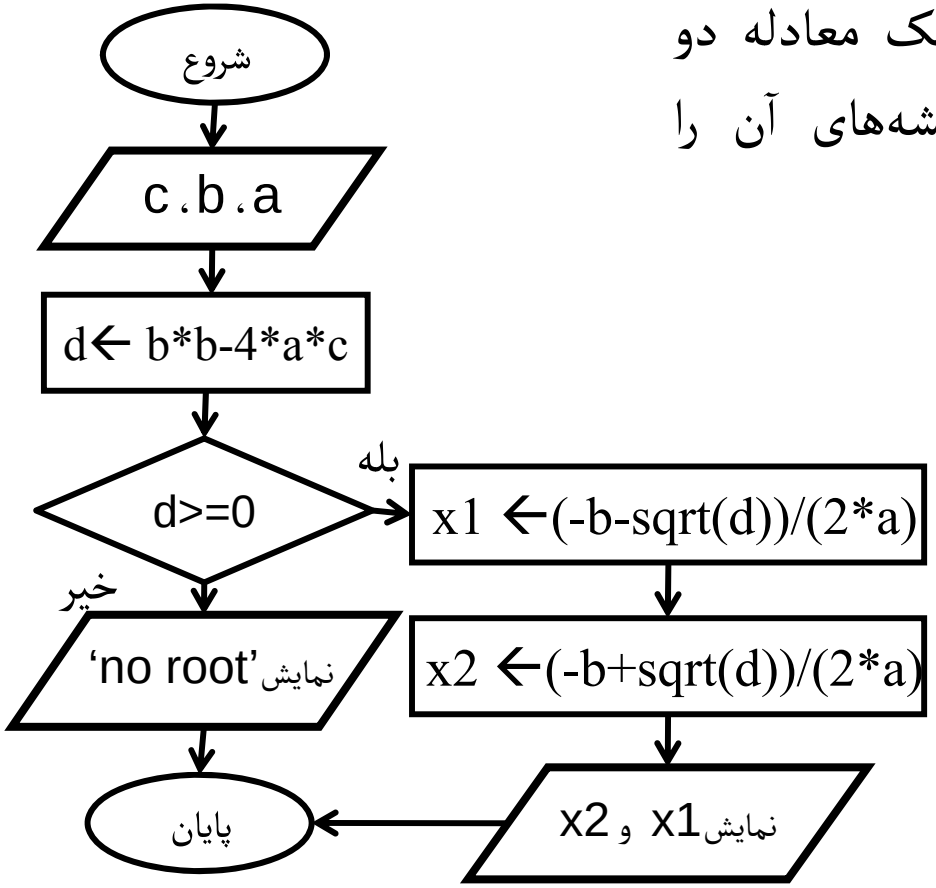
فلوچارت 7

فلوچارتی بنویسید که دو عدد دریافت کرده ،
عدد بزرگتر را نمایش دهد.



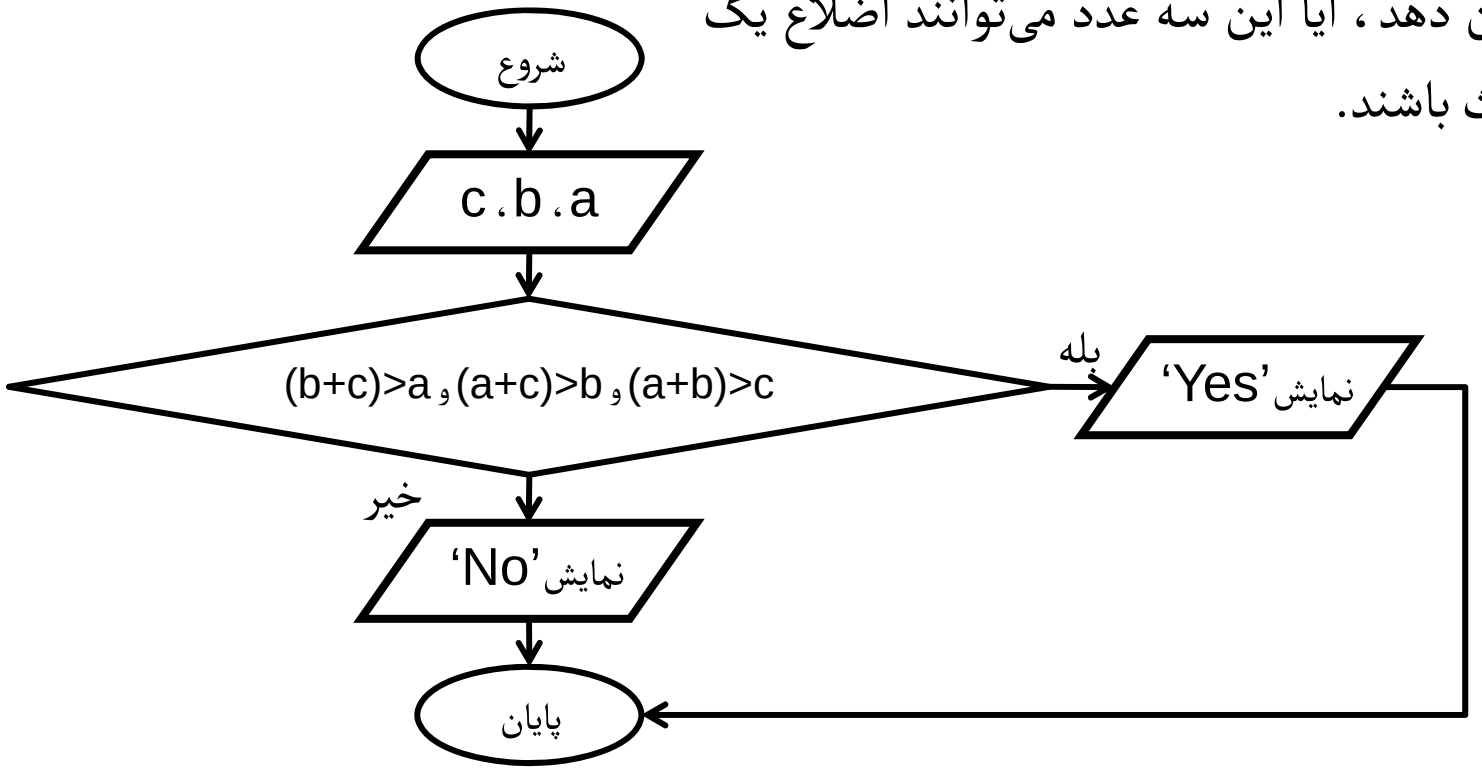
فلوچارت 8

فلوچارتی بنویسید که ضرایب یک معادله دو مجهولی را دریافت کرده، ریشه‌های آن را درصورت وجود نمایش دهد.



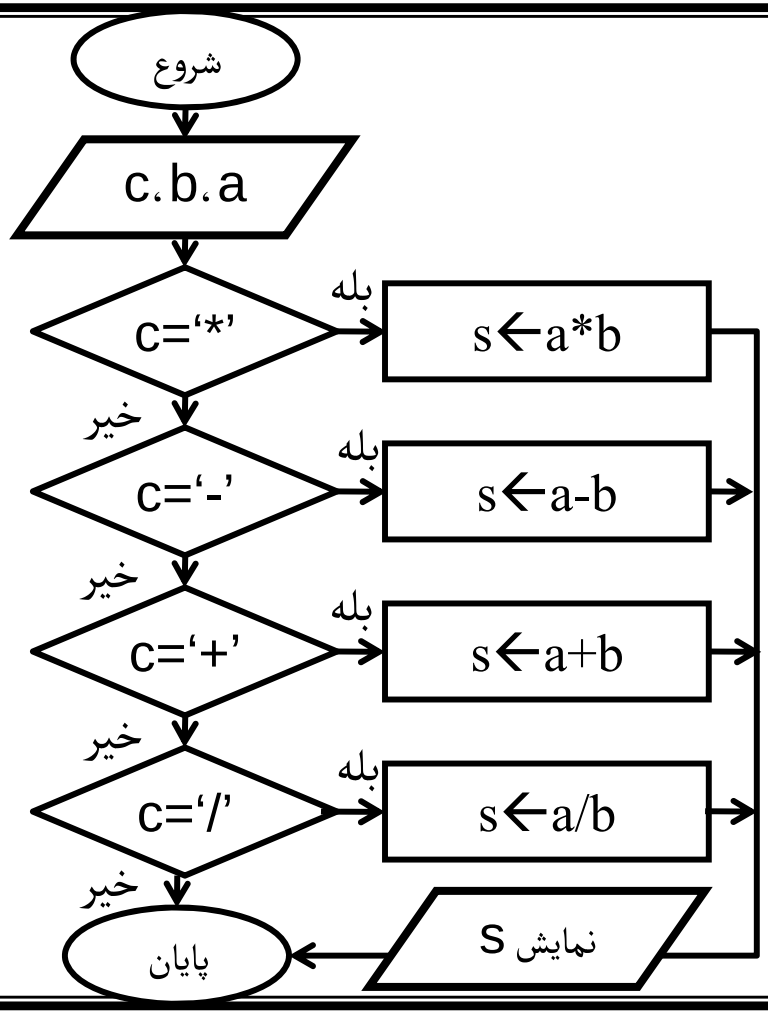
فلوچارت 9

فلوچارتی بنویسید که سه عدد دریافت کرده ، نشان دهد ، آیا این سه عدد می‌توانند اضلاع یک مثلث باشند.



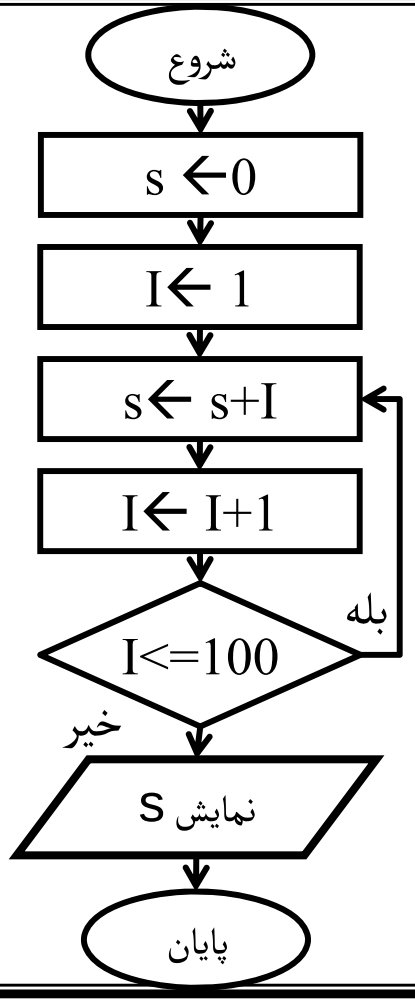
فلوچارت 10

فلوچارتی بنویسید که دو عدد به همراه یک عملگر را از ورودي دريافت کرده ، عمليات را روی اعداد انجام داده ، نتيجه را نمايش دهد.



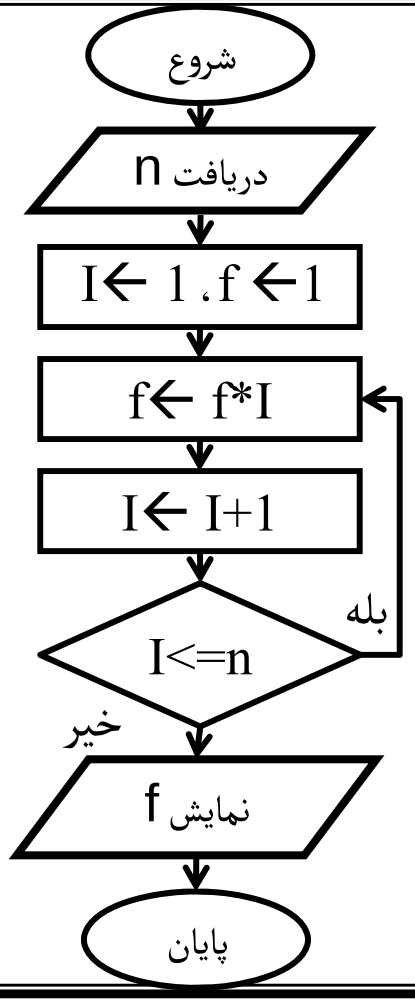
فلوچارت 11

فلوچارتی بنویسید که مجموع اعداد 1 تا 100 را محاسبه کرده و نمایش دهد.



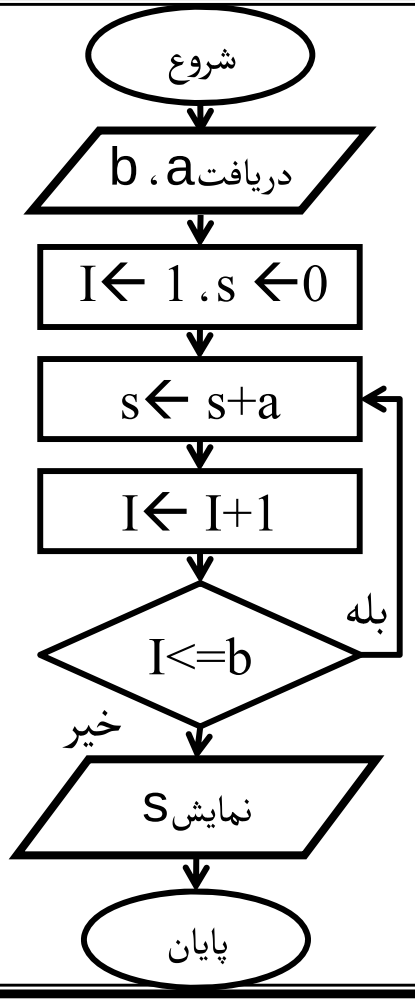
فلوچارت 12

فلوچارتی بنویسید که عدد صحیحي را از ورودي دریافت کرده و فاکتوریل آن را محاسبه کرده و نمایش دهد.



فلوچارت 13

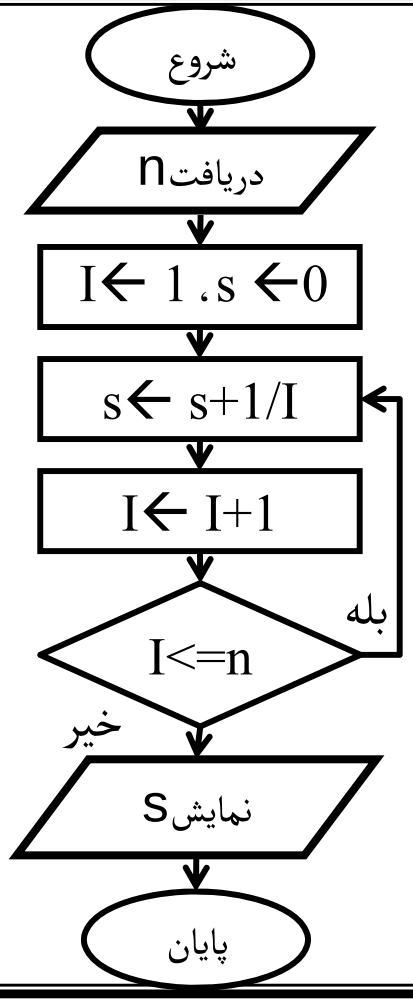
فلوچارتی بنویسید که دو عدد صحیح مثبت را از ورودي دریافت کرده و حاصلضرب آن دو را بدون استفاده از عملگر ضرب محاسبه کرده و نمایش دهد.



فلوچارت 14

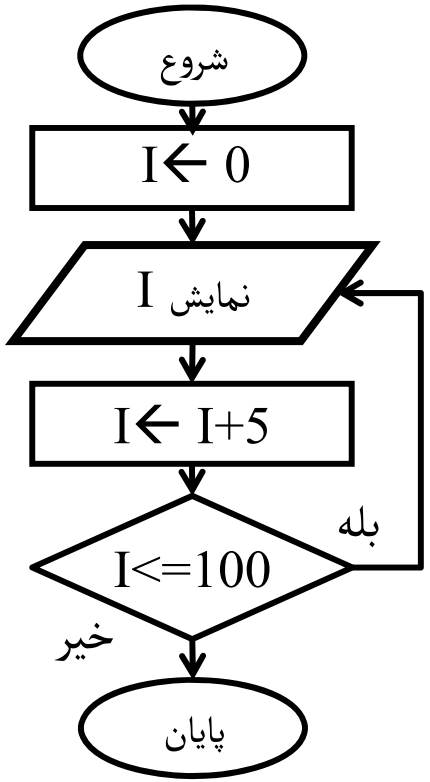
فلوچارتی بنویسید که عدد صحیح مثبت n را از ورودي دریافت کرده مجموع سری زیر را محاسبه کرده و نمایش دهد.

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$$



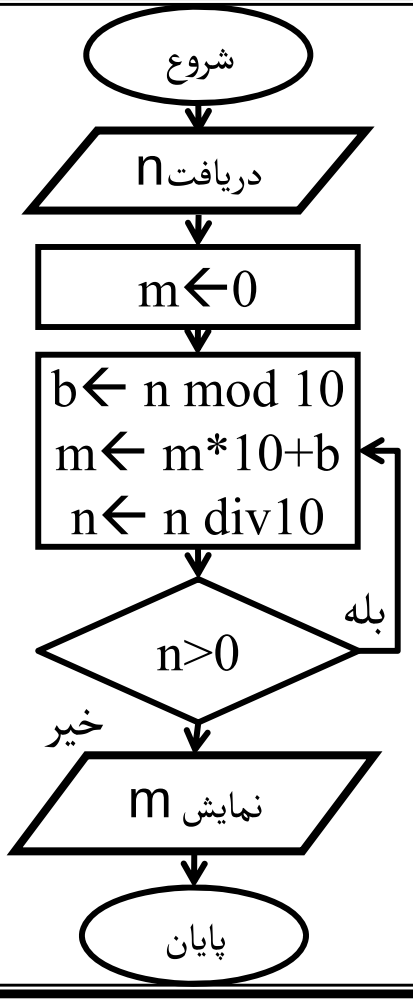
فلوچارت 15

فلوچارتی بنویسید که مضارب 5 کوچکتر از 100 را نمایش دهد.



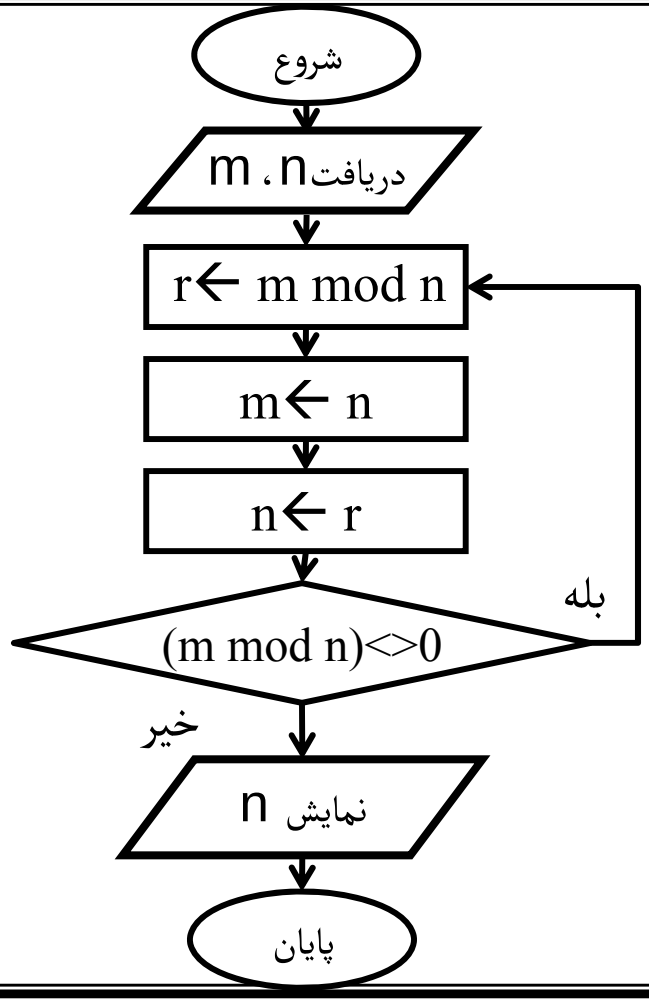
فلوچارت 16

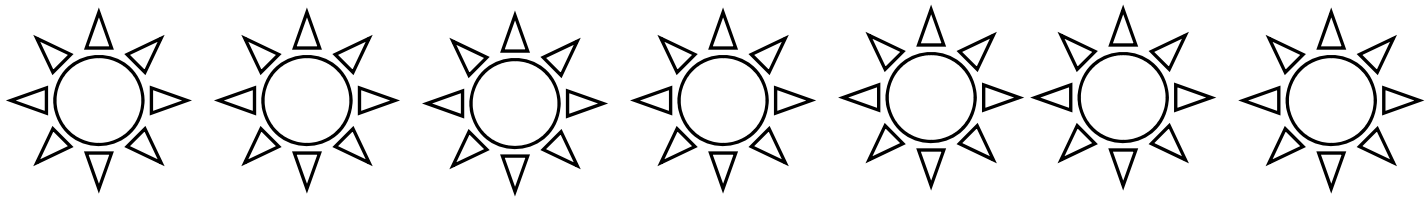
فلوچارتی بنویسید که یک عدد را دریافت کرده ،
مقلوب آن را نمایش دهد.



