الگوریتم برنامه ای بنویسید که دو عدد را از ورودی گرفته و حاصل جمع آنها را در خروجی نمایش دهد.

1 شروع
2 عدد aوb را از ورودی بگیر
3 c ← a+b
4 c را چاپ کن
5 پایان

الگوریتم برنامه ای بنویسید که دو عدد را از ورودی بگیرد آنگاه بزرگترین آن را در خروجی نمایش دهد .

1 شروع
2 عدد aوb را از ورودی بگیر
3 a a>b را چاپ کن
یا
C ← a ← a>b اگر
C ← b در غیر اینصورت
4 c را چاپ کن
5 پایان

4 پایان

الگوریتم برنامه ای بنویسید که سه عدد را از ورودی گرفته آنگاه بزرگترین آن را در خروجی نمایش دهد .

1 شروع
2 a,b,c را از ورودی بگیر بخوان
3 max ← a
4 max→b آنگاه b<max آنگاه
5 max←c آنگاه a>max اگر
6 max را چاپ کن
7 پایان

الگوریتم برنامه ای بنویسیدکه صدد عدد صحیح را از ورودی بگیرد آنگاه بزرگترین آن را در خروجی نمایش دهد.

```
1  شروع
2   a را بخوان
3  max  c ← a
4   count←1
5   a  را بخوان
6    count++
7   max ←a آنگاه   a>max اگر
8    5 برو به مرحله count<100 اگر
9    max را چاپ کن
10   پایان
```

الگوریتم برنامه ای بنویسید که 100 عدد را از ورودی گرفته آنگاه بزرگترین و کوچک ترین آن را در خروجی نمایش دهد .

```
1   شروع
2   a را بخوان
3    max←a
4    min ←a
5    count←1
6   a  را بخوان
7    count ++
8   max←a  آنگاه a>max  اگر
9   min←a  آنگاه a<max  اگر
10   6 برو به مرحله count<100 اگر
11   min و  max را چاپ کن
12   پایان
```

الگوریتم برنامه ای بنویسید که آرایه 10 تایی اعداد را از ورودی گرفته آنگاه آن را از بزرگ به کوچک مرتب کند.
1 شروع 2 NM , CL ←0 , count←0 3 A[count] را بخوان 4 count++ 5 3 برو به مرحله count<10 اگر 6 max←A[0] 7 count ← 1 8 max←A[count] آنگاه A[count]>max NM←count 9 count++ 10 8 برو به مرحله count<10 اگر 11 0←A[NM] ← B[Cl]←max 12 cl ++ 13 6 برو به مرحله cl<10 اگر 14 پایان

0 ↓ جایگاه	1	2	3	4	5	6	7	8	9

Babel sort

Sort

با استفاده از آرایه کمکی→←با استفاده از آرایه کمکی

(درجا)

~~5~~ 0	~~96~~ 0	~~23~~ 0

الگوریتم برنامه ای بنویسید که آرایه ای را با استفاده از روش آرایه کمکی از یزرگ به کوچک مرتب کند .

1 شروع تنظیم پارامترها
2 count1←0 , count←0 تنظیم پارامترها
3 min ←A[count] بدست آوردن min
4 count ++ بدست آوردن min

5 اگر A[count]
min بدست آوردن min
min ←A[count] آنگاه
6 4 برو به مرحله count<n اگر بدست آوردن min

7 count ← 0 محاسبه max خانه sk
8 max ←A[0] محاسبه max خانه sk
9 count ++ محاسبه max خانه sk
10 max ←A[count1] آنگاه max<A[count] اگر محاسبه max خانه sk
sk←count
11 9 برو به مرحله count <n اگر محاسبه max خانه sk
12 B[count 1]←max جاگذاری max درB جاگذاری min درA
13 A[sk]←min جاگذاری max درB جاگذاری min درA
14 count 1++ تکرار مراحل بالا به تعداد خانه های آرایه
15 8 برو به مرحله count 1<n اگر تکرار مراحل بالا به تعداد خانه های آرایه
16 پایان