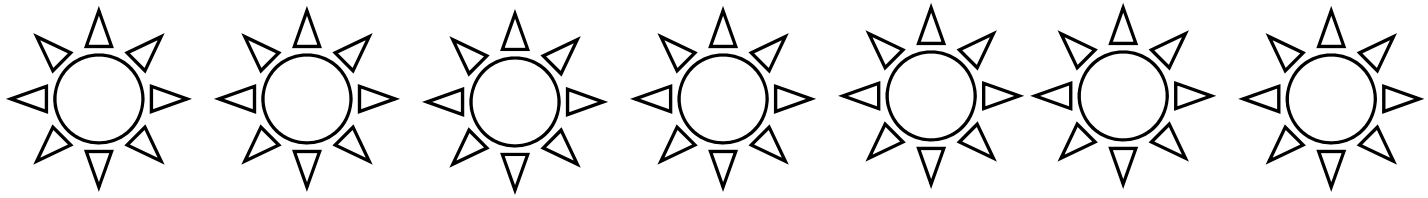
فلوچارت 17

فلوچارتی بنویسید که دو عدد را دریافت کرده ،
ب.م.م آن دو را نمایش دهد.





C++



2010

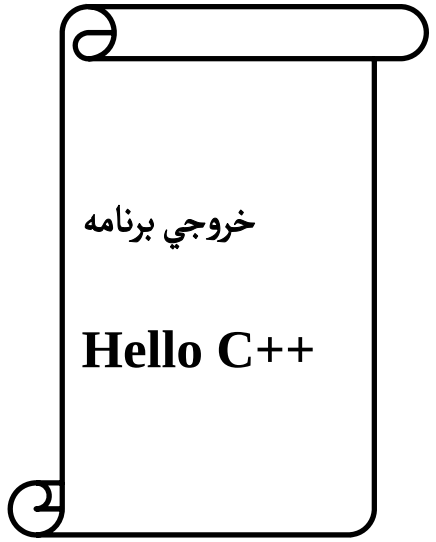
M. Damrudi

ساختار برنامه در C++

```
#include <نوع هدر فایل>
int main()
{
دستورات بدنه
return 0;
}
```

یک برنامه ساده

```
#include <iostream.h>
int main()
{
cout<<“Hello C++”;
return 0;
}
```



یک برنامه ساده

```
/*
A simple program.
This program contains all of the
basic elements
*/

#include <iostream.h>
int main()
//main is where program execution begins.
{
cout<<“Hello C++”;
return 0;
}
```

یک برنامه ساده

```
/* A simple program.
This program contains all of the
basic elements
*/
#include <iostream.h>
#include<conio.h>
int main()
//main is where program execution begins.
{
cout<<“Hello C++”;
getch();
return 0;
}
```

M. Damrudi

توضیح Comment

۱–Multiline comment : توضیحات چند خطی نامیده می‌شوند که با یک /* شروع و با */ به پایان می‌رسد.

۲– Single-line comment: توضیح یک خطی نامیده می‌شود که با // شروع شده و در همان خط پایان می‌یابد.

محتویات توضیح توسط کامپایلر نادیده گرفته می‌شود.

M. Damrudi

main()

1- همه برنامه‌ها در C++ تركيبي از يك يا چند تابع مي‌باشند.

2- تنها تابعي كه تمام برنامه‌ها دارند تابع main مي‌باشد.

3- main جايي است كه اجراي برنامه از آن‌جا شروع مي‌شود.

header files

1- زبان C++ چندين فايل تعريف مي‌كند كه فايل‌هاي سرآمد يا header files مي‌گويند.

2- هر فايل سرآمد شامل اطلاعاتي است كه براي برنامه ضروري مي‌باشند.

3- iostream.h براي پشتيباني از سيستم ورودي و خروجي مي‌باشد. براي استفاده از cout از اين فايل سرآمد استفاده مي‌شود.

4- conio.h براي استفاده از تابع getch مورد نياز مي‌باشد.

cout

يك شناسه از پيش تعريف شده است كه مخفف console output مي‌باشد و براي نمايش روي صفحه نمايش به كار مي‌رود.

انتهاي تمام دستورات در C++ به سمي‌كولون (;) ختم مي‌شود.

return 0

با اين دستور تابع main پايان مي‌يابد و مقدار صفر به پروسه فراخواننده main كه معمولا سيستم‌عامل است بازگردانده مي‌شود.

مقدار صفر نشان دهنده آن است كه برنامه به طور عادي پايان يافته است. مقادير ديگر نشان‌دنده آن است كه برنامه به علت خطايي پايان يافته است.

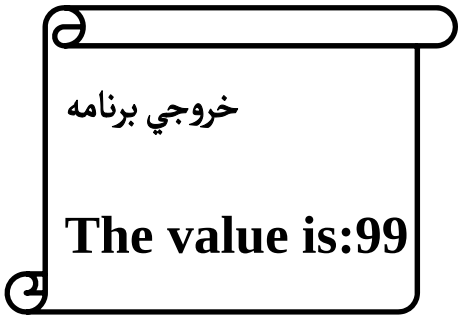
return يكي از كليدواژه‌هاي C++ مي‌باشد كه براي بازگرداندن مقداري از يك تابع به كار مي‌رود.

getch

این تابع باعث مي‌شود تا زمانيكه از ورودي كاراكتري دريافت نشده است اجراي برنامه متوقف گردد.

دومین برنامه ساده

```
#include <iostream.h>
#include<conio.h>
int main()
{
int num;
num=99;
cout<<“The value is:”;
cout<<num;
getch();
return 0;
}
```



تعريف متغير

متغير محل نامگذاري شده‌اي از حافظه است كه مي‌توان مقداري در آن قرار داد.

محتواي يك متغير قابل تغيير است و ثابت نيست.

براي تعريف متغير ابتدا نوع متغير و سپس نام دلخواهي براي آن تعيين مي‌كنيم.

```
type variable;
```

با علامت = مقدار 99 را داخل متغير num قرار داده مي‌شود.

مي‌توان چند متغير را همزمان تعريف نمود. `int a,b;`

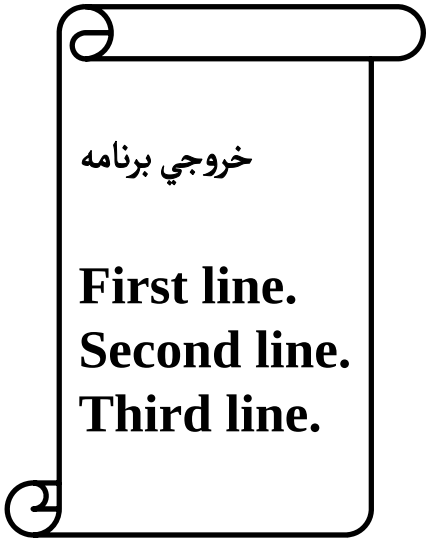
نوع داده‌های اصلی

کلمه کلیدی	دامنه	شرح
bool	مقدار درست یا نادرست(منطقی)	true و false
char	۱۲۷ تا ۱۲۸–	کاراکترهای ۸ بیتی 'A'
int	۳۲۷۶۷ تا ۳۲۷۶۸–	عدد صحیح
float	۴/۳E+۳۸ تا ۴/۳E–۳۸	مقدار اعشاری
double	۷/۱E+۳۰۸ تا ۷/۱E–۳۰۸	مقدار اعشاری با دقت مضاعف
void	اعمال نمی‌شود	بیانگر یک عبارت بدون مقدار
wchar_t	۰ تا ۶۵۵۳۵	کاراکترهای عریض (زبان چینی)

سومین برنامه ساده

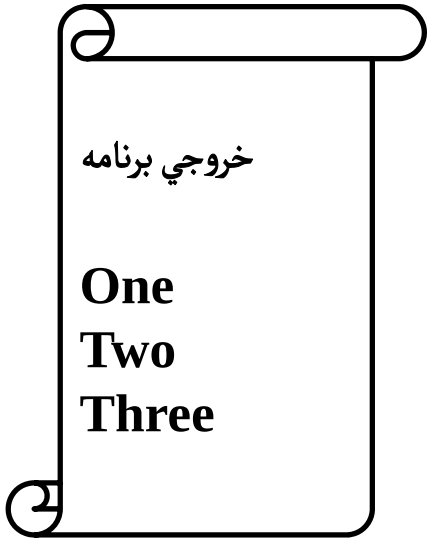
```
#include <iostream.h>
#include<conio.h>
int main()
{
cout<<“First line.\n”;
cout<<“Second line.\n”;
cout<<“Third line.\n”;
getch();
return 0;
}
```

کاراکتر خط جدید
(newline)



سومین برنامه ساده

```
#include <iostream.h>
#include<conio.h>
int main()
{
cout<<“One\nTwo\nThree”;
getch();
return 0;
}
```



دریافت اطلاعات از کاربر با دستور cin

```
#include <iostream.h>
#include<conio.h>
int main()
{
int a,b;
cin>>a>>b;
cout<<"The value a is:"<<a<<"\n";
cout<<"The value b is:"<<b<<"\n";
getch();
return 0;
}
```

1 مقادیر سه متغیر را دریافت کرده ، میانگین آنها را محاسبه کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

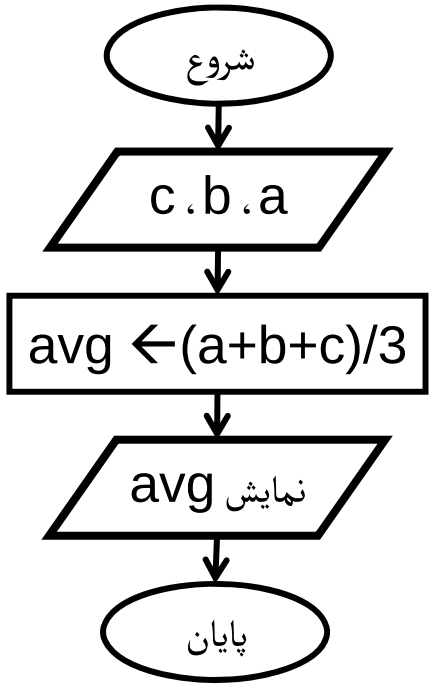
2- a ، b و c را از ورودی دریافت کن

3- avg←(a+b+c)/3

4- مقدار avg را نمایش بده

5-پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
float a,b,c;
float avg;
cin>>a>>b>>c;
avg=(a+b+c)/3;
cout<<avg;
getch();
return 0;
}
```

2 شعاع کره ای را دریافت کرده مساحت و حجم کره را محاسبه کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

2- r را از ورودی دریافت کن

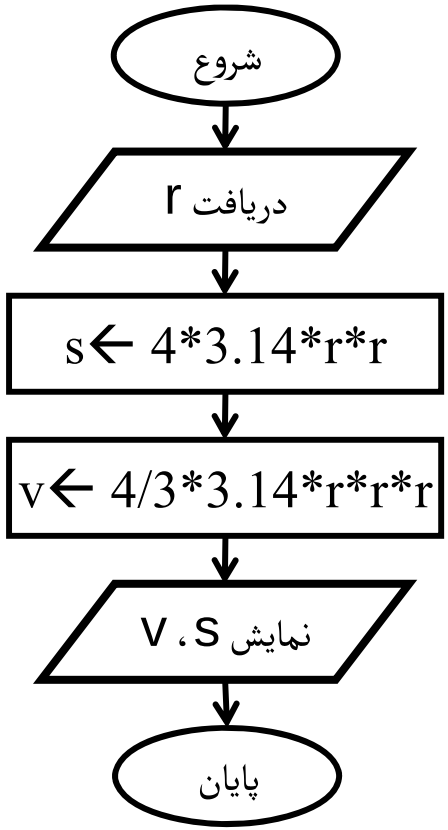
3- s← 4*3.14*r*r

4- v← 4/3*3.14*r*r*r

5- مقدار S و V را نمایش بده

6- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
float r,s,v;

cin>>r;
s=4*3.14*r*r;
v=4/3*3.14*r*r*r;
cout<<s<<v;
getch();
return 0;
}
```

3 چهار عدد a ، b ، c ، و d را دریافت کرده ، مجموع مکعبات آنها را محاسبه کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

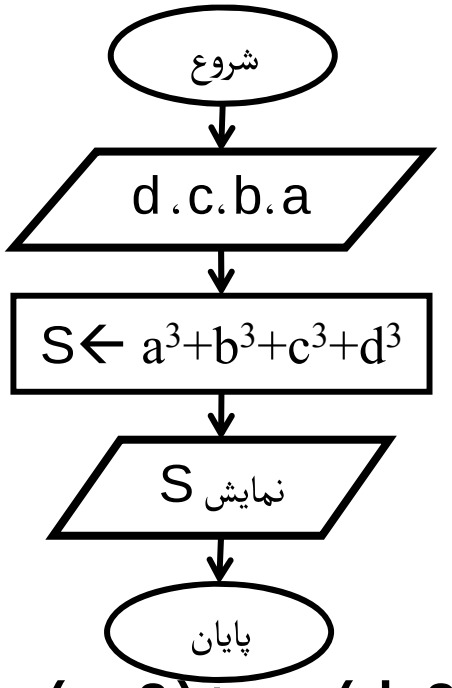
2- a ، b ، c ، وd را از ورودی دریافت کن

3- $S \leftarrow a^3+b^3+c^3+d^3$

4- مقدار S را نمایش بده

5- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
int main(){
int a, b, c, d, s;

cin>>a;
cin>>b;
cin>>c;
cin>>d;

s=pow(a,3)+pow(b,3)+pow(c,3)+pow(d,3);
cout<<s;
getch();
return 0;
}
```

2013 *M. Damrudi*

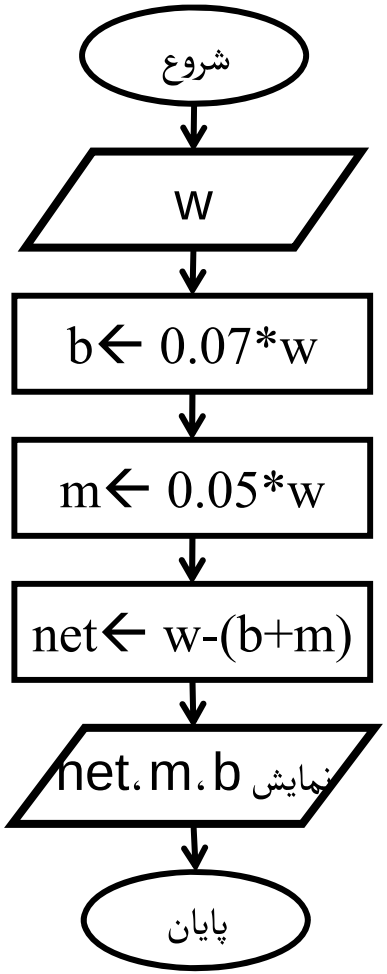
4 حقوق ناخالص یک کارمند را دریافت کرده ، 7% بیمه ، 5% حق مسکن از آن کسر کرده ، حقوق خالص ، حق بیمه و حق مسکن را محاسبه کرده و نتیجه را نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
float w, m, b, net;

cout<< “Enter W:”;
cin>>w;
b=0.07*w;
m=0.05*w;
net=w-(b+m);
cout<<“Insurance:”<<b;
cout<<“\nHousing:”<<m;
cout<<“\nNet:”<<net;
getch();
return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- w را از ورودی دریافت کن

3- b← 0.07*w

4- m← 0.05*w

5- net← w-(b+m)

6- مقدار net را نمایش بده

7-پایان

2013

M. Damrudi

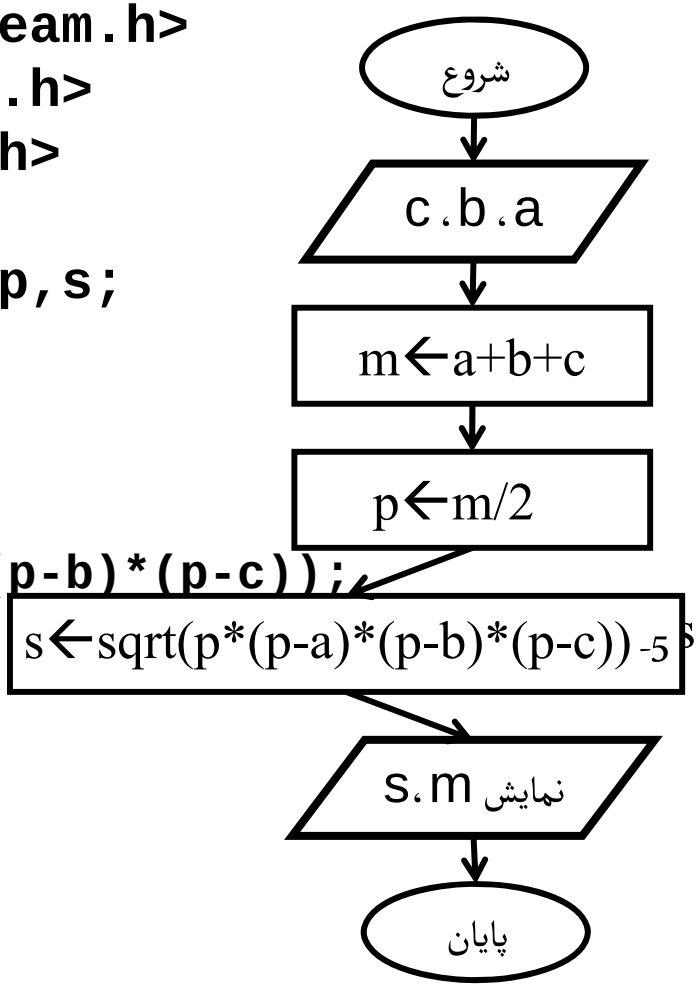
5 طول اضلاع یک مثلث را دریافت کرده ، محیط و مساحت آن را محاسبه کرده و نتیجه را نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
int main(){
float a,b,c,m,p,s;

cin>>a>>b>>c;
m=a+b+c;
P=m/2;
S=sqrt(p*(p-a)*(p-b)*(p-c));
Cout<<m<<s;
getch();
return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- a ، b و c را از ورودی دریافت کن

3- m←a+b+c

4- p←m/2

5- s←sqrt(p*(p-a)*(p-b)*(p-c))

6- مقدار m و s را چاپ کن

7- پایان

6 مقادیر متغیرهای a و b را دریافت کرده ، سپس مقادیر آنها را با یکدیگر تعویض کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

2- a ، b را از ورودی دریافت کن

3- temp ← a

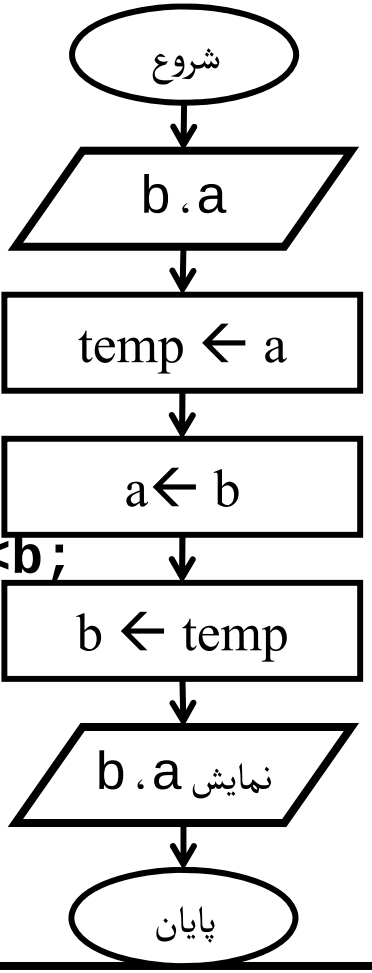
4- a← b

5- b ← temp

5- مقدار a و b را نمایش بده

6- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
int a, b, temp;

cin>>a>>b;
temp=a;
a=b;
b=temp;
cout<<"a="<<a<<"\nb="<<b;
getch();
return 0;
}
```

7 زمان t را به صورت عدد اعشار برحسب ساعت دریافت کرده ، به ساعت ، دقیقه و ثانیه نمایش دهد.

الگوریتم

1- شروع

2- t را دریافت کن

3- h ← int(t)

4- x ← t – h

5- x ← x * 60

6- m ← int (x)

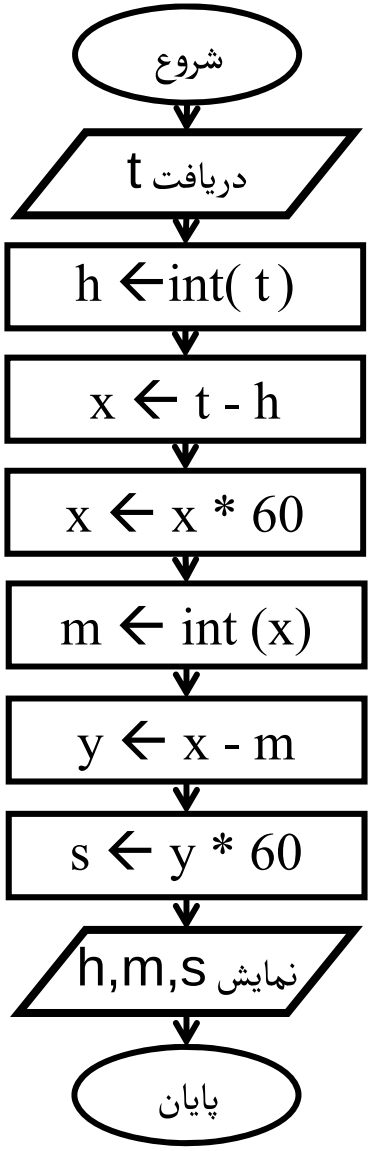
7- y ← x – m

8- s ← y * 60

9- h ، m ، s را نمایش بده

10- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
int main(){
 int h,m,s;
 float t,x,y;
 cin>>t;
 h=int(t);
 x=t-h;
 x=x*60;
 m=int(x);
 y=x-m;
 s=y*60;
 cout<<h<<':'<<m;
 cout<<':'<<s;
 getch();
 return 0;
}
```

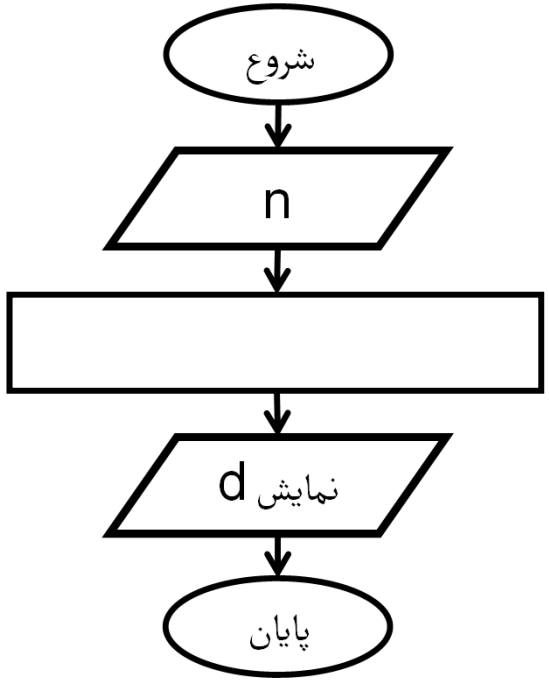
M. Damrudi

2013

8 یک عدد را دریافت کرده ، تعداد ارقام آن را با استفاده از لگاریتم تعیین کرده و نتیجه را نمایش دهد.

الگوریتم

فلوچارت



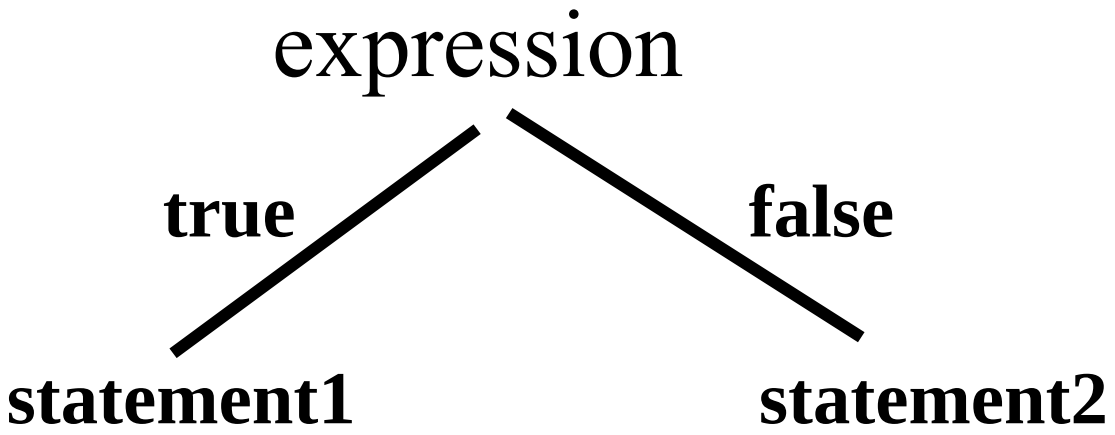
برنامه

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
int main(){
int n,d;

cout<<“Enter n:”;
cin>>n;
d=int(log10(n))+1;
cout<<d;
getch();
return 0;
}
```

دستور if

```
if (expression) statement1;
        else statement2;
```



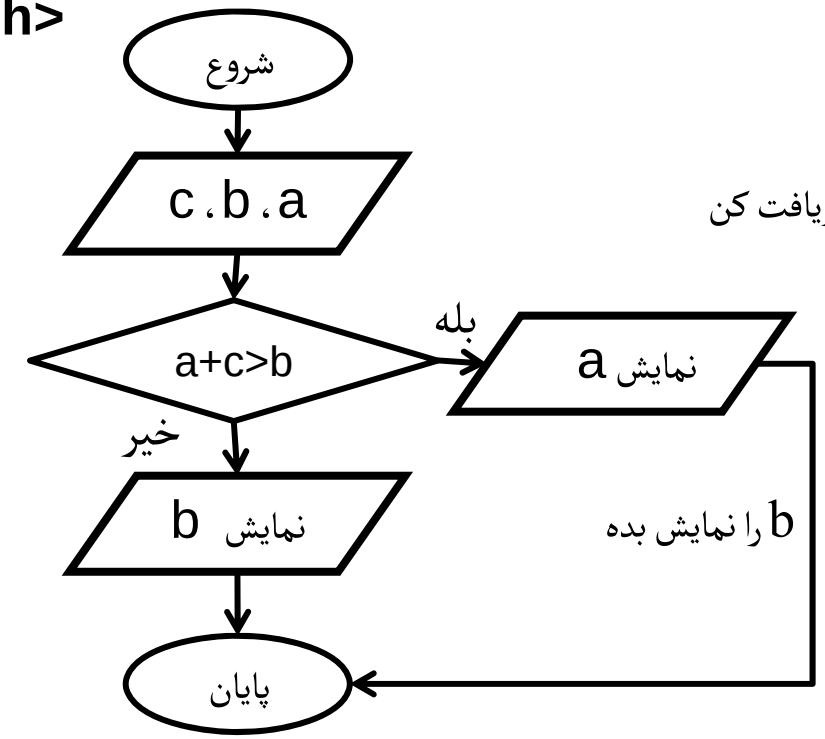
9 سه عدد را دریافت کرده ، اگر a+c>b باشد مقدار a و در غیراینصورت مقدار b را نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
int a,b,c;

cin>>a>>b>>c;
if(a+c>b)
   cout<<a;
else
   cout<<b;
getch();
return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- a و b و c را از ورودی دریافت کن

3- اگر a+c>b آنگاه مقدار a را نمایش بده

درغیراینصورت مقدار b را نمایش بده

4- پایان

10 یک عدد را دریافت کرده ، مثبت بودن و صحیح بودن آن را با پیغام مناسبی تعیین کنید.

الگوریتم

1- شروع

2- عدد n را دریافت کن

3- اگر n>0 آنگاه

اگر int(n)-n=0 آنگاه

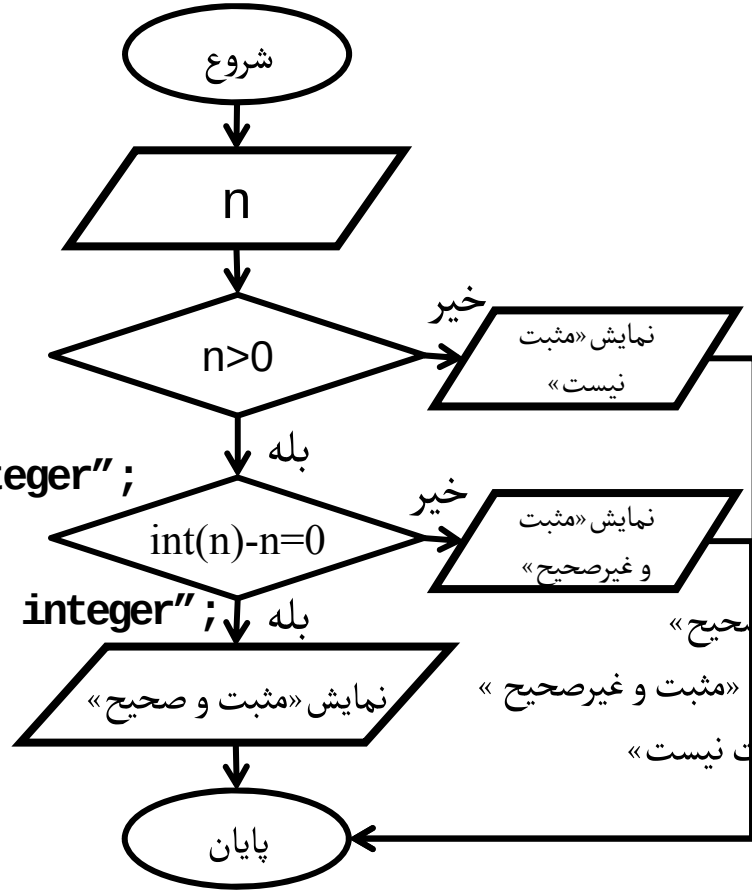
نمایش بده «مثبت و صحیح»

در غیر اینصورت نمایش بده «مثبت و غیرصحیح »

درغیراینصورت نمایش بده «مثبت نیست»

4- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
float n;
cin>>n;
if(n>0)
{
  if(int(n)-n==0)
    cout<<“positive & integer”;
  else
    cout<<“positive & not integer”;
}
else
cout<<“not positive”;
getch();
return 0;
}
```

عملگرهای ریاضی

عملگر	عمل
*	ضرب
/	تقسیم
+	جمع
-	تفریق
%	باقیمانده تقسیم

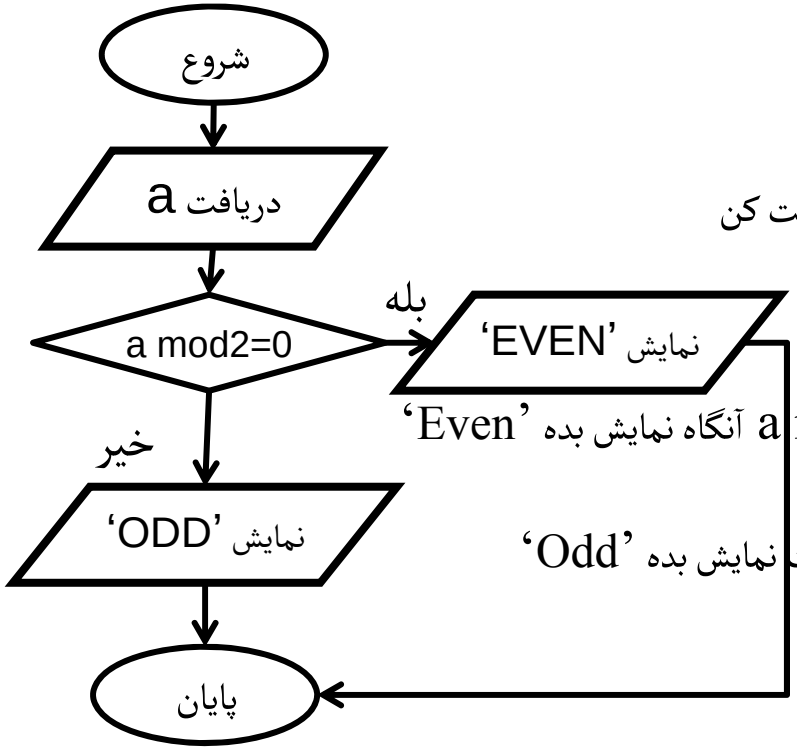
11 یک عدد صحیح مثبت را دریافت کرده ، سپس زوج یا فرد بودن آن را با پیام مناسبی نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
int main(){
int a;

cin>>a;
if(a%2==0)
   cout<<“Even”;
else
   cout<<“Odd”;
getch();
return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- a را از ورودی دریافت کن

3- اگر a mod 2 =0 آنگاه نمایش بده ‘Even’

در غیر اینصورت نمایش بده ‘Odd’

4- پایان

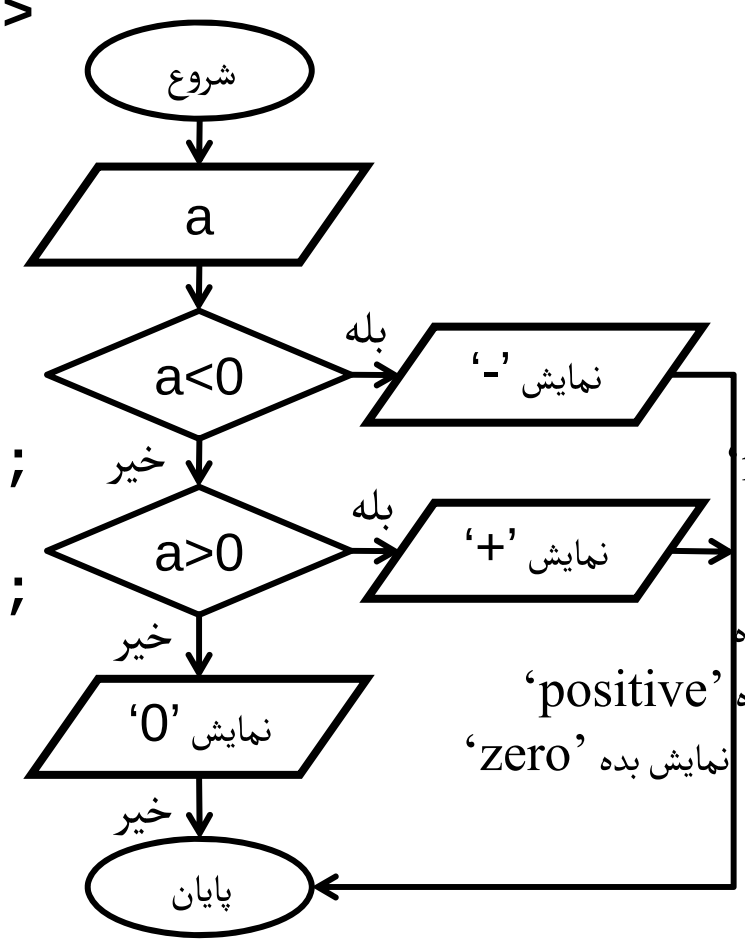
12 یک عدد دریافت کرده ، نشان دهد ، مثبت ، منفی یا صفر است.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
float a;

cin>>a;
if(a<0)
   cout<<“negative”;
else if(a>0)
   cout<<“positive”;
else
   cout<<“zero”;
getch();
return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- a را از ورودی دریافت کن

3- اگر a<0 آنگاه
'negative' نمایش بده
در غیراینصورت اگر a>0 آنگاه
'positive' نمایش بده
در غیراینصورت 'zero' نمایش بده

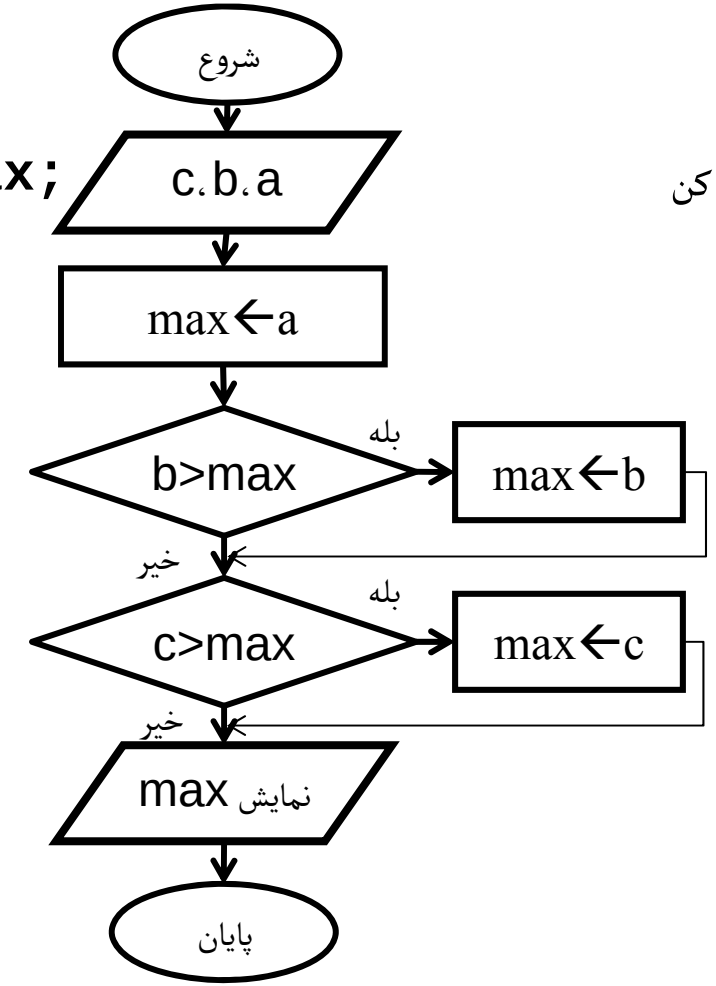
4- پایان

13 سه عدد را خوانده ، بزرگترین عدد را تعیین کرده و نتیجه را نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 double a, b, c, max;
 cin>>a;
 cin>>b;
 cin>>c;
 max=a;
 if(b>max)
    max=b;
 if(c>max)
    max=c;
 cout<<max;
 getch();
 return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- a و b و c را از ورودی دریافت کن

3- max ← a

4- اگر b> max آنگاه
max ← b

5- اگر c> max آنگاه
max ← c

6- مقدار max را نمایش بده

7- پایان

عملگرهای منطقی

عمل	عملگر
AND	&&
OR	||
NOT	!

مقادیر منطقی در زبان C

==0	=>	false
!=0	=>	true

14 سه عدد را به عنوان اضلاع یک مثلث دریافت کرده ، نشان دهد ، آیا این سه عدد می‌توانند یک مثلث را ایجاد نمایند.

الگوریتم

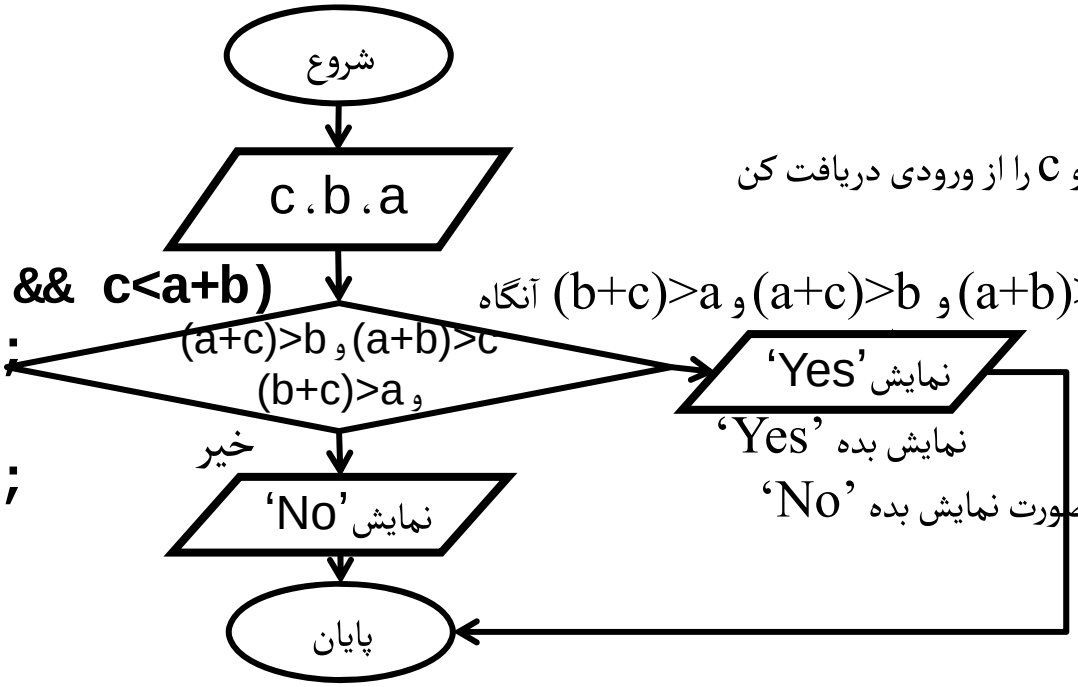
1- شروع

2- a و b و c را از ورودی دریافت کن

3- اگر (a+b)>c و (a+c)>b و (b+c)>a آنگاه
'Yes' نمایش
'Yes' نمایش بده
درغیراینصورت 'No' نمایش بده

4- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 double a, b, c;
 cin>>a>>b>>c;
 if(a<b+c && b<a+c && c<a+b)
     cout <<"Yes";
 else
     cout << "No";
 getch();
 return 0;
}
```