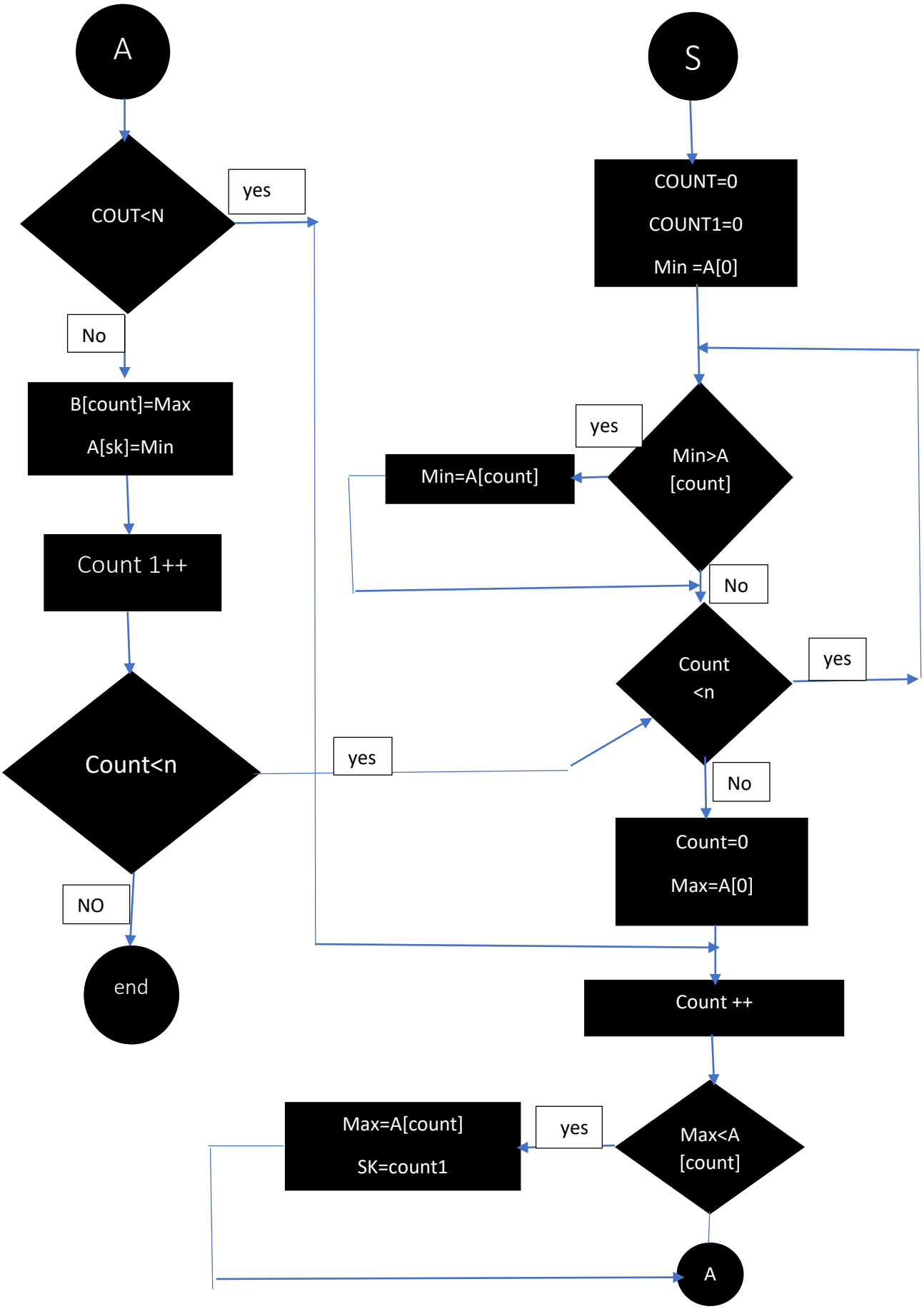
4→Max←220 NM→4

A

4	5	12	76	220

B

220	76	12	5	4


A
COUT<N
yes
No
B[count]=Max
A[sk]=Min
Count 1++
Count<n
yes
NO
end
S
COUNT=0
COUNT1=0
Min =A[0]
yes
Min>A
[count]
Min=A[count]
No
Count
<n
yes
No
Count=0
Max=A[0]
Count ++
Max<A
[count]
yes
Max=A[count]
SK=count1
A

شرط

```
IF (شرط)
  {
    دستورات IF
  }
 Else
  {
    دستورات else
  }
```

```
Max=a;
If(b>max)
   Max=b;
If(c>max)
    Max=c;
Cout<<max;
```

نکته : هر گاه در قسمت elseنیاز به انجام کاری بود می توان آن را حذف کرد .

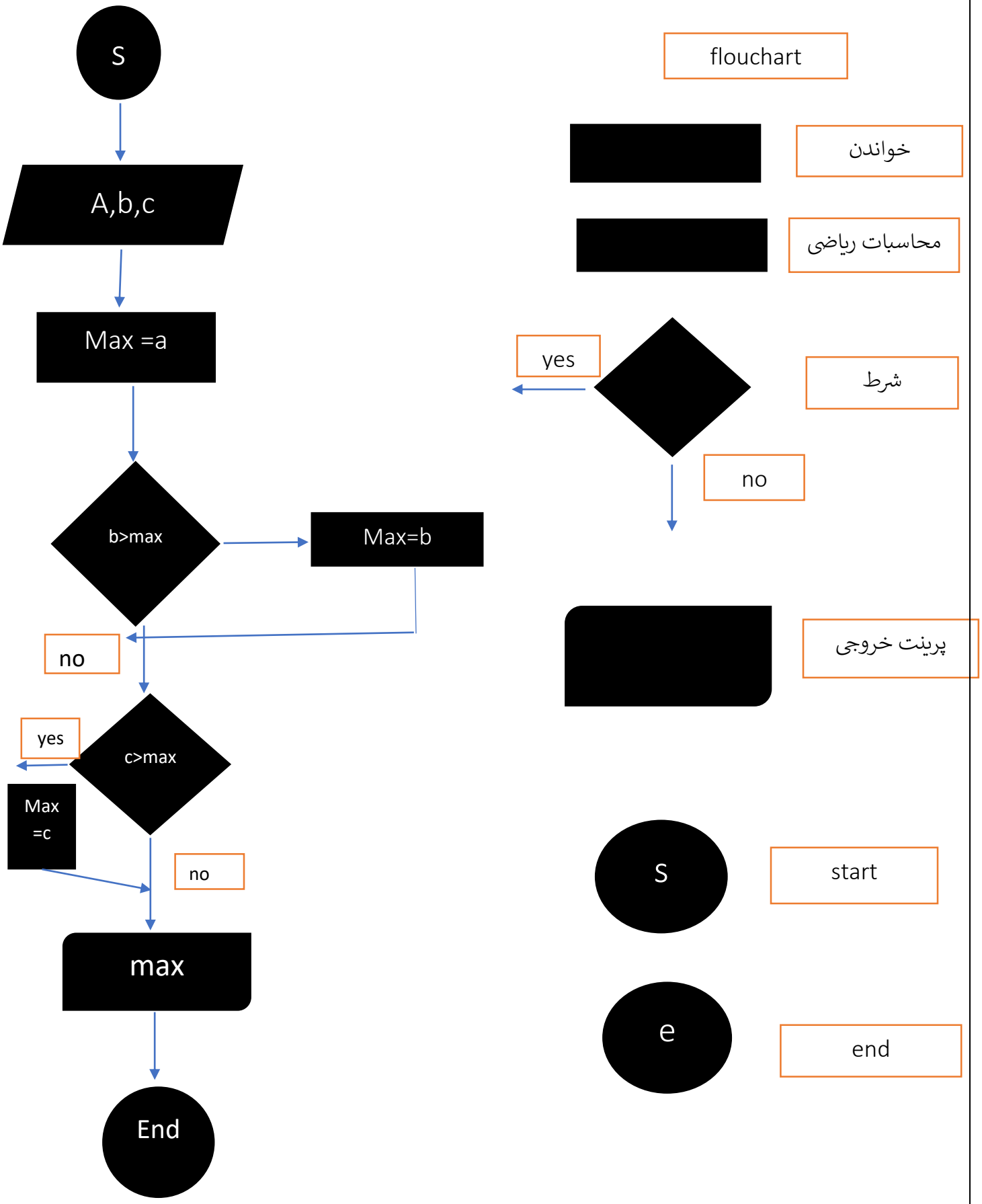
اعداد

1: صحیح

- Long int
- Signed Long int
- Unsigned Long int
- int
- signed int
- unsigned int