15 سه عدد را خوانده ، به ترتیب صعودی مرتب کرده و نمایش دهد.

```
#include<iostream.h>
#include<conio.h>
int main(){
 int a, b, c, temp;
 cin>>a>>b>>c;
 if(a>b)
 {
  temp=a;
  a=b;
  b=temp;
 }
 if(a>c)
 {
  temp=a;
  a=c;
  c=temp;
 }
 if(b>c)
 {
  temp=b;
  b=c;
  c=temp;
 }
 cout<<a<<b<<c;
 getch();
 return 0;
}
```

الگوریتم

1- شروع

2- a و b و c را از ورودی دریافت کن

3- اگر a>b آنگاه

temp← a

a← b

b← temp

4- اگر a>c آنگاه

temp← a

a← c

c← temp

5- اگر b>c آنگاه

temp← b

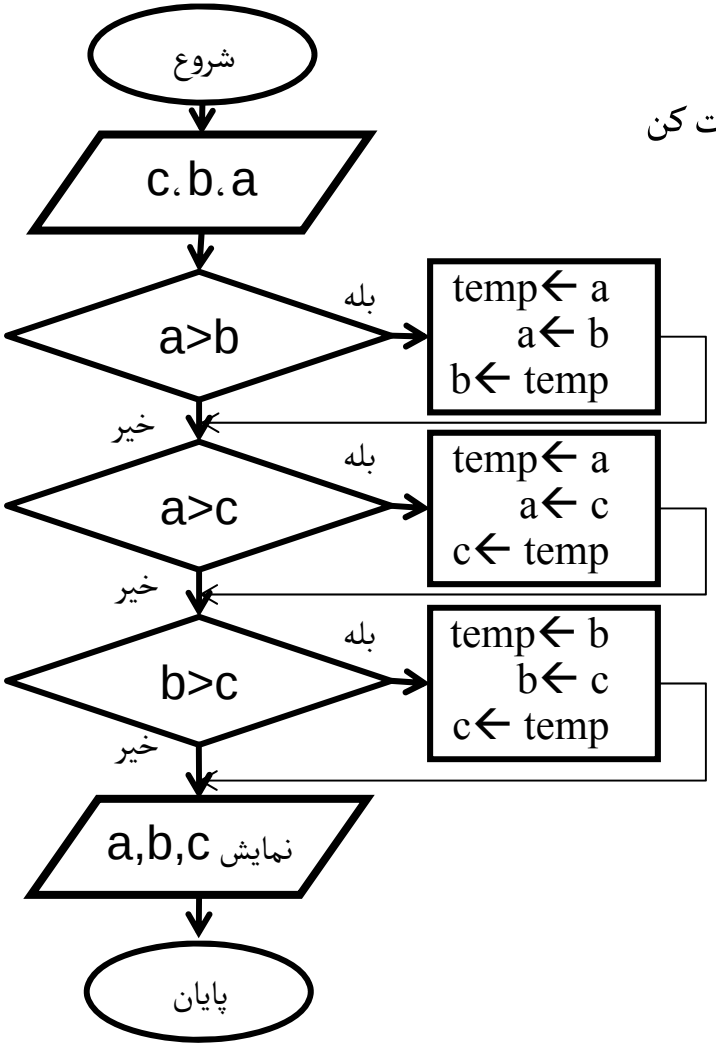
b← c

c← temp

6- مقدار a ، b و c را نمایش بده

7- پایان

فلوچارت



2013

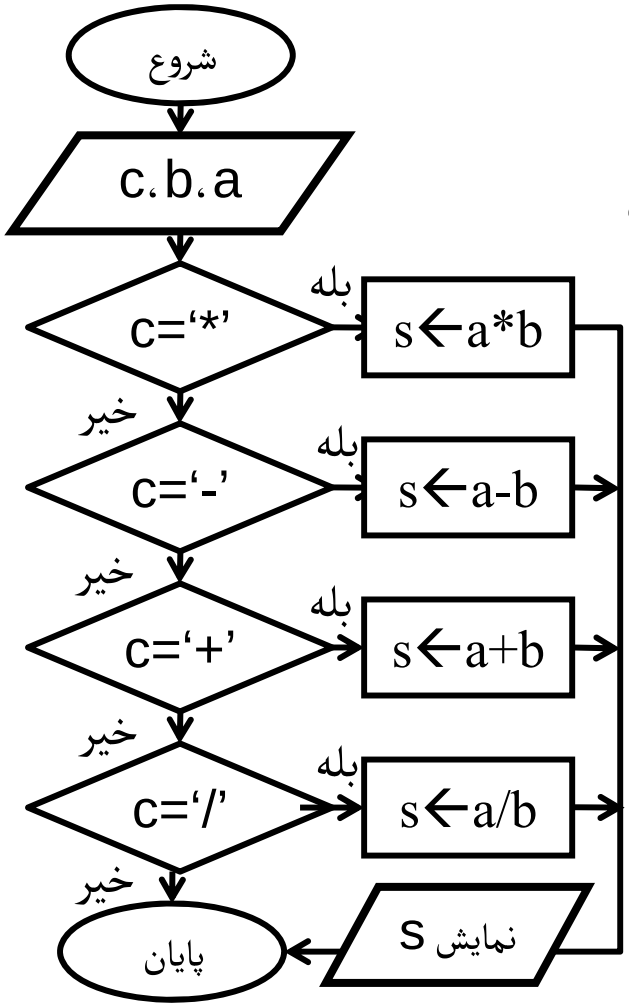
M. Damrudi

16 دو عدد ویکی از چهار عملگر اصلی را دریافت کرده ، عملیات را روی اعداد انجام داده ، نتیجه را نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 float a, b;
 char c;
 cin>>a>>b>>c;
 if(c=='*')
   cout <<"*:"<<a*b;
 else if(c=='-')
   cout <<"-:"<<a-b;
 else if(c=='+')
   cout <<"+:"<<a+b;
 else if(c=='/')
   cout <<"/:"<<a/b;
 else
   cout <<"wrong opt";
 getch();
 return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- a و b و c را از ورودی دریافت کن

3- اگر c='*' آنگاه s← a * b

4- اگر c='-' آنگاه s← a - b

5- اگر c='+' آنگاه s← a + b

6- اگر c='/' آنگاه s← a / b

7- مقدار S را نمایش بده

4- پایان

دستور switch

شکل کلي دستور به صورت زير است:

```
switch(expression){
      case(val1):
      {
      instructions
      }
      break;
      case(val2):
      {
      instructions
      }
      break;

      default:
      {
      instructions
      }
}
```

به کار بردن break

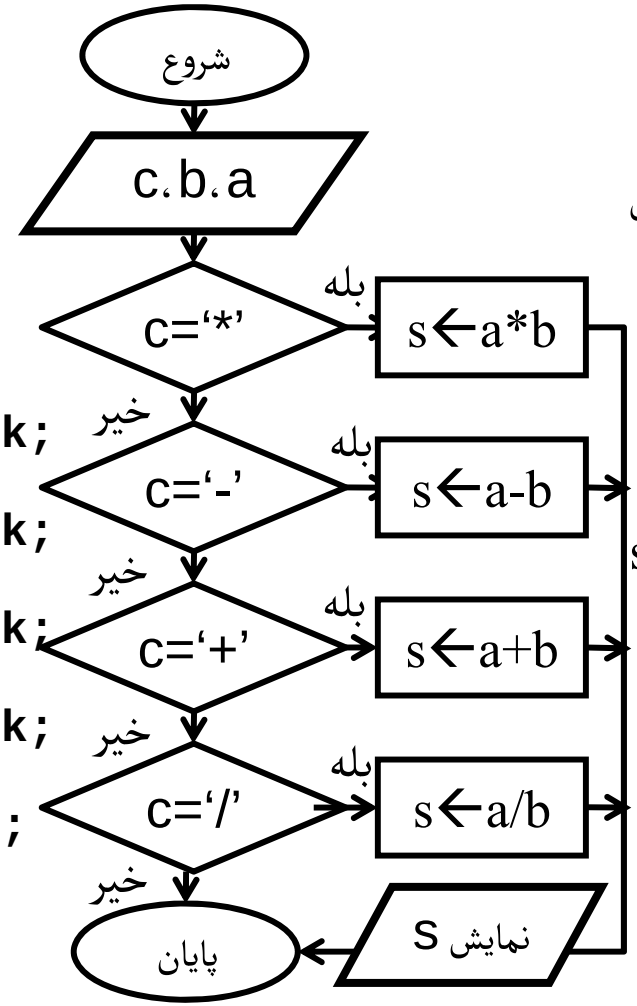
```
#include <iostream.h>
#include <conio.h>
int main(){
      int num;
      cin>> num;
      switch (num) {
            case 1:
            cout<< “First Case”; break;
            case 2:
            cout<< “Second Case”;break;
            case 3:
            cout<< “Third Case”; break;
            case 4:
            cout<< “Forth Case”; break;
            case 5:
            cout<< “Fifth Case”; break;
            default:
            cout<<“Nothing”;
      }
      getch();
      return 0;
}
```

16 دو عدد ویکی از چهار عملگر اصلی را دریافت کرده ، عملیات را روی اعداد انجام داده ، نتیجه را نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 float a, b;
 char c;
 cin>>a>>b>>c;
 switch (c)
 {
 case '*':
   cout <<"*:"<<a*b;break;
 case '-':
   cout <<"-:"<<a-b;break;
 case '+':
   cout <<"+:"<<a+b;break;
 case '/':
   cout <<"/:"<<a/b;break;
 default:
   cout <<"wrong operat";
 }
 getch();
 return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- a و b و c را از ورودی دریافت کن

3- اگر c='*' آنگاه s← a * b

4- اگر c='-' آنگاه s← a - b

5- اگر c='+' آنگاه s← a + b

6- اگر c='/' آنگاه s← a / b

7- مقدار S را نمایش بده

4- پایان

for حلقه

حالت كلي اين دستور به صورت زير مي‌باشد.

for (initialization; expression; increment/decrement) statement;

مقداردهي اوليه متغير كنترل كننده حلقه

تنها يكبار و پيش از شروع حلقه اجرا مي‌گردد.

شرط حلقه
در آغاز هر تكرار انجام مي‌شود.

افزايش يا كاهش مقدار متغير كنترل كننده حلقه

پس از اجراي بدنه حلقه ، اجرا مي‌شود.

حلقه for

كاهش و افزايش متغير كنترل كننده مي‌تواند بيش از يك واحد باشد.

```
for(num=10; num>0; num= num-4)
```

```
for(num=0; num<11; num= num+4)
```

استفاده از عملگرهای افزایشی و کاهشی

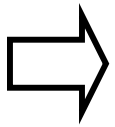
```
i=i+1;   =   i++;
```

```
i=i-1;   =   i--;
```

استفاده از عملگرهای افزایشی و کاهشی

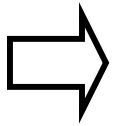
مي‌توان عملگرها را پيش از متغيرها هم به كار برد. اما معناي متفاوتي دارند.

```
j=i++;
```



ابتدا مقدار i به j نسبت داده شده و سپس i يك واحد اضافه مي‌شود.

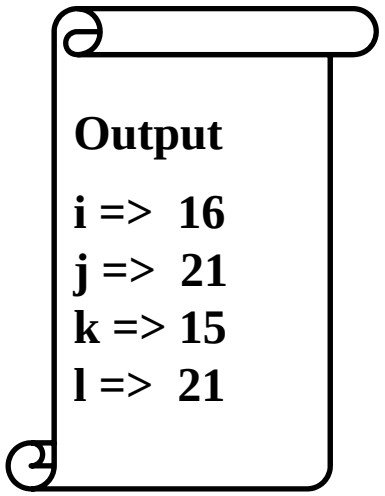
```
j=++i;
```



ابتدا مقدار i يك واحد اضافه مي‌شود و سپس مقدار i به j نسبت داده مي‌شود.

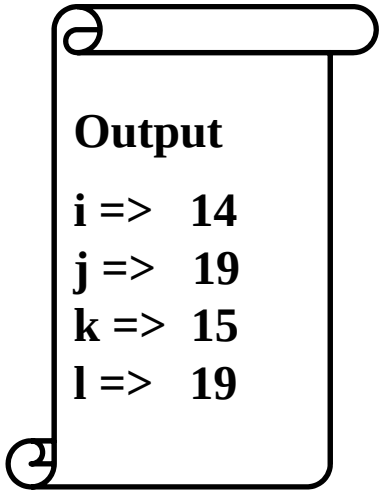
مثال

```
#include <iostream.h>
#include <conio.h>
int main()
{
int i, j, k, l;
i=15;
j=20;
k=i++;
l=++j;
cout<<i << j<< k<< l;
getch();
return 0;
}
```



مثال

```
#include <iostream.h>
#include <conio.h>
int main()
{
int i, j, k, l;
i=15;
j=20;
k=i--;
l=--j;
cout<<i << j<< k<< l;
getch();
return 0;
}
```



17 مجموع اعداد صحیح و مثبت از 1 تا 100 را محاسبه کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

2- $s \leftarrow 0$

3- $I \leftarrow 1$

4- $s \leftarrow s + I$

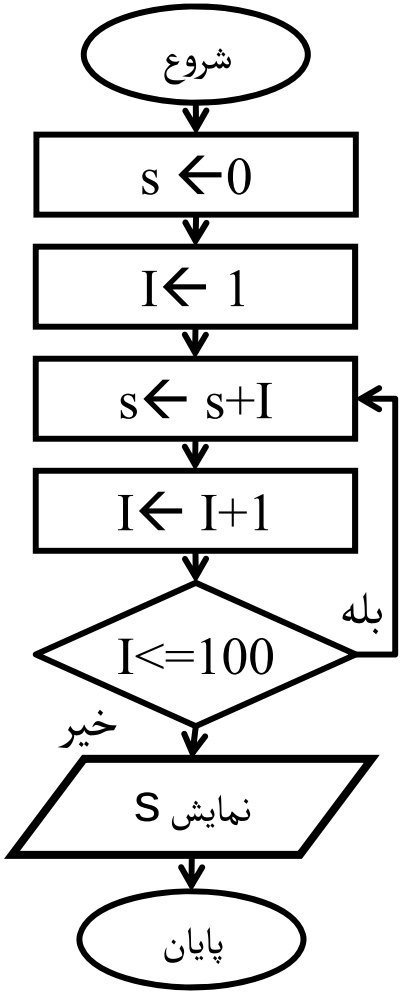
5- $I \leftarrow I + 1$

6- اگر $I<=100$ آنگاه برو به مرحله 4

7- مقدار S را نمایش بده

8- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 int I, s;
 s=0;
 I=1;
 for(I=1; I<=100; I++)
     s+=I;
 cout<<s;
 getch();
 return 0;
}
```

حلقه while

حالت كلي اين دستور به صورت زير مي‌باشد.

while (expression) statement;

شرط حلقه

تا زمانیکه شرط برقرار باشد دستورات اجرا می‌شود.

شرط حلقه در ابتدای حلقه کنترل می‌شود.

17 مجموع اعداد صحیح و مثبت از 1 تا 100 را محاسبه کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

2- $s \leftarrow 0$

3- $I \leftarrow 1$

4- $s \leftarrow s + I$

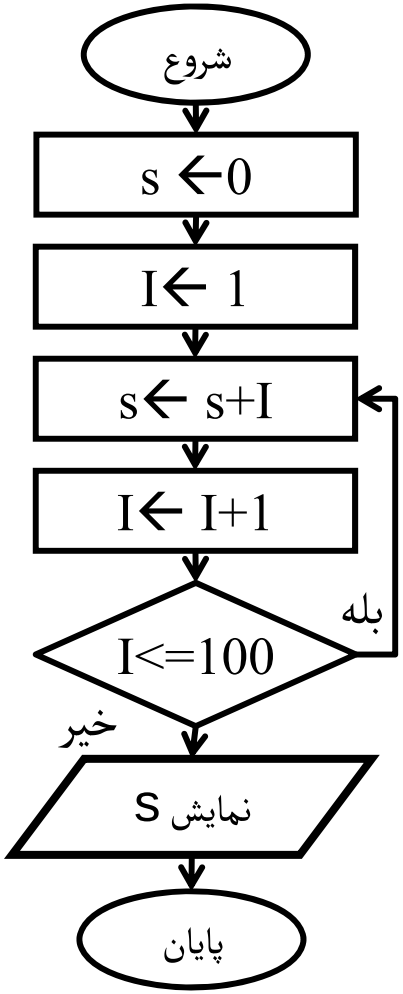
5- $I \leftarrow I + 1$

6- اگر $I<=100$ آنگاه برو به مرحله 4

7- مقدار S را نمایش بده

8- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 int I, s;
 s=0;
 I=1;
 while(I<=100)
 {
     s+=I;
     I++;
 }
 cout<<s;
 getch();
 return 0;
}
```

حلقه do

حالت كلي اين دستور به صورت زير مي‌باشد.

```
do {
        statements
 } while (expression);
```

شرط حلقه

تا زمانیکه شرط برقرار باشد دستورات اجرا می‌شود.

شرط حلقه در انتهای حلقه کنترل می‌شود.

حلقه حداقل یکبار اجرا خواهد شد.