2 :اعشاری

- Float
- double


S
A,b,c
Max =a
b>max
Max=b
no
yes
c>max
Max
=c
no
max
End
flouchart
خواندن
محاسبات ریاضی
yes
شرط
no
پرینت خروجی
S
start
e
end

نکته : اگر زیر مجموعه if یاelse فقط یک دستور باشد نیاز {} نیست .

حلقه

1 : حلقه با تعداد تکرار ثابت

2 : حلقه با تعداد تکرار نامشخص

```
For(i=0 ; i<10 ; i++)
For ( i=10 ; i<20 ; i+2)
    {
     دستورات حلقه
}
```

5 بار تکرار می شود

```
For ( i=0 ; i<100 ; i*=2)
    {
      دستورات حلقه
}
```

بی نهایت بار تکرار می شود .

```
For (i==100 ; i>10 ; i\=2 )
    {
      دستورات حلقه
    }
```

4 بار تکرار می شود .

```
For (j=0 ; j<5 ; i+=6)
    {
     دستورات حلقه
}
```

یکبار تکرار می شود .

1

برنامه ای بنویسید که 100 عدد صحیح را از ورودی گرفته و بزرگترین آنها را در خروجی چاپ کند .

```
# inculde <stdio.h>
      Main ()
    {
       Int max ; n ;I ;
         Cin>>n;
          Max =n ;
          For (i=1 ; i<100 ; i++)
           {
             Cin >>n;
              If (n>max )
                Max = n ;
              }
                Cout << max ;
```

2

برنامه ای بنویسید که عددی را از ورودی گرفته اگر این عدد زوج بود 'even'را در خروجی چاپ کند در غیر اینصورت 'odd'را چاپ کند .

```
#inculde <iostream . h >\
       Main ()
    {
      Int n ;
       Cin >>n;
         If (n%2==0)
              Cout <<'even' ;
          Else
               Cout <<'odd'
          }
```

3

برنامه ای بنویسید که عدد صحیح aوbرا از ورودی گرفته آنگاه اعداد زوج بین آنها را در خروجی نمایش دهد .

```
#inculde <iostream.h>
      Main ()
     {
      Int a ; a ; b ; temp ; first ; j ;
          Cin>>a>>b;
              If(a<b)
              {
                Temp =a;
                 A=b;
                  B= temp ;
                 }
                   If ( a%2==0)
                    First=a+2;
                  Else
                        First=a+1;
                         For (j= first ; j <b ; j+2)
                         Cout <<j;
```

مجهول 1

b>a ?

یا

a>b?

if(b>a)

{

Temp=a;

A=b;

B= temp;

}

مجهول 2

If(a%2==0)

First =a+2;

Else

First = a+1;

A 57

B 93

First = 58

4

برنامه ای بنویسید که عددnرا از ورودی گرفته آنگاه اگر اول بود در خروجی ’F‘درغیر اینصورت در خروجی ’H‘را چاپ کند .

عددی اول است که فقط بر خودش بخش پذیر باشد .

N-1تا 2تقسیم کنیم به هیچکدوم نباید بخش پذیر باشد .

```
#inculde <iostream .h>
     Main()
  {
    Int a ; k ; counter=0;
     Cin>>n;
      For (k=2 ; k<=n-1 ; k++)
        If (n%k==0)
          Counter++;
     {
        Break;
         If(counter)
           Cout <<’H’
          Else
            Cout <<’F’
```

5

برنامه ای بنویسید که دو عدد صحیحaوbرا از ورودی گرفته آنگاه تمامی اعداد اول بین آنها را در خروجی چاپ کند .

```
#inculde<iostream.h>
      Main()
   {
      Int a ; b ; n ; counter ;
        Cin>>a>>b;
          If(a>b)
       {
            A=a+b;
             B= a-b;
              A=a-b;
        }
          For(n=a+1 ; n<b ; n++)
         }
           For (j=2 ; i<n%2 ; j++)
                   If ( n%j ==0)
          {
            Counter ++;
              Break ;
           }
             if (!counter)
               cout<<n;
            }
           }
```

آرایه
ساختمان داده ای است که

1: با تعداد خانه های مشخص و ثابت

2 : نوع خانه های مشخص و ثابت

نحوه تعریف آرایه

نوع خانه های آرایه

اسم آرایه [تعداد خانه های آرایه]

int – char

A آرایه است با 100 خانه از نوع اعداد صحیح a [100]

B آرایه است با 156 خانه از نوع حرف b [156]

نکته : به طور پیش فرض آرایه ها در زبان c از خانه صفر شروع می شوند .

آرایه

مرتب سازی	1 : sort
جستجو	2 : search
ادغام	3 : merge

1

برنامه ای بنویسید که عدد صحیح را از ورودی گرفته آنگاه اگر اول بود در خروجی ture در غیر اینصورت falseرا بیان کند .

```
#inculde<iostream.h>
      Main()
    {
       Int n; I; sum=0;
        Cin>>n;
      For(i=1 ; i<=n ; i++)
           If (n%i==0)
            Sum+=i
       If(sum==2*n)
        Cout <<'ture';
       Else
           Cout<<'false'
     }
```

2

برنامه ای بنویسید که اعداد کامل بین aوbرا نمایش دهد .

```
#inculde<iostream.h>
      Main()
  {
    Cin>>a>>b;                    aوb را بگیر
     If(a>b)                              همواره a را کوچکتر قرار بده
  {
    Temp = a;
     A=b;
     B=temp;
   }
   For(n=a+1 ; n<b ; n++)              کارهای زیر را برای اعداد بین aوbانجام بده
   {
      Sum=o
For (i=1 ; i<=n ; n++)
     If (n%i==0)
      Sum =I                          مقسوم علیه های n را با هم جمع کن ودر sumقرار بده
      If  (sum==z*n)
        Cout <<n;
    }
 }
```

اگر حاصل جمع مقسوم علیه های n دو برابر خودش باشد در خروجی آن را چاپ کن

.

نحوه خواندن آرایه از ورودی

```
For(i=0 ; i<n ; i++)
    Cin>>A[i];
```

نحوه چاپ کردن خانه های آرایه

```
For(i=0 ; i<n ; i++)
    Cout<<A[i] ;
```

3

برنامه ای بنویسید که آرایه ای با 100عنصر صحیح را از ورودی گرفته آنگاه بزرگترین آنها را در خروجی نمایش دهد .

```
#inculde<iostream.h>
  Main ()
 {
   Int A[100] ; I ; max ;
   For ( i= 0 ; i<100 ; i++)
       Cin >>A[i];
   Max =A[0];
   For (i=1 ; i<100 ; i++)
        If (A[i]>max)
            Max = A[i] ;
    Cout << max ;
   }
```

4

برنامه ای بنویسید که آرایه با 100 عنصر صحیح بگیرد آنگاه از بزرگ و کوچک مقادیر آنها را مرتب کند در خروجی نمایش دهد .

```
#inculde<iostream.h>
   Main()
 {
  Int A[100] ; I ; j=temp ;
  For (i=o ; i<100 ; i++)
        Cin >>A[i] ;
   For ( j=0 ; j <99 ; j ++)
   For (i=0 ; i<n -1 ; i++)
       If ( A[i] ; <A[i+1])
    {
       Temp = A[i];
        A[i]= A[i+1] ;
        A[i+1]=temp ;
      }
        For (i=0 ; i<100 ; i++)
             Cout << A[i] ;
        }
```

4 9	9 7	11 7	5 10	53 10	16 53	6 3

0 100

```
For(j=0 ; j<99 ; j++)
For (i=0 ; i<n-1 ; i++)
   If ( A[i]<A[i=1])
{
  Temp= A[ I ] ;
   A[ I ] = A[ I +1 ];
  A[i+1]= temp ;
```

4

برنامه ای بنویسید که آرایه ای صحیح را از ورودی گرفته آنگاه متغیری به نام keyرا از ورودی بگیرد .

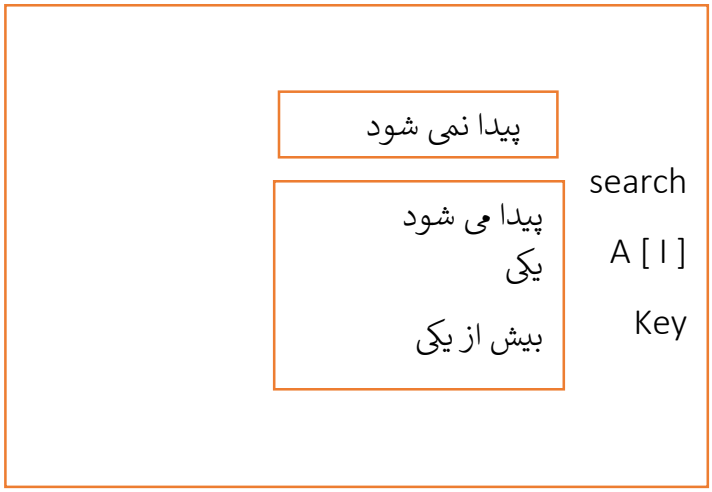
```
#inculde<iostream.h>
     Main()
  {
    Int a , A[100],key ;  →B[100],j=0
    For(i=0 ; i<100 ; i++)
           Cin>>A[ i ];
           Cin>>key ;
    For(i=0 ; i<100 ; i++ )
          If(key=A[ I ] )
           Break; →B[ j++]=I ;
      If(i==100)
      Cout<<'not found ' ; →if(j==0)
     Else→cout<<'not found ' ;
          Cout <<'found key: n<<I;→else
                                          For ( i=0 ; i<j ; i++)
                                           Cout<<'found key in <<B[ I ];
        }
```

```
#inculde<iostream.h>
    Main()
   {
     Int I , A[ 100 ] , B[ 100 ] , key , j=0 ;
        For ( i=0 ; i<100 ; i++ )
             Cin>>A [ I ] ;
        Cin>>key ;
     For ( i=0 ;  i<100 ; i++ )
        If (key=A[ I ] )
             B[j++]=I ;
        If (j==0)
           Cout<<'not found ' ;
         Else
             For ( i=0 ; i<j ; i++)
         Cout <<'found key in '<<B[ I ] ;
     }
```

5

برنامه ای بنویسید که دو آرایه صد عنصری را از ورودی گرفته و مرتب شده آن را جوری ادغام کند آنها را در آرایه cقرار دهد وآن را چاپ کند .

```
#inculde<iostream.h>
    Main()
  {
   For(i=0 ; i<100 ; i++ )
      Cin>> A[ I ] ;
   For ( i=0 ; i<100 ; i++ )
      Cin >>B [ I ] ;
    For( i= 0 ; i<100 ; i++)
       C [ 100+I ] = B [ I ] ;
     For ( I = 0 ; i<100 ; i++ )
     For ( i=0 ; i<100 ; i++)
        If ( C[ j ] = C [ j +1] ;
          C[ j + 1 ] = temp ;
      }
         Cout << C [ I ] ;
```