17 مجموع اعداد صحیح و مثبت از 1 تا 100 را محاسبه کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

2- $s \leftarrow 0$

3- $I \leftarrow 1$

4- $s \leftarrow s + I$

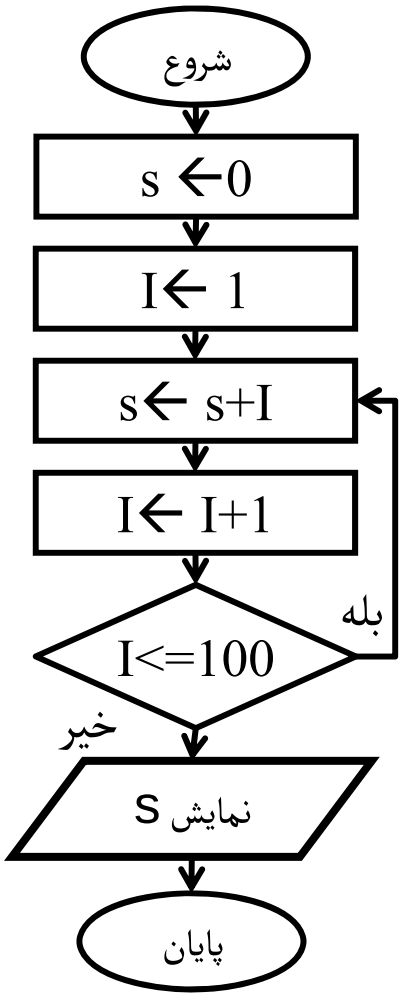
5- $I \leftarrow I + 1$

6- اگر $I<=100$ آنگاه برو به مرحله 4

7- مقدار S را نمایش بده

8- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 int I, s;
 s=0;
 I=1;
 do
 {
     s+=I;
     I++;
 }
 while(I<=100)
 cout<<s;
 getch();
 return 0;
}
```

2013

M. Damrudi

18 ضرایب معادله درجه اول $ax+b=c$ را با شرط $a\neq0$ خوانده ، ریشه آن را محاسبه کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

2- a ، b ، c را دریافت کن

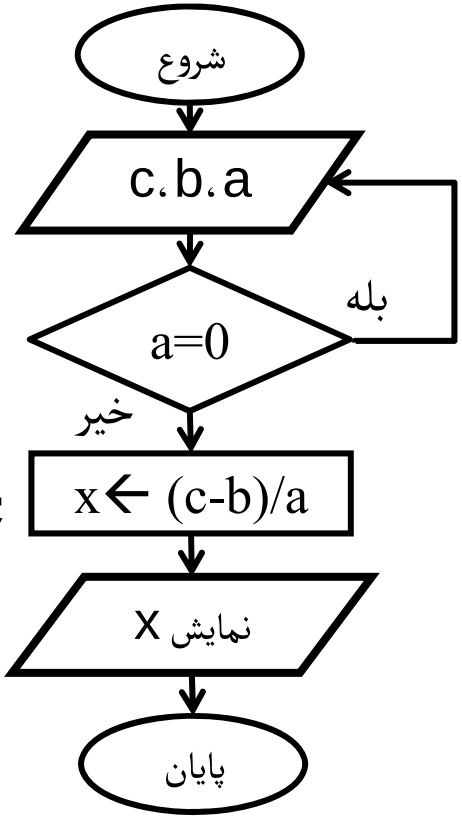
3- اگر $a=0$ آنگاه برو به مرحله 2

4- $x \leftarrow (c-b)/a$

5- مقدار x را نمایش بده

6- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 float a,b,c,x;
 cin>>a>>b>>c;
 while(a==0)
 {
  cout<<"Enter nonzero a";
  cin>>a;
 }
 x=(c-b)/a;
 cout<<x;
 getch();
 return 0;
}
```

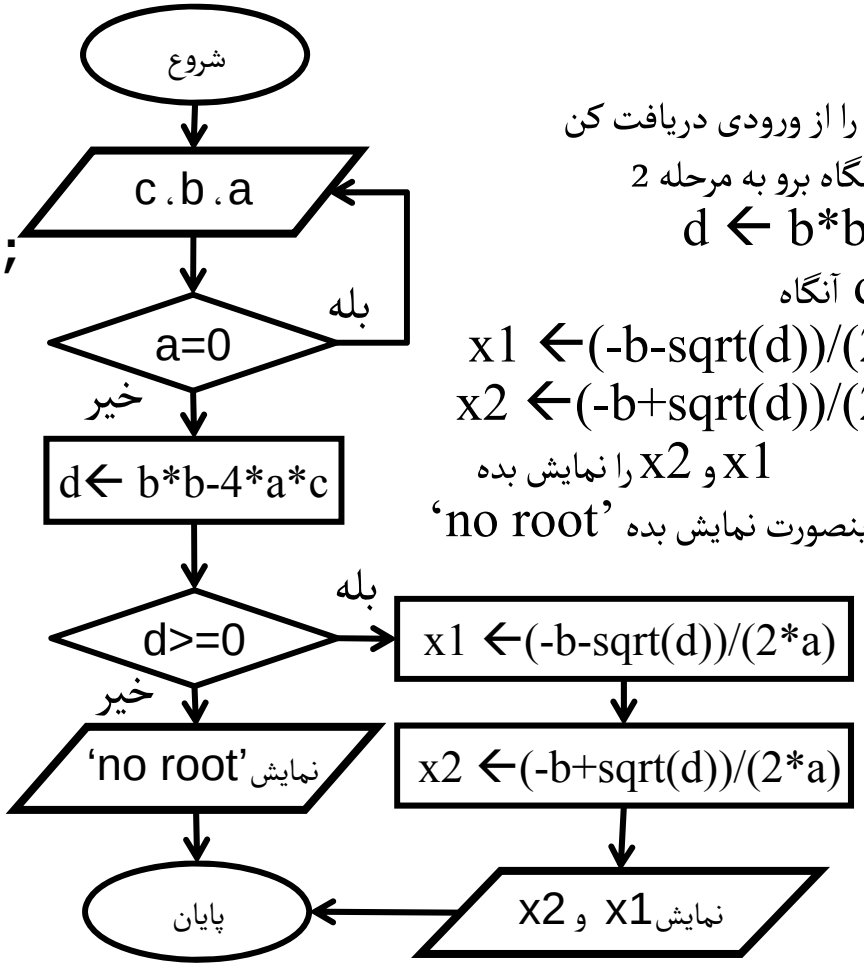
19 ضرایب معادله درجه دوم $ax^2+bx+c=0$ را با شرط $a\neq0$ خوانده ، ریشه های حقیقی آن را محاسبه کرده و نتیجه را نمایش دهد. در غیراینصورت پیام مناسب را نمایش دهد.

```
#include<iostream.h>
#include<conio.h>
int main(){
 double a,b,c,x1,x2;
 cin>>a>>b>>c;
 while(a==0)
 {
  cout<<"Enter nonzero a";
  cin>>a;
 }
 d=b*b-4*a*c;
 if (d<0)
   cout<<"No roots";
 else
 {
   x1=(-b+sqrt(d))/(2*a);
   x2=(-b-sqrt(d))/(2*a);
   cout<<x1<<x2;
 }
 getch();
 return 0;
}
```

الگوریتم

1- شروع
2- a و b و c را از ورودی دریافت کن
3- اگر a=0 آنگاه برو به مرحله 2
4- d ← b*b-4*a*c
5- اگر d>=0 آنگاه
x1 ←(-b-sqrt(d))/(2*a)
x2 ←(-b+sqrt(d))/(2*a)
x1 و x2 را نمایش بده
درغیر اینصورت نمایش بده 'no root'
6- پایان

فلوچارت



20 یک عدد صحیح مثبت n را خوانده ، فاکتوریل را محاسبه کرده و نتیجه را نمایش دهد.

الگوریتم

1- شروع

2- n را دریافت کن

3- I← 1 ، f ← 1

4- f ← f * I

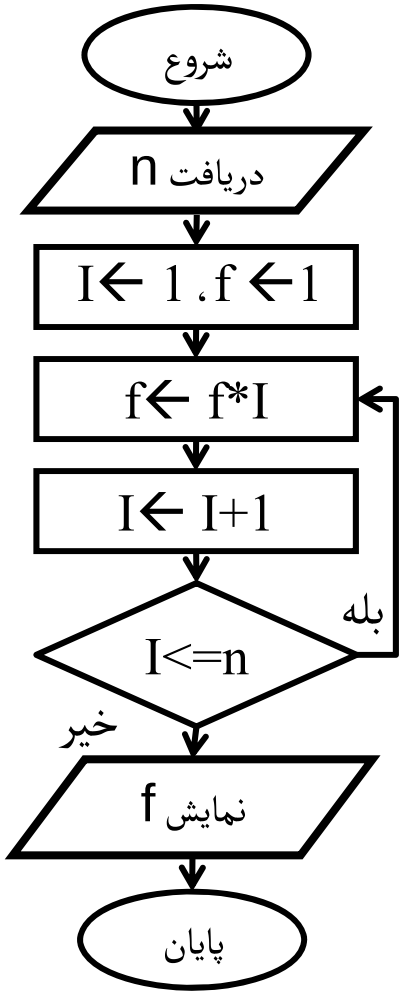
5- I ← I + 1

6- اگر I<=n آنگاه برو به مرحله 4

7- مقدار f را نمایش بده

8- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned n,i;
 long f=1;
 cin>>n;
 for(i=1; i<=n; i++)
    f=f*I;
 cout<<"Factorial is:"<<f;
 getch();
 return 0;
}
```

21 بزرگترین عدد دو رقمی صحیح و مثبت را که بر 16 بخش پذبر است را نمایش دهد.

الگوریتم

1- شروع

2- $n \leftarrow 100$

3- $n \leftarrow n-1$

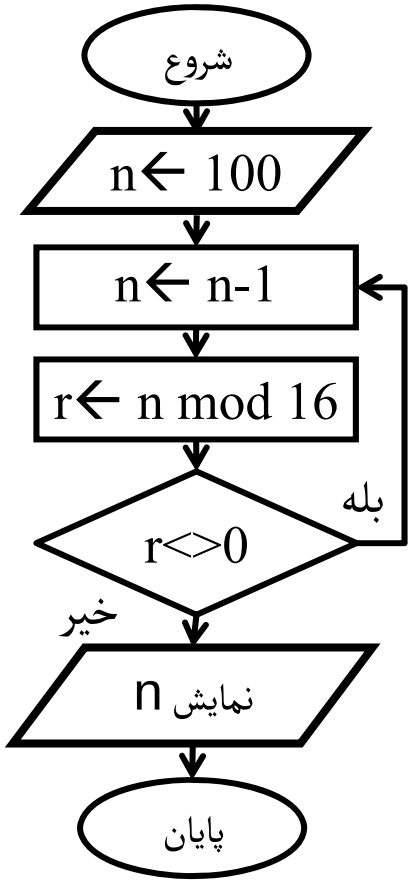
4- $r \leftarrow n \bmod 16$

5- اگر $r <> 0$ آنگاه برو به مرحله 3

6- مقدار n را نمایش بده

8- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned n=100;
 do
  {
    n--;
    r=n%16;
  }
 while(r!=0);
 cout<<n;
 getch();
 return 0;
}
```

22 عدد طبیعی n را دریافت کرده ، مجموع سری را محاسبه کرده و نمایش دهد.

$$S = \frac{1}{2} + \frac{1}{4} + ... + \frac{1}{2n}$$

الگوریتم

1- شروع

2- n را دریافت کن

3- $s \leftarrow 0$

4- $I \leftarrow 2$

5- $s \leftarrow s + 1/I$

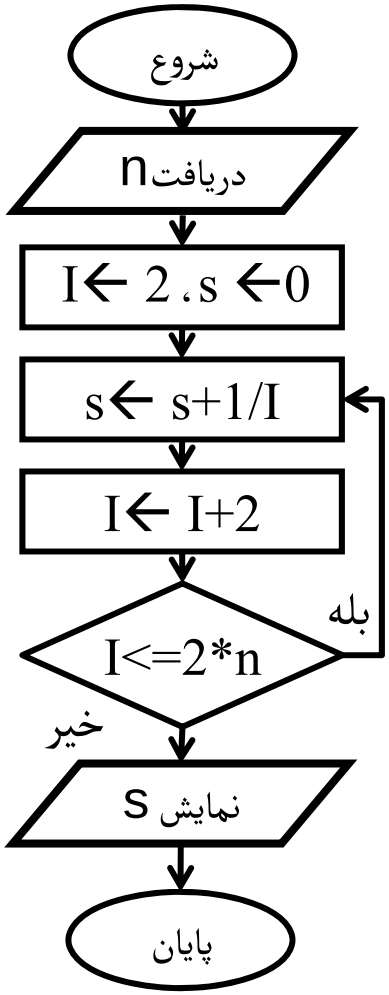
6- $I \leftarrow I + 2$

7- اگر $I<=2*n$ آنگاه برو به مرحله 5

8- مقدار s را نمایش بده

9- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned n,i;
 float s=0;
 cin>>n;
 for(i=2; i<=2*n; i+=2)
    s+=(float)1/i;
 cout<<s;
 getch();
 return 0;
}
```

23 مجموع اعداد 4 رقمی مضرب 3 را محاسبه کرده و نتیجه را نمایش دهد. $S = 1002 + 1005 + 1008 + ... + 9999$

الگوریتم

1- شروع

2- $s \leftarrow 0$

3- $I \leftarrow 1002$

4- $s \leftarrow s + I$

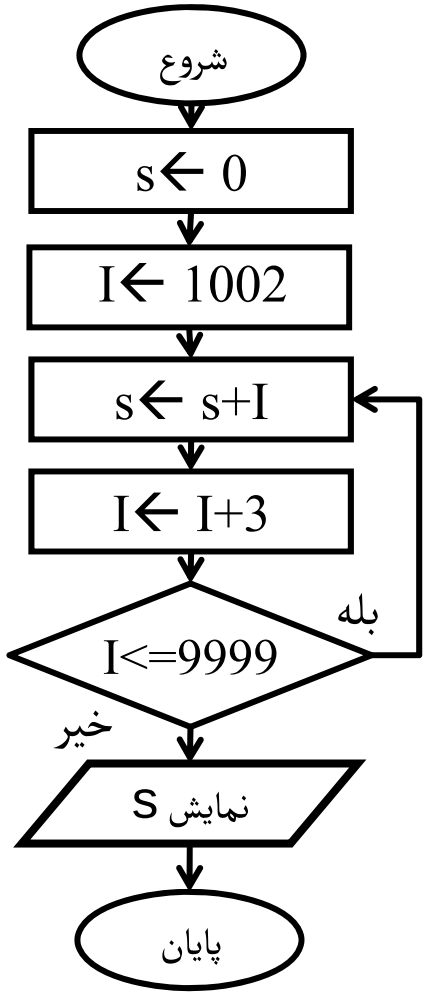
5- $I \leftarrow I+3$

6- اگر $I<=9999$ آنگاه برو به مرحله 4

7- مقدار S را نمایش بده

8- پایان

فلوچارت



برنامه

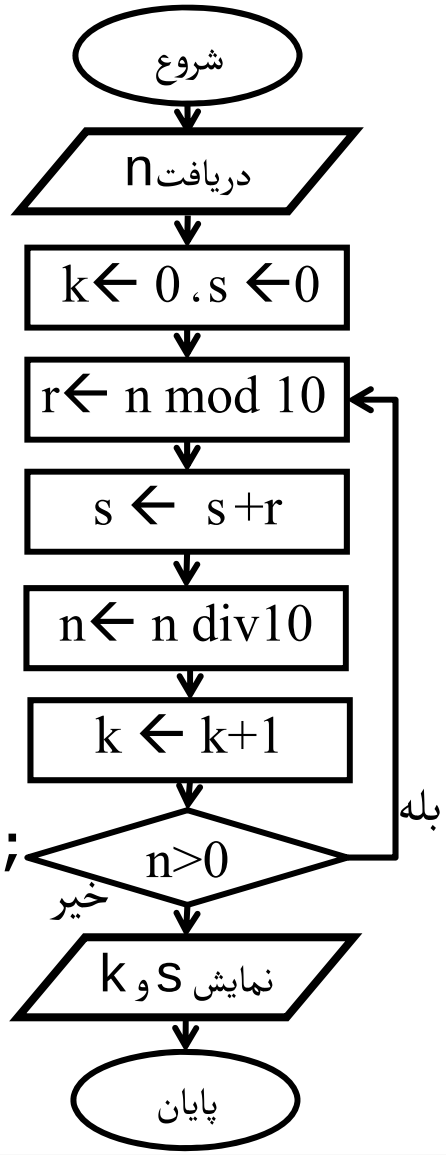
```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned i;
 unsigned long s=0;
 for(i=1002;i<=9999;i+=3)
    s+=i;
 cout<<s;
 getch();
 return 0;
}
```

24 یک عدد صحیح و مثبت را دریافت کرده ، تعداد ارقام و مجموع ارقام آنرا محاسبه کرده و نتیجه را نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned int n,r;
 unsigned int s=0, k=0;
 cin>>n;
 while (n != 0)
 {
    r = n % 10;
    s+=r;
    n=int(n/10);
    k++;
 }
 cout<<"S:"<<s<<"\nk:"<<k;
 getch();
 return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- n را دریافت کن

3- $k \leftarrow 0$

4- $s \leftarrow 0$

5- $r \leftarrow n \bmod 10$

6- $s \leftarrow s+r$

7- $n \leftarrow n \text{ div } 10$

8- $k \leftarrow k+1$

8- اگر $n>0$ آنگاه برو به مرحله 5

7- مقدار s و k را نمایش بده

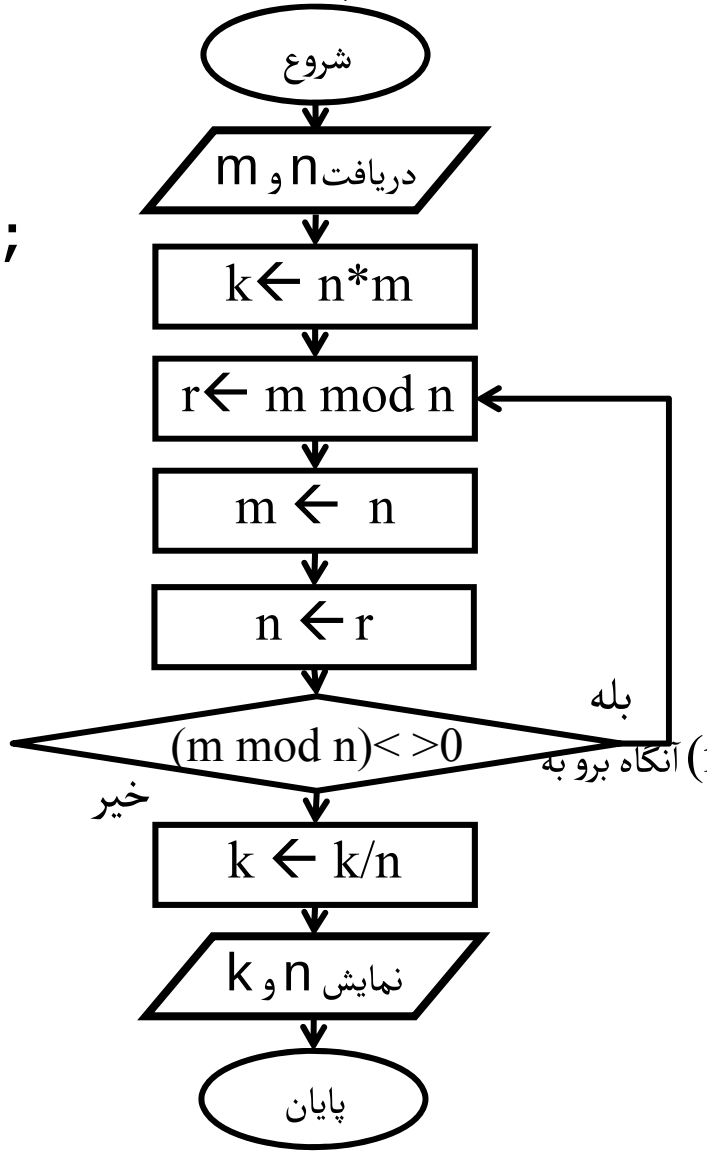
8- پایان

25 دو عدد صحیح و مثبت را دریافت کرده ب.م.م و ک.م.م را محاسبه کرده و نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned int m,n,k,r;
 cin>>n>>m;
 k=n*m;
 r= m % n;
 while(r != 0)
 {
   m=n;
   n=r;
   r=m % n;
 }
 k=k/n;
 cout<<k<<"\n"<<n;
 getch();
 return 0;
} 2013
```

فلوچارت



الگوریتم

1- شروع

2- n و m را دریافت کن

3- k← n*m

4- r ← m mod n

5- m ← n

6- n ← r

7- اگر (m mod n)< >0 آنگاه برو به مرحله 4

8- k ← k/n

9- مقدار k و n را نمایش بده

10- پایان

M. Damrudi

26 محاسبه کرده و نمایش دهد.

الگوریتم فلوچارت برنامه

27 با استفاده از سری زیر مقدار p را محاسبه کرده و نمایش دهد.

$$P = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} + \ldots\right)$$

الگوریتم

1- شروع

2- n را دریافت کن

3- $s \leftarrow 0$

4- $i \leftarrow 1$

5- $j \leftarrow 1$

6- $k \leftarrow 1$

7- اگر $j<=n$ آنگاه برو به مرحله 8
درغیراینصورت برو به مرحله 12

8- $s \leftarrow s + k/i$

9- $i \leftarrow i+2$

10- $k \leftarrow -k$

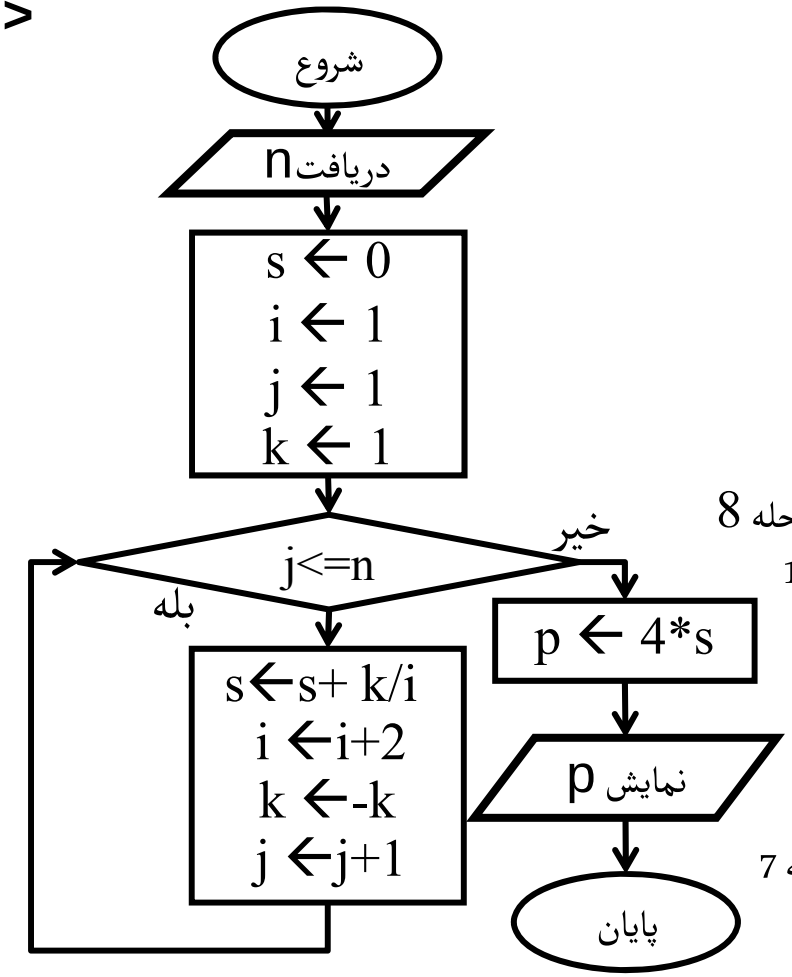
11- $j \leftarrow j+1$ ،برو به مرحله 7

12- $p \leftarrow 4*s$

13- مقدار p را نمایش بده

14- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 int n,i=1,k=1,j=1;
 float s=0, p;
 cin>>n;
 while(j<=n)
  {
    s+=(float)k/i;
    i+=2;
    k=-k;
    j++;
  }
 p=4*s;
 cout<<p;
 getch();
 return 0;
}
```

2013

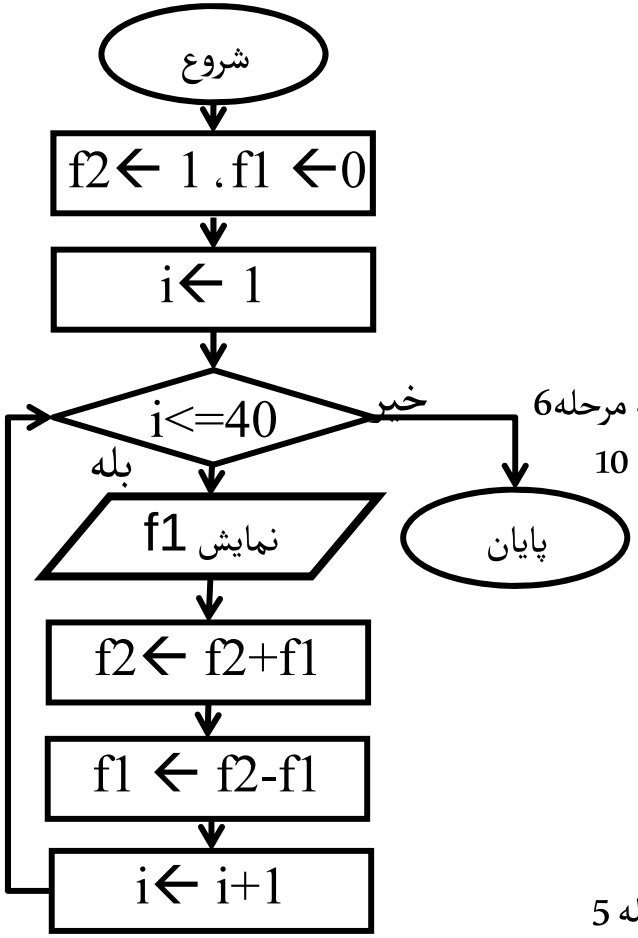
M. Damrudi

28 چهل جمله اول سری فیبوناچی را نمایش دهد. 0 1 1 2 3 5 8 13

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 int i;
 long f1=0,f2=1;
for(i=1;i<=40;i++)
 {
    cout<<f1<<"\t";;
    f2+=f1;
    f1=f2-f1;
 }
 getch();
 return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- f1 ← 0

3- f2 ← 1

4- i ← 1

5- اگر i<=40 آنگاه برو به مرحله6
درغیر اینصورت برو به مرحله 10

6- مقدار f1 را نمایش بده

7- f2← f2+f1

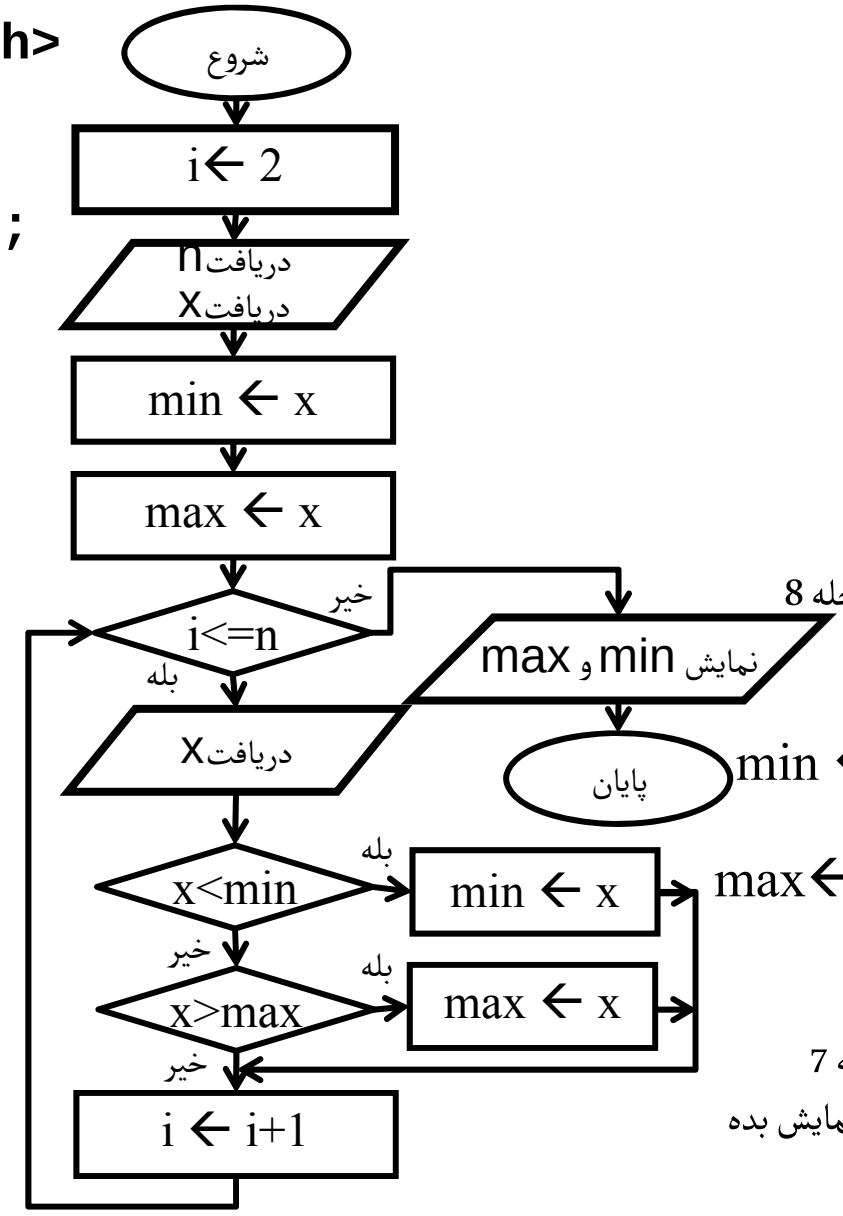
8- f1 ← f2-f1

9- i← i+1 و برو به مرحله 5

10- پایان

29 n عدد را دریافت کرده کوچکترین و بزرگترین عدد را نمایش دهد.

```
#include<iostream.h>
#include<conio.h>
int main(){
 int i,n,x,min,max;
 cin>>n>>x;
 min=x;
 max=x;
 for(i=2;i<=n;i++)
 {
   cin>>x;
   if(x<min)
     min=x;
   else if(x>max)
            max=x;
 }
 cout<<min<<max;
 getch();
 return 0;
}
```



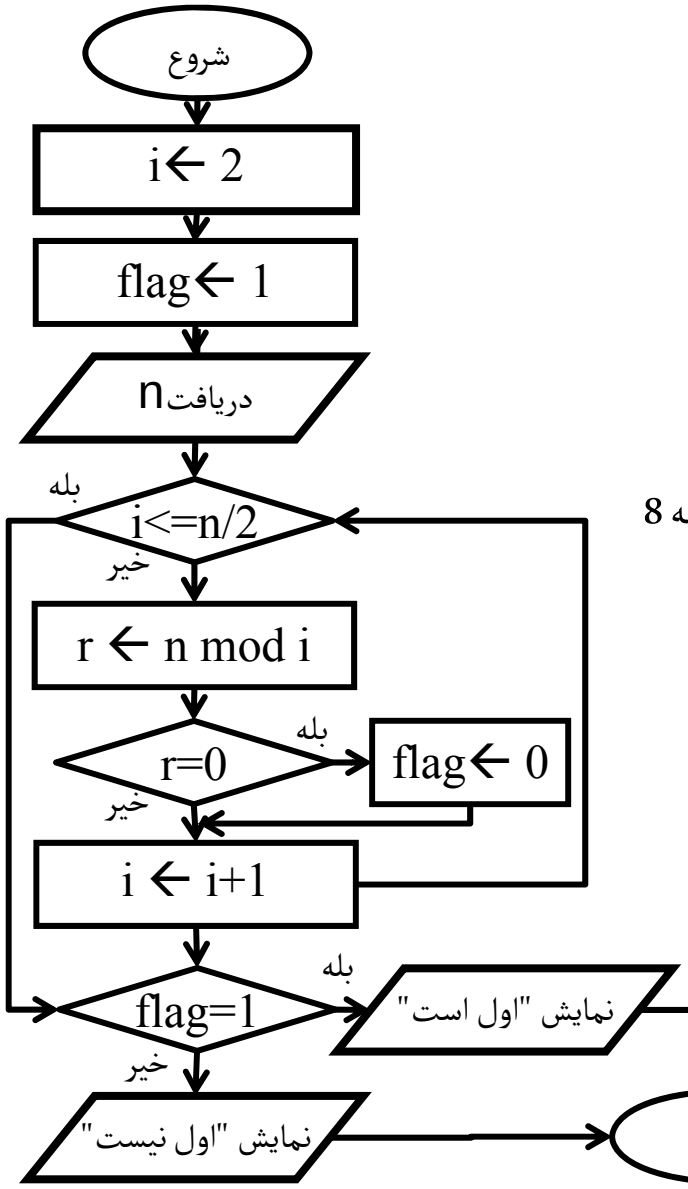
الگوریتم

1- شروع
2- $i \leftarrow 2$
3- n را دریافت کن
4- x را دریافت کن (اولین x)
5- $min \leftarrow x$
6- $max \leftarrow x$
7- اگر $i<=n$ آنگاه برو به مرحله 8 درغیراینصورت برو به مرحله 12
8- x را دریافت کن
9- اگر $x<min$ آنگاه $min \leftarrow x$ و برو به مرحله 11
10- اگر $x>max$ آنگاه $max \leftarrow x$ و برو به مرحله 11
11- $i \leftarrow i+1$ و برو به مرحله 7
12- مقدار min و max را نمایش بده
13- پایان

30 عدد طبیعی n را دریافت کرده مشخص کند که آیا عدد اول است یا خیر.

```
#include<iostream.h>
#include<conio.h>
int main(){
 int i,n,r,flag=1;
 cin>>n;
 for(i=2;i<=n/2;i++)
 {
   r=n%i;
   if(r==0)
     flag=0;
 }

 if (flag==1)
    cout<<"prime";
 else
    cout<<"not prime";
 getch();
 return 0;
}
```



الگوریتم

1- شروع

2- i ← 2 ، flag=1

3- n را دریافت کن

4- اگر i<=n/2 آنگاه برو به مرحله 8

5- r ← n mod I

6- اگر r=0 آنگاه flag ← 0

7- i← i+1 و برو به مرحله 4

8- اگر flag=1 نمایش بده "اول است" درغیراینصورت نمایش بده "اول نیست"

9- پایان

31 یک عدد طبیعی را دریافت کرده مقلوب آن را نمایش دهد.

الگوریتم

1- شروع

2- n را دریافت کن

3- m ← 0

4- b← n mod 10

5- m ← m * 10 +b

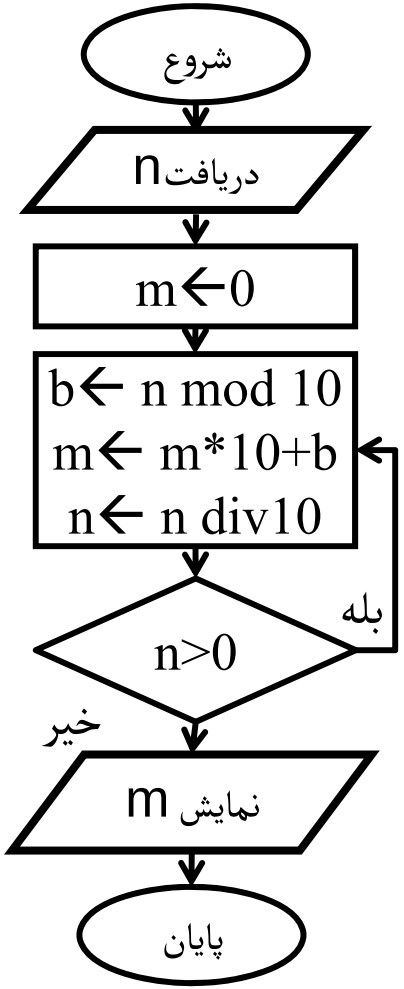
6- n← n div10

7- اگر n>0 آنگاه برو به مرحله 4

8- مقدار m را نمایش بده

9- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 int n,m,b;
 cin>>n;
 m=0;
 while(n>0)
  {
    b= n%10;
    m=m*10+b;
    n=int(n/10);
 }
 cout<<m;
 getch();
 return 0;
}
```

2013

M. Damrudi

32 را نمایش دهد.

الگوریتم فلوچارت برنامه

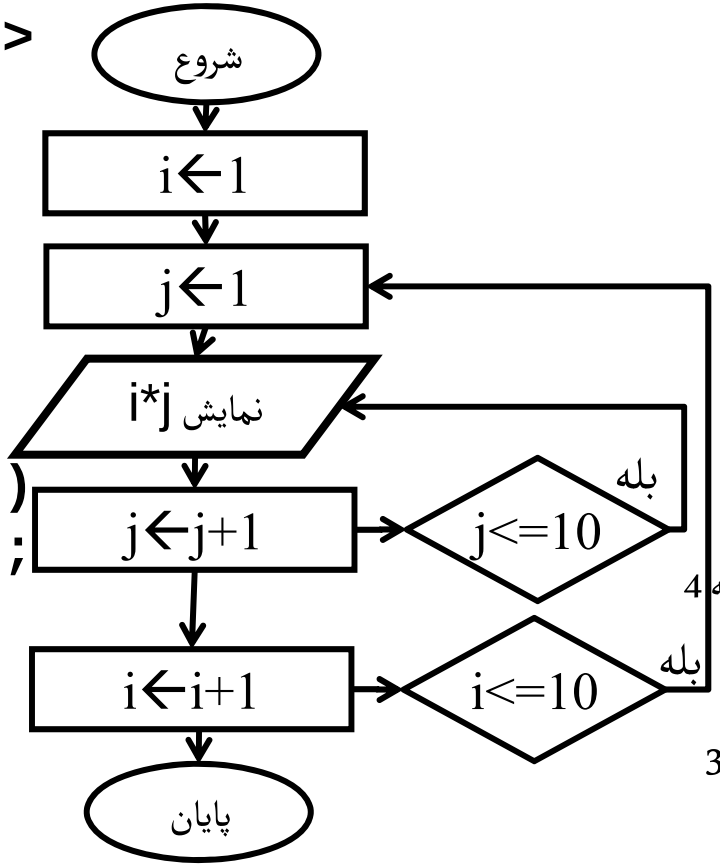
33 جدول ضرب 10*10 را نمایش دهد.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 int i,j;

 for(i=1;i<=10;i++)
 {
  for(j=1;j<=10;j++)
    cout<<i*j<<"\t";
  cout<<"\n";
 }
getch();
 return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- i ← 1

3- j ← 1

4- نمایش بده i*j

5- j← j+1

6- اگر j<=10 آنگاه برو به مرحله 4

7- i← i+1

8-اگر i<=10 آنگاه برو به مرحله 3

9- پایان

34 دو عدد صحیح مثبت را از ورودي دریافت کرده و حاصلضرب آن دو را بدون استفاده از عملگر ضرب محاسبه کرده و نمایش دهد.