6

برنامه ای بنویسید تا زمانیکه کاربر عدد منفی وارد نکرده است اعدادی از کاربر گرفته و حاصل جمع آنها را محاسبه کند . در نهایت هر وقت کاربر عدد منفی وارد کرد از حلقه خارج شده و حاصل جمع اعداد وارد شده را در خروجی نمایش دهد .

```
#inculde<iostream.h>
  Main()
{
 Int n=0 , sum=0 ;
    While (n>=0)                    for(   ;  n>=0 ;      )
   {
    Cin >> n;
     Sum=sum+n;                     for(cin>>n ; n>=0 ;        )
    }                                for ( sum=0 ; n>=0 ; n>=0 )
      Cout<<sum-n;
    }
```

For (اجزای تشکیل دهنده)

N	Sum
~~0~~	~~0~~
~~1~~	~~1~~
~~2~~	~~3~~
3	6
-7	

حلقه

حلقه با تعداد تکرار مشخص

{

دستورات حلقه

حلقه با تعداد تکرار نامشخص

While (شرط)

{

دستورات حلقه

}

7

برنامه ای بنویسید که یک عدد صحیح را از ورودی گرفته آنگاه فاکتوریل آن را حساب کند .
(فقط از حلقه while استفاده شود)

```
#inculde<iostream.h>
    Main()
  {
   Int n , fact=1 ;
      Cin >> n ;
       While (n>1)
        {
          Fact=fact*n
          n--;
          }
           Cout<<fact ;
          }
```

N	Fact	خروجی
4	4*3*2	24
3		
2		
1		

8

برنامه ای بنویسید که دو عدد صحیح nوmرا از ورودی گرفته آنگاه تا وقتی mوnباشد
!(m-n)محاسبه کرده و در خروجی نمایش دهد .

```
#inculde<iostream.h>
    Main()
  {
 Int m, n , f ;
  While(m>n)
   {
     Fact=1;
      Cin>>m>>n;
       F=m-n
     For (i=1 ; i< f ; i++ )
           Fact *I = I ;
          }
            Cout <<fact ;
           }
```

جواب سوال 8

```
#inculde<iostream.h>
    Main()
  {
    Int m=1, n=0 , fact , x ;
       While (m>n)
     {
       Fact=1;
        X=m-n ;
         While (x>1)
        {
         Fact = fact*x ;
          x-- ;
         }
           Cout <<fact ;
            Cin >>m>>n;
          }
```

9

خروجی برنامه زیر چیست ؟خروجی برنامه زیر 120 است .

```
#inculde<iostream.h>
    Main()
  {
    Int k , I , t=1 ;
     While(2>1)
   {
    K=5
    For(i=1 ; i<5 ; i++)
         T=t*i ;
     Cout <<t*k ;
     Break ;
    }
  }
```

I	K	T	خروجی
–	–	–	120
1	5	T*i=1	1
2		1	
3		2	
4		6	
		24	

10

خروجی برنامه زیر چیست ؟ خروجی برنامه 120 720 5040

```
#inculde<iostream.h>
   Main ()
  {
Int k , I , t=1 ;
  K=7 ;
   While (2>1)
  {
   For (i=1 ; i<k ; i++)
          T*=I ;
     Cout <<t*k ;
       If (t<120)
      Break ;
       K -- ;
  }
     }
```

I	K	T	خروجی
–	–	I	
	7	1*2*3*4*5*6 720*7	5040
	6	1*2*3*4*5 120*6	720
	5	1*2*3*4 24*5	120

نوع خروجی اسم تابع (ورودی های تابع)

{

دستورات تابع

{

خروجی Return

1 : جلوگیری از تکرار 2 : خواناتر شدن برنامه

1

تابعی بنویسید که طول و عرض مستطیل را گرفته و مساحت آن را به برنامه اصلی بازگرداند .

```
Float  masahat(float a , float b )
   {
     Return a*b ;
    }
```

2

تابعی بنویسید که عدد صحیح را از ورودی گرفته آنگاه اگر این عدد اول بود به برنامه اصلی بازگرداند . در غیر اینصورت 0 برگرداند .

```
Int is first (int a )
  {
    Int I ;
     For (i=2 ; i<n\2 ; i++)
          If ( n%i==0)
           Return 0 ;
      Return 1 ;
   }
```

3

تابعی بنویسید که عددی صحیح را از ورودی گرفته آنگاه اگر کامل بود 1 و در غیر اینصورت 0 را به برنامه اصلی بازگرداند .

```
Int is –comp( int n )
  Int I , sum=0 ;
For (i=1 ; i<n ; i++)
  If (n%i==0)
    Sum+=I ;
    Return 1 ;
Else
      Return 0 ;
```

4

تابعی بنویسید که آرایه 100 عنصری را از ورودی گرفته آنگاه بزرگترین آن ها را در خروجی نمایش دهد .

```
#inculde<iostream.h>
 Main()
   {
     Int max arr()
     For(i=0 ; i<100 ; i++)
           Cin >>A[ I ] ;
        Max = A [ 0 ] ;
     For ( i=1 ; i<100 ; i++ )
           If ( A [ I ] > max )
              Max = A [ I ] ;
       Return max ;
    }
```

5

با استفاده از تابع برنامه ای بنویسید که اعداد اول بین 1 تا 1000000 را محاسبه کرده و در خروجی نمایش دهد .

```
#inculde<iostream . h >
          Is first  تابع
   Main()
{   Long int I ;
      For (i=1 ; i<1000000 ; i++)
            If ( is first (i) )
             Cout << I ;
        }
```

تابعی بنویسید که سه عدد صحیح را از خروجی گرفته و بزرگترین آن ها را به خروجی بفرستد .

```
#inculde<iostream.h>
  Main ()
 {
   Int maximum(int a , int b , int c )
 {
  Int max ;
  Max sa ;
   If (max<b)
     Max=b ;
    If (max<c)
         Max = c ;
      Return max ;
```

تابع بازگشتی ←اگر در تعریف تابعی از خود تابع استفاده شود به آن تابع بازگشت می گوییم .