الگوریتم

1- شروع

2- a و b را دریافت کن

3- I ← 1 ، s ← 0

4- s ← s + a

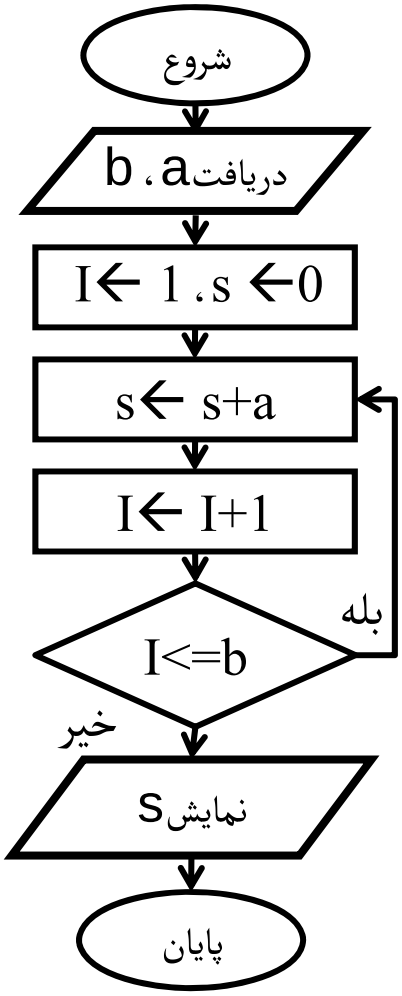
5- I ← I + 1

6- اگر I<=b آنگاه برو به مرحله 4

7- مقدار S را نمایش بده

8- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned a,b,i,s=0;
 cin>>a>>b;
 for(i=1;i<=b;i++)
    s += a;
 cout<<s;
 getch();
 return 0;
}
```

35 دو عدد دریافت کرده ، خارج قسمت و باقیمانده صحیح تقسیم را بدون استفاده از عمل تقسیم نمایش دهد.

الگوریتم

1- شروع

2- b و a را دریافت کن

3- $q \leftarrow 0$

4- $a \leftarrow a - b$

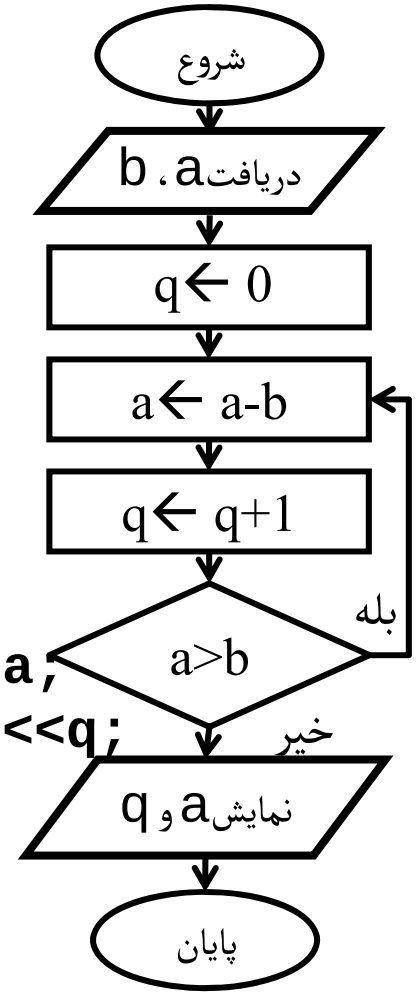
5- $q \leftarrow q + 1$

6- اگر a>b آنگاه برو به مرحله 4

7- a و q را نمایش بده

8- پایان

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned a,b,q=0;
 cin>>a>>b;
 while(a>=b)
  {
    a-=b;
    q++;
  }
 cout<<"remainder:"<<a;
 cout<<"\t quotient:"<<q;
 getch();
 return 0;
}
```

36 به ازاء مقادیر مختلف x از 4- تا 5 و گام افزایشی 0.5 مقادیر y را از $y=3x^2-6x+7$ محاسبه کرده و مقادیر x و y را نمایش دهد.

الگوریتم

1- شروع

2- $x \leftarrow -4$

3- $y \leftarrow 3*x*x-6*x+7$

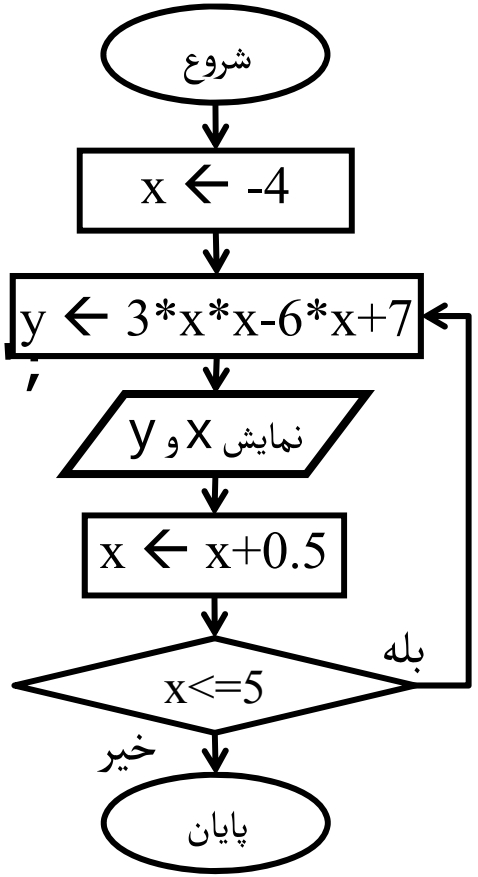
4- x و y را نمایش بده

5- $x \leftarrow x+0.5$

اگر $x<=5$ آنگاه برو به مرحله 3

8- پایان

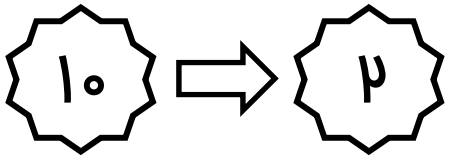
فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 float x,y;
 cout<<"x\t"<<"y\n";
 cout<<"---\t"<<"---\n";
 for(x=-4;x<=5;x+=0.5)
 {
   y=3*x*x-6*x+7;
   cout<<x<<"\t"<<y;
   cout<<"\n";
 } getch();
 return 0;
}
```

مبنا

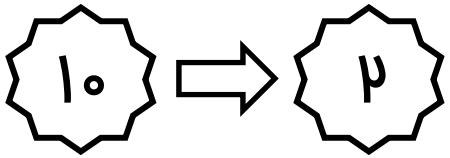


٣٧	٢
٢	١٨
١٧	١٨
١٦	٠
١	

١٨	٢
٩	٢
٨	٢
١	٢
	١

٤	٢
٤	٢
٠	٢
	٠

$(٣٧)_{١٠} = (١٠٠١٠١)_{٢}$

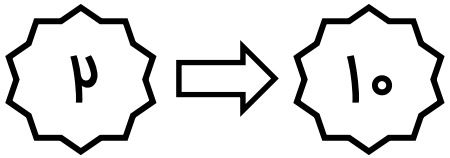


مبنا

$(٣٧)_{١٠} = (?)_{٢}$

64	32	16	8	4	2	1
0	1	0	0	1	0	1

٣٧= ٣٢ + ٤ + ١



مبنا

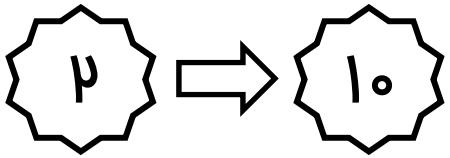
۵ ۴ ۳ ۲ ۱ ۰

$(۱\ ۰\ ۰\ ۱\ ۰\ ۱)_{۲} = (۳۷)_{۱۰}$

$1*2^5+0*2^4+0*2^3+1*2^2+0*2^1+1*2^0=32+4+1$

0 0 0

مبنا



۳۲ ۱۶ ۸ ۴ ۲ ۱

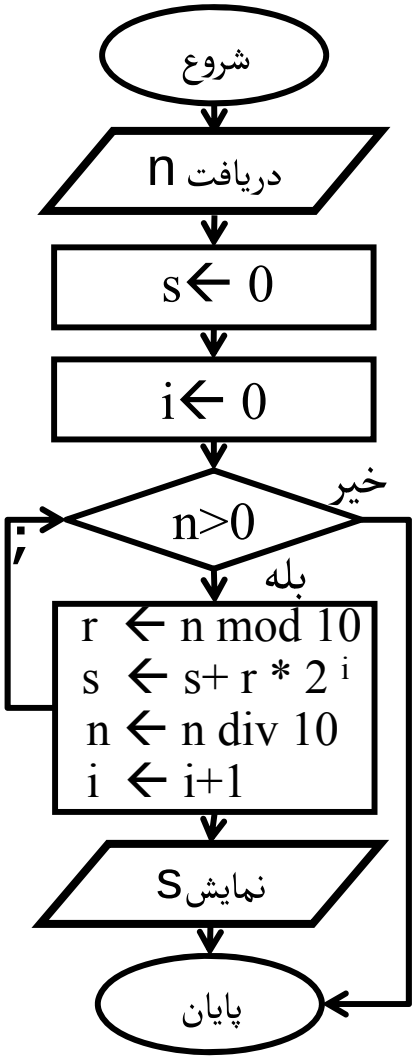
$(۱\ ۰\ ۰\ ۱\ ۰\ ۱)_{۲} = (۳۷)_{۱۰}$

۱+۴+۳۲=۳۷

37 عدد صحیح مثبت n رادر مبنای 2 دریافت کرده ، معادل آنرا در مبنای 10 محاسبه کرده و نمایش دهد.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
int main(){
 long double s=0,r;
 int n,i=0;
 cin>>n;
 while(n>0)
 {
   r=n%10;
   s+=r*pow((long double)2,i);
   n=n/10;
   i++;
 }
 cout<<"\n"<<s;;
 getch();
 return 0;
}
```

فلوچارت



الگوریتم

1- شروع

2- n را دریافت کن

3- $s \leftarrow 0$

4- $i \leftarrow 0$

5- اگر $n>0$ آنگاه برو به مرحله 6
درغیراینصورت برو به مرحله 10

6- $r \leftarrow n \bmod 10$

7- $s \leftarrow s + r * 2^{i}$

8- $n \leftarrow n \operatorname{div} 10$

9- $i \leftarrow i+1$ و برو به مرحله 5

10- S را نمایش بده

11- پایان

2013

M. Damrudi

38 را نمایش دهد.

الگوریتم فلوچارت برنامه

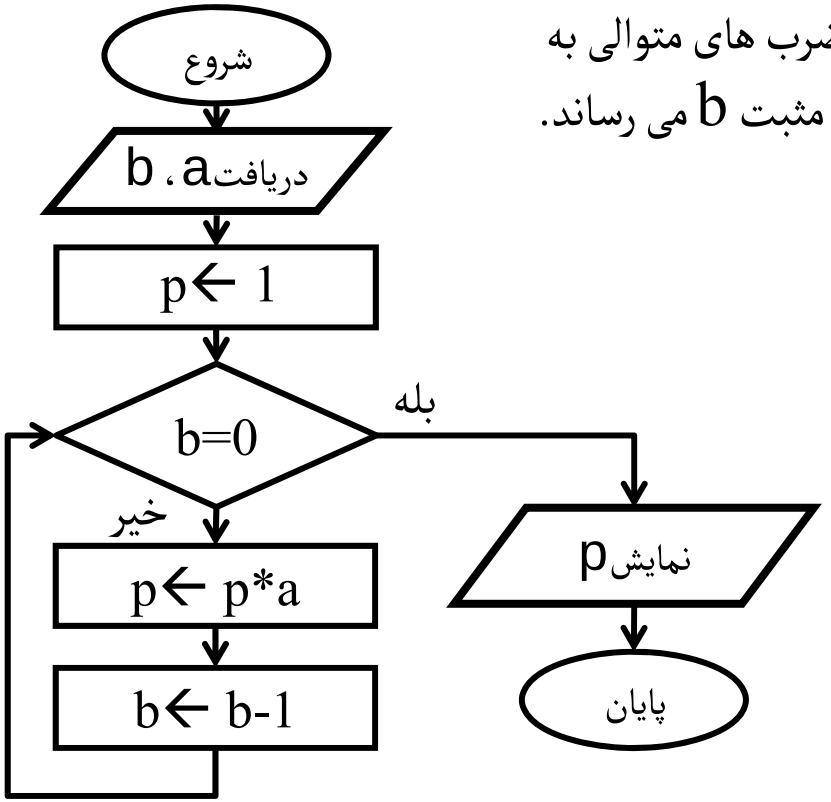
39 پس از تکمیل جدول ردیابی برای فلوچارت زیر و اعداد a=3 و b=4 عملکرد آنرا توضیح دهید.

عملکرد

عدد a را به روش ضرب های متوالی به توان عدد صحیح و مثبت b می رساند.

a^b

فلوچارت



a	b	p
3	4	1
	3	3
	2	9
	1	27
	0	81

برنامه

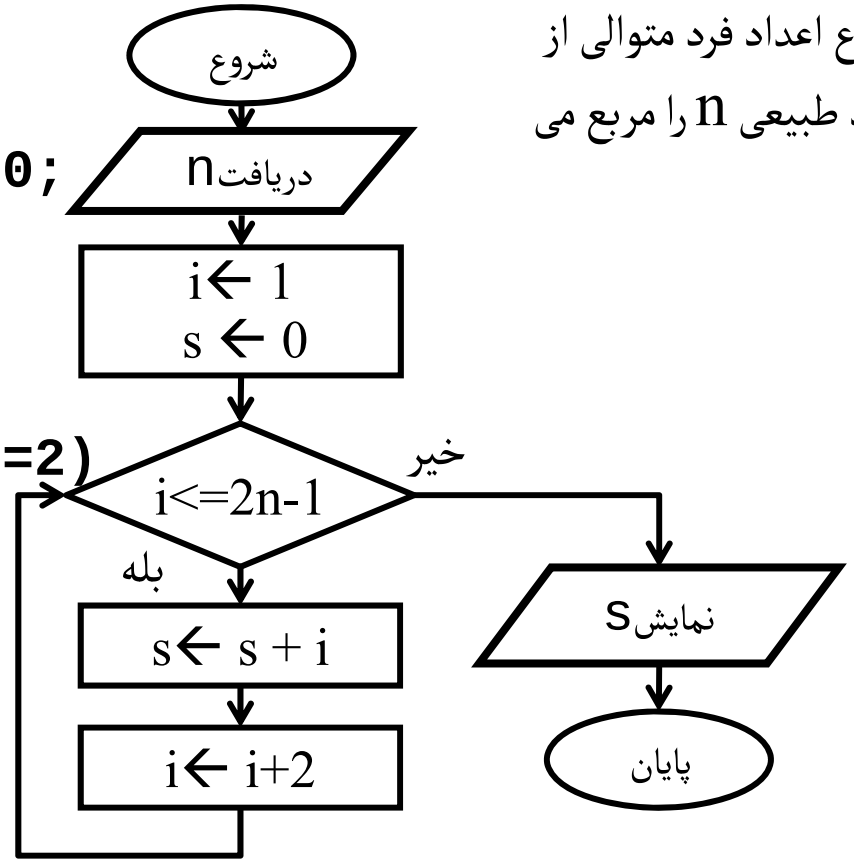
```
#include<iostream.h>
#include<conio.h>
int main(){
 long p=1;
 int a,b;
 cin>>a>>b;
 for(;b!=0;b--)
    p *= a;
 cout<<p;
 getch();
 return 0;
}
```

40 پس از تکمیل جدول ردیابی برای فلوچارت زیر و عدد n=4 عملکرد آنرا توضیح دهید.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned int n,i,s=0;
 int a,b;
 cin>>n;
 for(i=1;i<=2*n-1;i+=2)
    s += i;
 cout<<s;
 getch();
 return 0;
}
```

فلوچارت



عملکرد

با استفاده از مجموع اعداد فرد متوالی از یک تا 2n-1 عدد طبیعی n را مربع می کند.

n	i	s
4	1	0
	3	1
	5	4
	7	9
	9	16

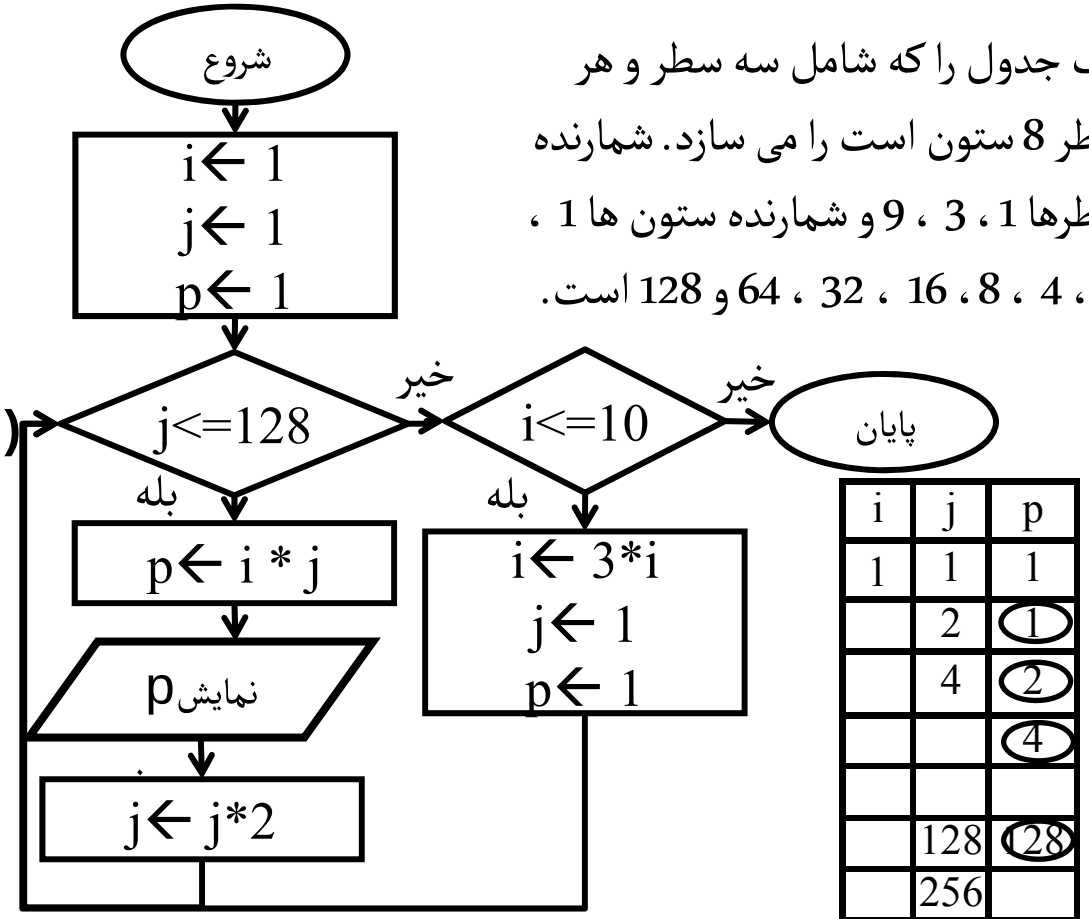
41 پس از تکمیل جدول ردیابی برای فلوچارت زیر عملکرد آنرا توضیح دهید.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned int j,i,p;
 int a,b;
 for(i=1;i<=10;i*=3)
  {
  for(j=1;j<=128;j*=2)
   {
   p = i*j;
   cout<<p<<"\t";
   }
  cout<<"\n";
  }
 getch();
 return 0;
}
```

2013

فلوچارت



i	j	p
1	1	1
	2	1
	4	2
		4
	128	128
	256	
3	1	1

عملکرد

یک جدول را که شامل سه سطر و هر سطر 8 ستون است را می سازد. شمارنده سطرها 1 ، 3 ، 9 و شمارنده ستون ها 1 ، 2 ، 4 ، 8 ، 16 ، 32 ، 64 و 128 است.