نوع خروجی اسم تابع (ورودی های تابع)
{
If (شرط بازگشت ; return مقدمه بازگشت ;

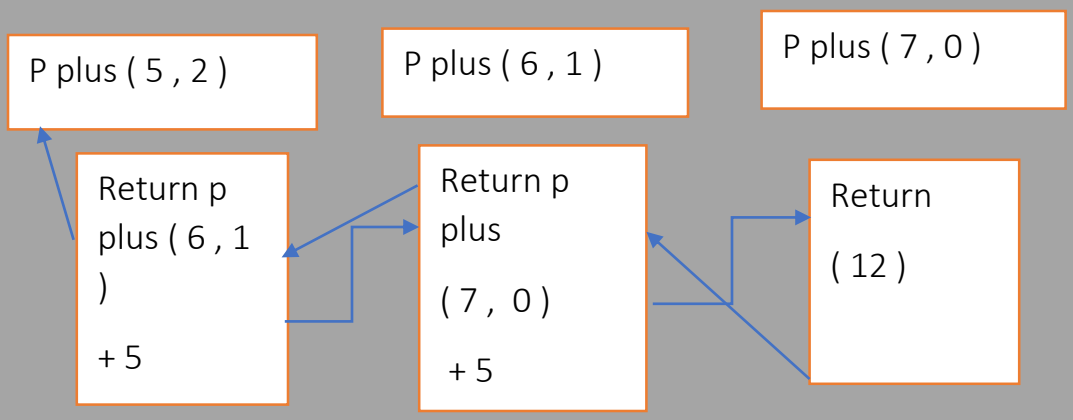
تعریف بازگشت Else return

}

```
Main()
{
  Int x=4 , k
  Int y =3 ;
  K=plus(x+y)
  Cout<<k ;
```

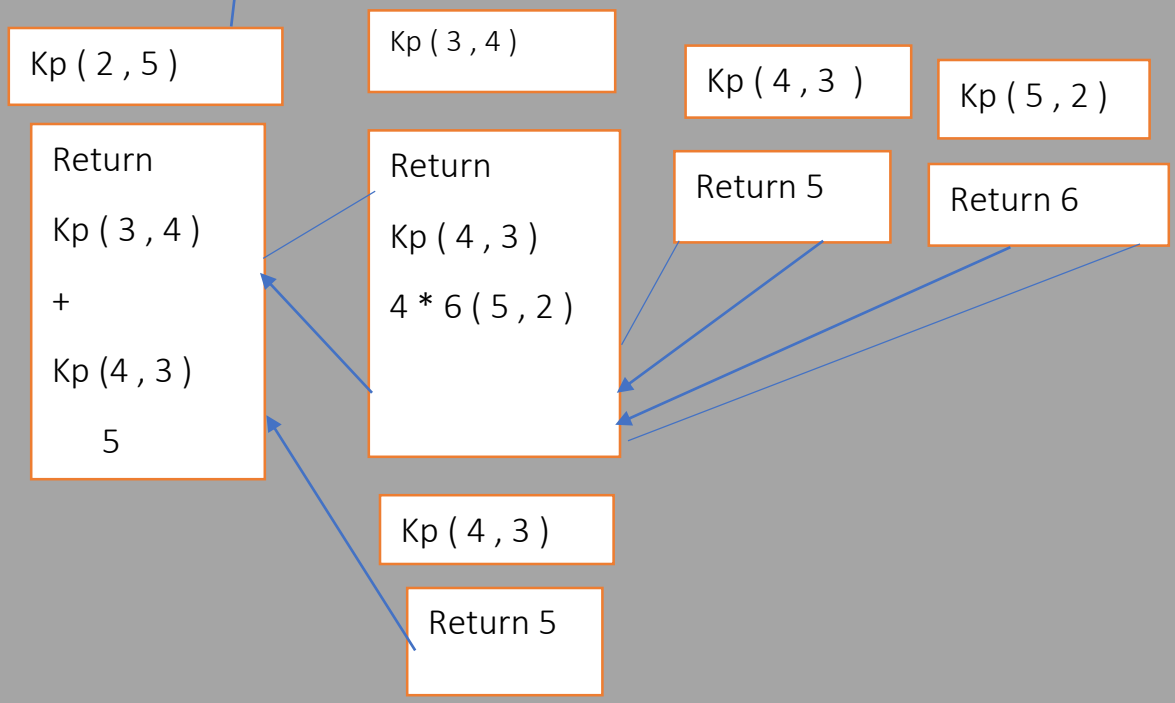
عملیات x+yرا بازگشتی تعریف کنیم .

```
If y==0  → x
Else → x + ( y – 1 ) + 1

Int plus ( int x , y )
{
   If ( y==0 )      return x ;
    Else return plus ( x + 1) , y – 1 ) ;
}

#inculde<iostream.h>
    Main ()
  {
    Int x , y , z ;
     X = 5 ;
     Y = 2 ;
     Z= p plus ( x , y ) ;
     Cout << z ;
  }

Int p plus ( int x , y )
{
  If ( y == 0 )  return x + 5 ;
   If ( y > 0 ) return plus ( x+5 , y – 1 ) + 5 ;
    If ( y < 0 ) return plse ( x – 1 , y + 1 ) +  5 ;
```

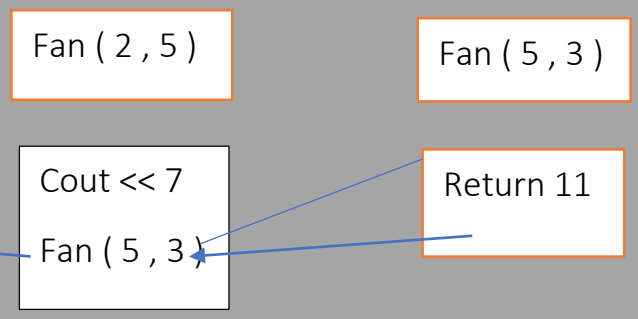



```
Int fan ( int x, y )
 {
   If ( x > y )    return ( x + 6) ;

Else
      {
        Cout << x + y ;

#inculde<iostream.h>
   Main()
 {
  Int x , y , z ;
   X = 2 ;
    Y = 5 ;
    Z = fan ( x , y ) ;
    Cout << z ;
   }
```

```
Int kp (int x , y )
{
  If (x>y) return x + 1 ;
Else return kp ( x + 1 , y_1 ) + kp ( x + 2 , y – 2 )
  }
  Main()
  {
    Int m , n , p ;
     M = 5 ;
      N = 2 ;
       P = kp ( n , m ) ;
       Cout << p ;
```



```
Main ()
{
  Int x , y , z ;
   X = 2 ;
   Y = 5 ;
   Z = fan ( x , y ) ;
   Cout << z ;
 }

Fan ( Int x , y )
If ( x > y ) return ( x + 6 )
Else
     {
        Cout << x + y ;
```



تفریق با استفاده از تابع بازگشتی (x – y)

شرط بازگشت If y = = 1 → return x – 1

If (y > 1) → (x – 1) – (y – 1)

If (y < 1) → (x + 1) _ (y + 1)

```
Int main ()
{
  If ( y > 1 ) return main ( x – 1 , y – 1 ) ;
  If ( y < 1 ) return main  ( x + 1  , y + 1 ) ;
   If ( y = = 1 ) return  x – 1  ;
 }
```

```
Int div ( int x , y )
 {
   If ( x < y )  return 0 ;
else return 1 + div ( x – y , y ) ;
```

تعریف بازگشت تقسیم صحیح

X | Y
C

If (x < y) return 0
Else return fan (x – 1 , y + 1)

X < y →

X | y

return0

1+ (x – y) | Y

5 | 2 → 1+3 | 2 → 1+1 | 2
0

7

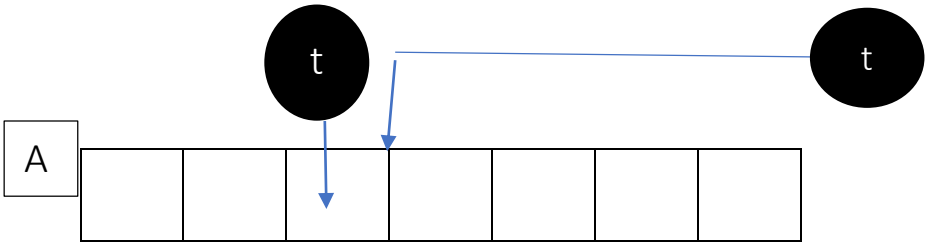
تابعی بنویسید که آرایه a را از ورودی گرفته آنگاه از کوچک به بزرگ مرتب کند . در نهایت آن ها را به خروجی ارسال کند .

```
Int  * sort ( int *A)
{
 Int I , j , temp ;
  For ( i=0 ; i<len (A) ; i++ )
  For ( j =I ; j <len (A) ; j ++ )
     If ( A [ I ] < A [ j + 1 ]
    {
      Temp = A [ I ] ;
       A [ j ] = A [ j + 1 ]
        A [ j + 1 ] = temp ;
     }
       Return A ;
      }
```

8

برنامه ای بنویسید که توسط سه تابع داده ای صحیح را از درون آرایه به مقدار 100 عنصر صحیح جستجو کند در صورت یافتن آدرس خانه ی آن را به برنامه اصلی باز گرداند در صورت یافت نشدن آن مقدار 1- را به برنامه اصلی بازگرداند .

```
#inculde<iostream.h>
 Main ()
{
 Int A [ 100 ] ;
 A = Read Array (A , 100 )
  A = sort ( A ) ;
   Cin >> t ;
  K = search ( A , t ) ;
Int * Read Array ( int * A , int I )
    {
      Int j ;
      For ( j=0 ;  j < i ; j++ )
           Cin >> A [ j ] ;
       Return  A  ;
}
```



If t > A [3]

از کوچک به بزرگ مرتب شده است .

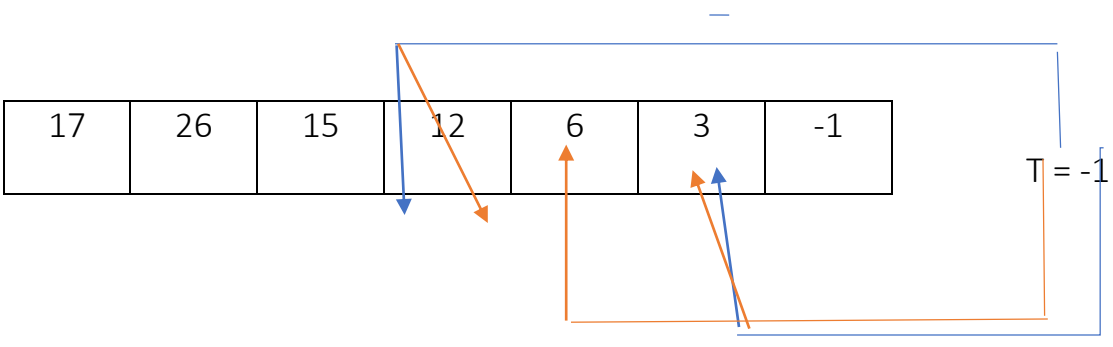
این تیکه از آرایه در جواب ما نیست دیگر نیاز به گشتن نیست

نکته : باینری سرچ برای سرچ های بزرگ استفاده می شود هر بار آن را با وسط آرایه چک می کنیم .

در هر گام آرایه را نصف می کنیم

```
Int search B ( int * A , int t )
    Int mid , s , d ;                    عنصر وسط , start , dis
     S=0
     D=99 ;
      While ( 2 > 1 )        حلقه بی نهایت
       {
         Mid = ( s + d + 1 ) /2 ;
          If ( t > A [ mid ]
          D = mid ;
           If t < A [ mid ]
             S = mid ;
            If ( t ==A [ mid ] )
                Return mid ;
              If ( s==d )
                Return -1 ;
```

.