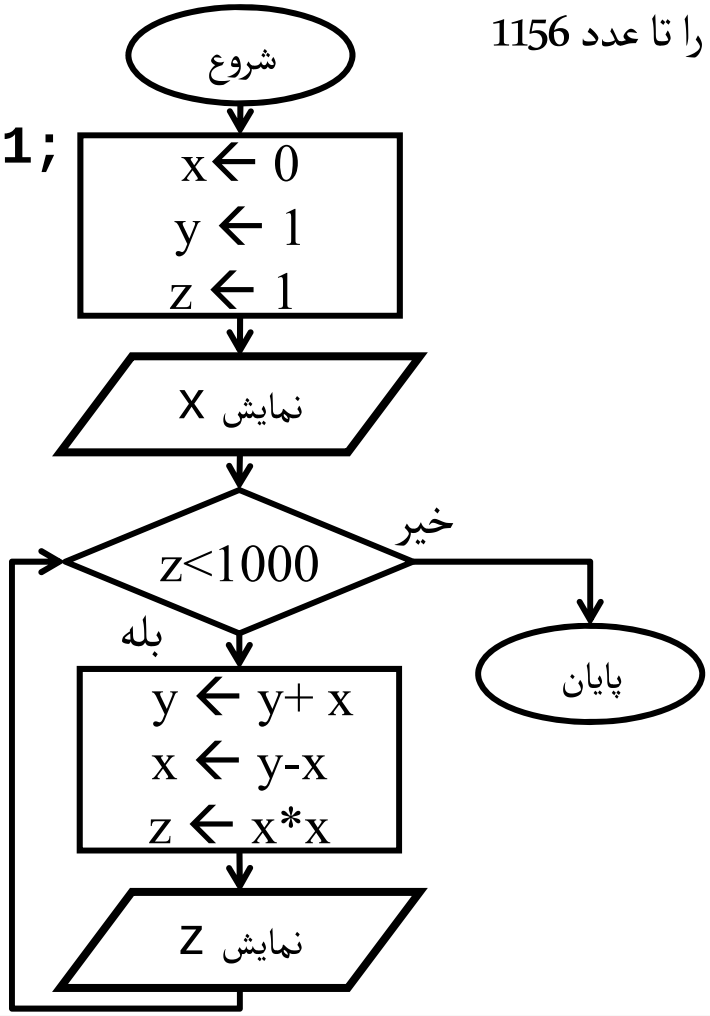
M. Damrudi

42 پس از تکمیل جدول ردیابی برای فلوچارت زیر عملکرد آنرا توضیح دهید.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned int x=0,y=1;
 unsigned int z=1;
 cout<<x<<"\t";;
 while(z<1000)
 {
    y += x;
    x = y-x;
    z = x*x;
    cout<<z<<"\t";;
 }
 getch();
 return 0;
}
```

فلوچارت



عملکرد

مربع جملات سری فیبوناچی را تا عدد 1156 نمایش می دهد.

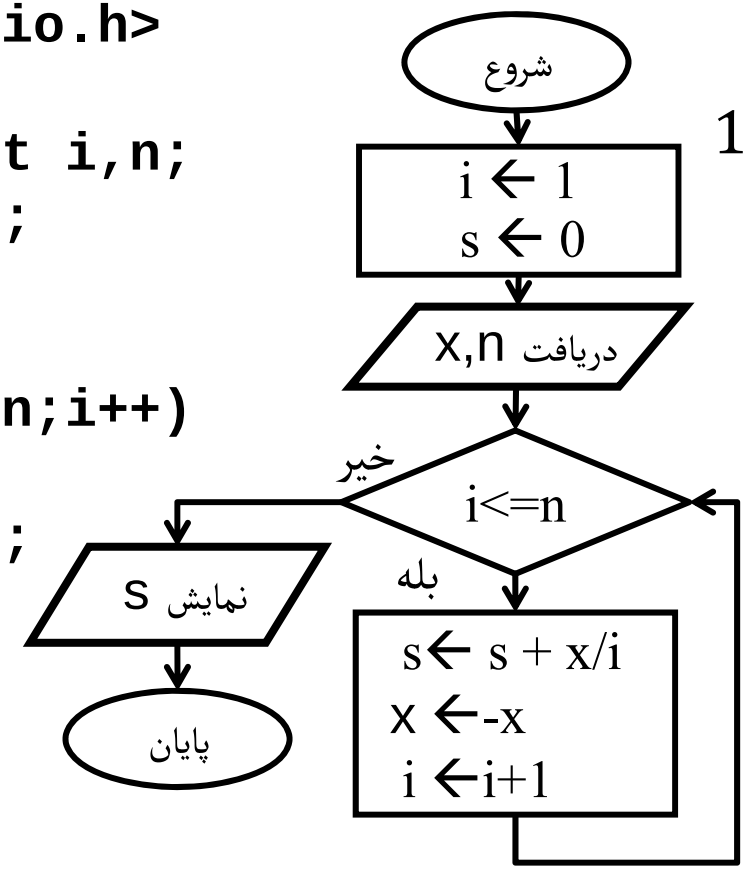
x	y	z
0	1	1
1	1	1
1	2	4
2	3	9
3	5	25
5	8	64
8	13	169
13	21	441
21	34	1156

43 پس از تکمیل جدول ردیابی برای فلوچارت و اعداد x=1 و n=5 زیر عملکرد آنرا توضیح دهید.

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned int i,n;
 float s=0,x;
 cin>>x;
 cin>>n;
 for(i=1;i<=n;i++)
 {
    s += x/i;
    x = -x;
 }
 cout<<s;
 getch();
 return 0;
}
```

فلوچارت



عملکرد

فلوچارت n جمله از سری زیر را محاسبه و چاپ می کند. $1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\cdots$

n	x	i	s
5	1	1	0
	-1	2	$0+\frac{1}{1}$
	1	3	$1-\frac{1}{2}$
	-1	4	$1-\frac{1}{2}+\frac{1}{3}$
	1	5	$1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}$
	-1	6	$1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\frac{1}{5}$

2013

44 پس از تکمیل جدول ردیابی برای فلوچارت زیر عملکرد آنرا توضیح دهید.

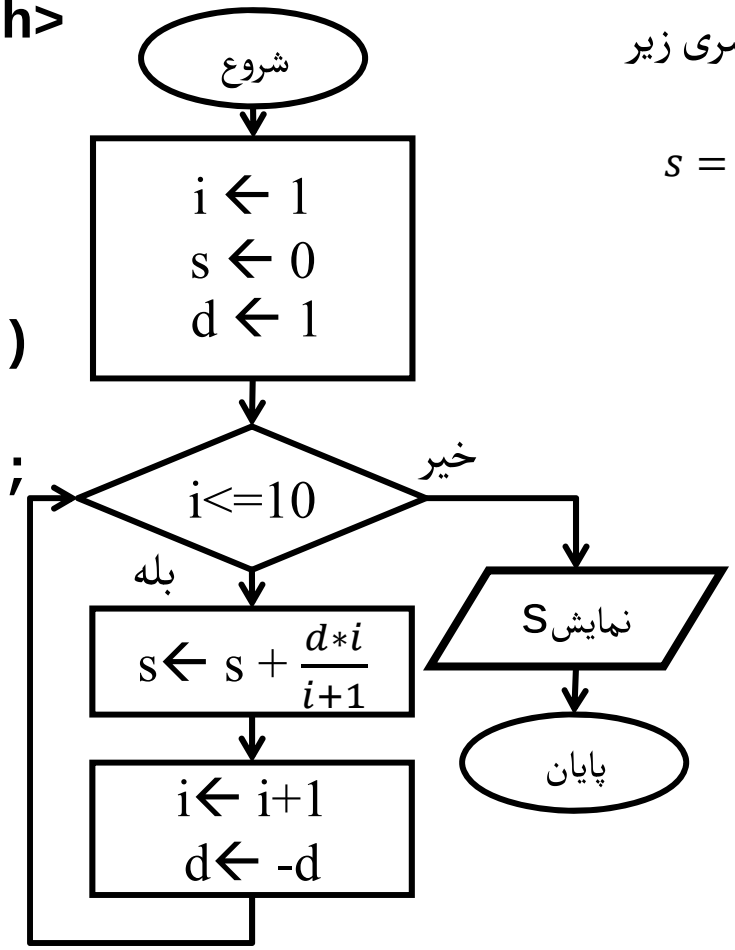
عملکرد

محاسبه و مجموع 10 جمله از سری زیر

$$s = \frac{1}{2} - \frac{2}{3} + \frac{3}{4} - \frac{4}{5} + \cdots - \frac{10}{11}$$

i	d	s
1	1	0
2	-1	$0+\frac{1\times1}{1+1}$
3	1	$\frac{1}{2}-\frac{2}{3}$
4	-1	$\frac{1}{2}-\frac{2}{3}+\frac{3}{4}$
5	1	$\frac{1}{2}-\frac{2}{3}+\frac{3}{4}-\frac{4}{5}$

فلوچارت



برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 unsigned int i;
 float s=0, d=1;
 for(i=1;i<=10;i++)
 {
    s += d*i/(i+1);
    d=-d;
 }
 cout<<s;
 getch();
 return 0;
}
```

2013

نگاهی دقیق‌تر به نوع داده‌ها

با استفاده از توصیف‌کننده‌های نوع (type modifiers) می‌توان نوع داده‌های char, int, float, double را تعدیل کرد.

توصیف‌کننده‌های نوع عبارتند از :

- signed
- unsigned
- long
- short

توصیف کننده نوع پیش از نوع داده آورده می‌شود.

```
long int i;
```

signed (علامت‌دار)

signed را مي‌توان در مورد انواع int و char به‌كار برد.

استفاده از signed براي اعداد صحيح كار زايدي است زيرا به طور پيش فرض اعداد صحيح علامت‌دار هستند.

بسته به نوع پياده‌سازي يك char مي‌تواند علامت‌دار يا بدون علامت باشد. در صورتي كه بدون علامت باشد اعداد مثبت بين 0 تا 255 را نگاه مي‌دارد. درصورتي كه به عنوان signed باشد مي‌تواند اعدادي در دامنه 128- تا 127 را نگاه مي‌دارد.

```
signed char x;
```

unsigned (بدون علامت)

unsigned را مي‌توان در مورد نوع داده‌هاي int و char به‌كار برد. اين توصيف‌كننده را همراه توصيف‌كننده‌هاي short و long نيز مي‌توان مورد استفاده قرار داد.

long

long را مي‌توان در مورد نوع داده‌هاي int و double به‌كار برد.

long بر روي int طول آن را بر حسب بيت دوبرابر مي‌كند. اگر در محيطي طول int 16 بيت باشد با استفاده از long طول آن 32 بيت مي‌شود.

long بر روي double طول double را نيز بر حسب بيت دوبرابر مي‌كند.

```
long int g;
```

short

short را مي‌توان تنها در مورد نوع داده int به‌كار برد. short بر روي int طول آن را بر حسب بيت نصف مي‌كند. اگر در محيطي طول int 32 بيت باشد با استفاده از short طول آن 16 بيت مي‌شود.

signed & unsigned

۱- تفاوت اعداد صحیح علامت‌دار در نحوه تفسیر بیت با مرحله بالاتر می‌باشد.
۲- اگر عدد صحیح بدون علامت باشد از تمامی بیت‌ها برای نگه‌داری مقادیر استفاده می‌شود.
۳- اگر عدد صحیح علامت‌دار باشد در آن صورت کامپایلر کدی تولید می‌کند که فرض می‌نماید از بیت با مرتبه بالاتر به عنوان علامت استفاده می‌شود. اگر ۰ باشد عدد مثبت و اگر ۱ باشد عدد منفی است.
۴- عموما اعدا منفی با روش two’s complement (مکمل دو) بیان می‌شوند.

ترکیبات ممکن از توصیف‌کننده‌ها

نوع	عرض	دامنه
char	۸	۱۲۸– تا ۱۲۷
unsigned char	۸	۰ تا ۲۵۵
signed char	۸	۱۲۸– تا ۱۲۷
int	۱۶	۳۲۷۶۸– تا ۳۲۷۶۷
unsigned int	۱۶	۰ تا ۶۵۵۳۵
signed int	۱۶	۳۲۷۶۸– تا ۳۲۷۶۷
short int	۱۶	مثل int
unsigned short	۱۶	مثل unsigned int
signed short int	۱۶	مثل short int

ترکیبات ممکن از توصیف‌کننده‌ها

نوع	عرض
long int	۳۲
unsigned long	۳۲
signed long int	۳۲
float	۳۲
double	۶۴
long double	۸۰

نکته

ممکن است در محیط کاری شما long و short اصلا تاثیرین داشته باشد. بستگی به کامپایلر دارد.

توابع ریاضی

math.h

عملیات ریاضی	تابع در زبان C
$\cos t$	double cos(double t)
e^t	double exp(double t)
x%y	double fmod(double x, double y)
$\log^t$	double log10(double t)
x^y	double pow(double x, double y)
$\sin t$	double sin(double t)
$\sqrt{t}$	double sqrt(double t)

آرایه‌ها و رشته‌ها

یک آرایه مجموعه‌ای از متغیرهای به‌هم مرتبط است که به وسیله یک نام مشترک مشخص می‌شود.

آرایه یک بعدی فهرستی از متغیرهایی از یک نوع واحد است که با استفاده از یک نام مشترک به همه آنها مراجعه می‌شود.

به هر متغیر یک آرایه یک عنصر آرایه گفته می‌شود.

حالت کلی آرایه‌ها به صورت زیر می‌باشد:

```
type var_name[size];
```

برای دسترسی به هر عنصر آرایه باید از شماره آن عنصر به عنوان index استفاده نمود. اندیس تمام آرایه‌ها از صفر شروع می‌شود و تا size-1 ادامه دارد.

آرایه‌ها و رشته‌ها

```
int d[20];
d[0]=100;
```

مقدار عنصر صفر آرایه را برابر با ۱۰۰ قرار می‌دهد.

برای مقداردهی آرایه‌ها نیاز به حلقه‌های for داریم. مقادیر عناصر را باید تک به تک قرار داد.

مشکلی که وجود دارد این است که بر روی ایندکس آرایه‌ها عمل بررسی دامنه (bound checking) صورت نمی‌پذیرد.

به عنوان برنامه‌نویس خود باید مراقب این مساله باشید.

مثال:

```
int t[20];
t[25]=14;
```

دریافت و ذخیره ۱۰ عدد در آرایه و نمایش اعداد از انتها به ابتدا

```
#include <iostream.h>
#include <conio.h>
int main()
{
int i;
int n[10];
for(i=0; i<10;i++){
     cin>>n[i];
}
for(i=9; i>=0;i--)
    cout<<n[i];
getch();
return 0;
}
```

برنامه‌ای بنویسید که ۳۰ عدد صحیح مثبت را دریافت کرده و max آنها را بیابید، به همراه کل اعداد نمایش دهید.

```
#include <iostream.h>
#include <conio.h>
int main(){
int i, max;
int n[30];
cin>>n[0];
max=n[0];
for(i=1; i<30;i++){
     cin>>n[i];
     if(n[i]>max) max=n[i];
}
for(i=0; i<30;i++)
    cout<<n[i];
cout<<max;
getch();
return 0;}
```

برنامه‌ای بنویسید که ۲۰ عدد را دریافت، در یک آرایه ذخیره نموده و میانگین آنها را نمایش دهد.

```
#include <iostream.h>
#include<conio.h>
int main()
{
int i,n[20],s=0;
for(i=0; i<20;i++){
    cin>>n[i];
    s=s+n[i];
    }
cout<<s/20;
getch();
return 0;
}
```

رشته‌ها (strings)

متداول‌ترین کاربرد آرایه‌های یک بعدی رشته‌ها می‌باشند.

هر رشته به صورت یک آرایه کاراکتری null-terminated ختم شده به تهی تعریف می‌شود.

آرایه‌ای که قرار است یک رشته را نگه دارد باید به کاراکتر تهی ختم شود به این معنی که باید یک بایت بزرگتر در نظر گرفته شود تا کاراکتر تهی در آن قرار گیرد. مقدار کاراکتر تهی صفر است.

```
char name[11];
```

حداکثر ۱۰ کاراکتر را نگه می‌دارد.

هر ثابت رشته‌ای نیز به یک کاراکتر تهی ختم می‌شود که کامپایلر به طور اتوماتیک آن را به انتهای رشته اضافه می‌کند.

برنامه‌ای بنویسید که یک رشته حداکثر ۲۰ کاراکتری بدون فاصله را دریافت نماید، تمام حروف C را با K جایگزین نماید

```
#include <iostream.h>
#include <conio.h>
int main()
{
int i;
clrscr();
char s[21];
cin>>s;
for(i=0;s[i];i++)
   if (s[i]=='c') s[i]='k';
cout<<s;
getch();
return 0;
}
```

تابع clrscr() برای پاک کردن صفحه خروجی به کار می رود. header file آن conio.hمی باشد.

برنامه‌ای بنویسید که یک رشته حداکثر ۲۰ کاراکتری را دریافت نماید، تمام حروف C را با K جایگزین نماید

```
#include <iostream.h>
#include <conio.h>
int main()
{
int i;
clrscr();
char s[21];
cin.getline(s,21);
for(i=0;s[i];i++)
   if (s[i]=='c') s[i]='k';
cout<<s;
getch();
return 0;
}
```

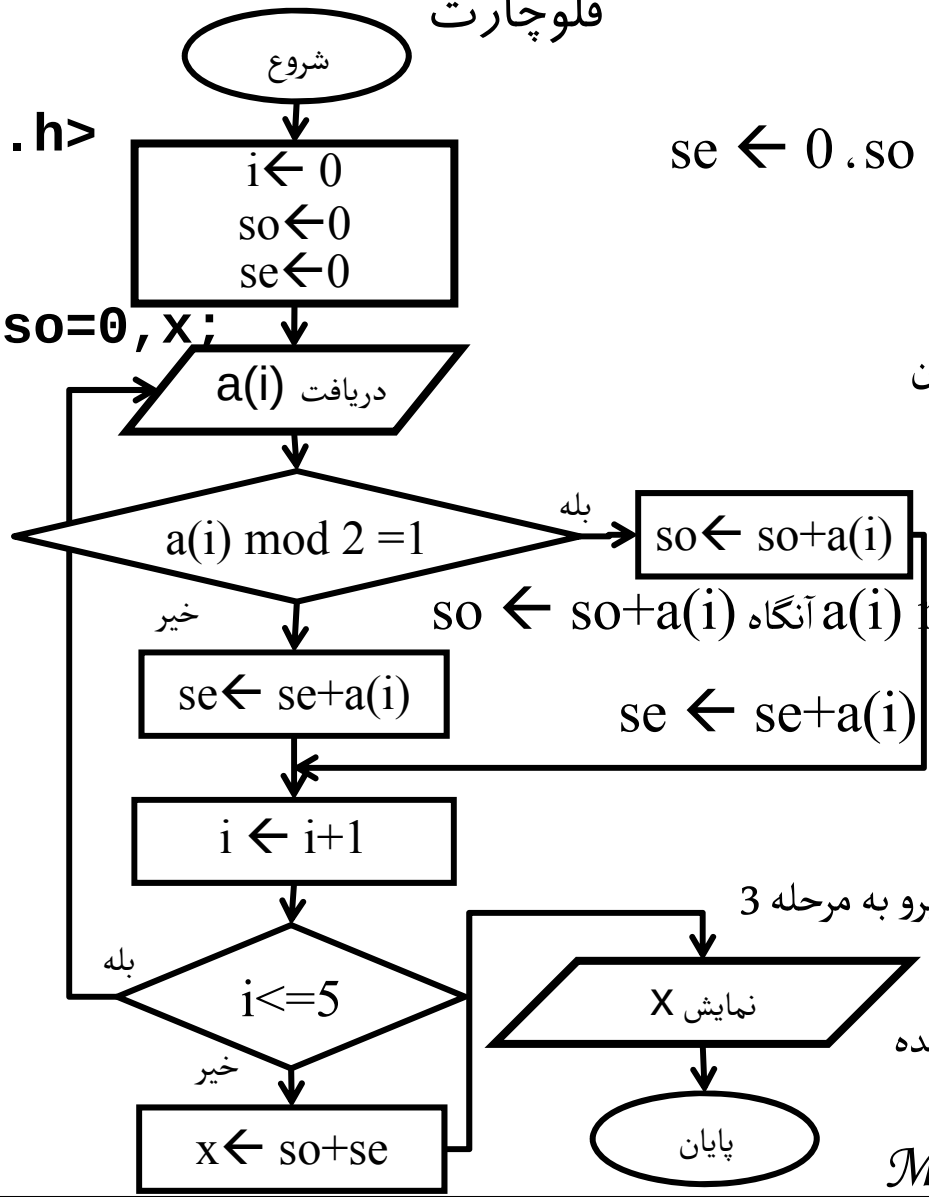
تابع cin.getline() برای دریافت یک متن با فاصله به کار می رود. آن header file iostream.h می باشد.

47 اختلاف بین مجموع اعداد فرد و زوج را در یک آرایه را پیدا کرده و نمایش دهد (5 عدد).

برنامه

```
#include<iostream.h>
#include<conio.h>
int main(){
 int i,a[5],se=0,so=0,x;
 for(i=0;i<5;i++)
  {
   cin>>a[i];
   if (a[i]%2==1)
      so+=a[i];
   else
      se+=a[i];
   }
 x=so-se;
 cout<<x;
 getch();
 return 0;
} 2013
```

فلوچارت



الگوریتم

1- شروع

2- $se \leftarrow 0 \text{ ، } so \leftarrow 0 \text{ ، } i \leftarrow 0$

3- $a(i)$ را دریافت کن

4- اگر $a(i) \bmod 2 = 1$ آنگاه $so \leftarrow so+a(i)$ درغیراینصورت $se \leftarrow se+a(i)$

5- $i \leftarrow i+1$

6- اگر $i<=5$ آنگاه برو به مرحله 3

7- $x \leftarrow so-se$

8- مقدار x زا نمایش بده

9- پایان

M. Damrudi