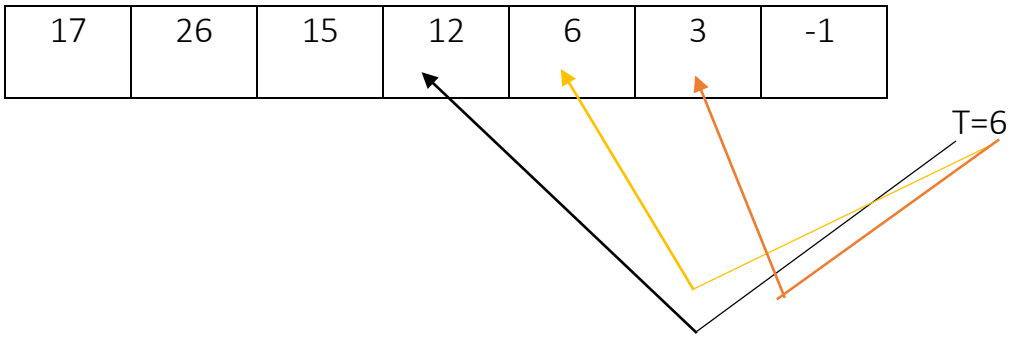


بدون 1+

s	0	3	4	5
d	6	6	6	**6**
mid	6+0=6%2=3	6+3=9%2=4	4+6=10%2=5	**5+6=11%2=5**

با 1+

s	0	3	5	6
d	6	6	6	**6**
mid	0+6+1=7%2=3	3+6+1=10%2=5	5+6+1=12%2=6	**Return6**



s	0	3	3
d	6	6	5
mid	0+6+1=7%2=3	6+3+1=10%2=5	3+5+1=9%2=4 Return 4

سرچ باینری را می توان به صورت تابع بازگشتی نوشت .

تابع سرچ باینری را به صورت بازگشتی بنویسید .

```
Int search B R ( int *A ,int s , int d , int t )
  {
    Int mid ;
    Mid ( s + d + 1 ) /2 ;
    If ( t > A [ mid ] )
   Return search B R ( A , s , mid , t ) ;
   If ( t < A  [ mid ] )
    Return search  B R ( A , mid , d , t ) ;
     If ( t == A [ mid ] )
       Return mid ;
         If ( s == d )
            Return _1 ;
     }
```

تعریف متغیر

Global سراسری قبل از تابع تعریف می شود .

تا جای ممکن متغیرهای Global را کمتر تعریف می کنیم چرا چون از اول برنامه تا آخر برنامه حافظه را اشغال می کند .

Local محلی

در خود تابع تعریف می شود .

تعریف کتابخانه 1

اسم (ورودی)

→ Global uar

تعریف تابع شماره 1 2 →

نوع خروجی

{

Local uar

}

Return خروجی

شماره 2

شماره 3

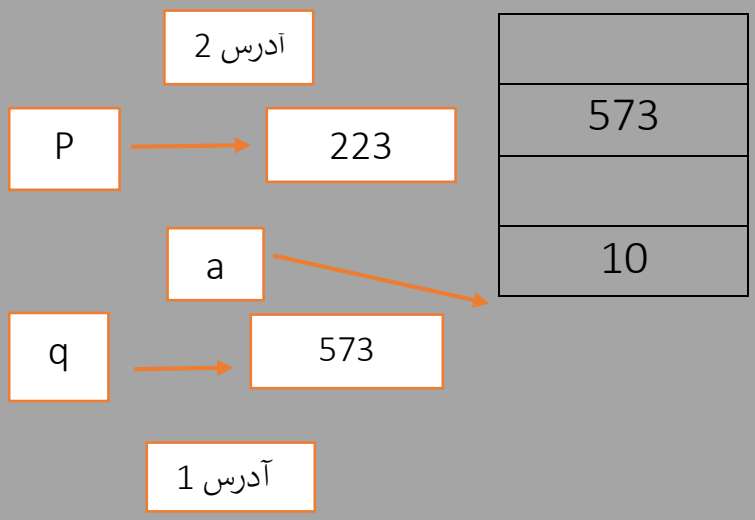
.....و

بعد از تابع

```
Main()
 {
{
```

هر جا که تعریف شود همان جا شناخته می شود .

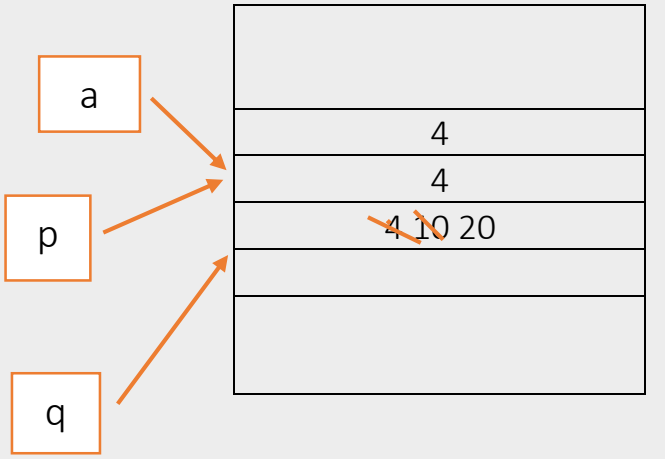
اشاره گر ها
Pointer




```
Int a , **p , q* ;                          int a = 10 ;
a =10 ;                                     int * p ;
q = & a ;                                   p = & a ;
p = & q ;
cout <<a << q << p ;
10  آدرس 1 آدرس 2
Cout << * q ;                                    متغیر a
          10
Cout <<* p ;                              آدرس متغیر q
        آدرس 1
Cout << ** p ;                                        آدرس آدرس متغیر p
            10
```

نکته

اسم اشاره یک اشاره گر است که به خانه صفرم آرایه اشاره می کند .



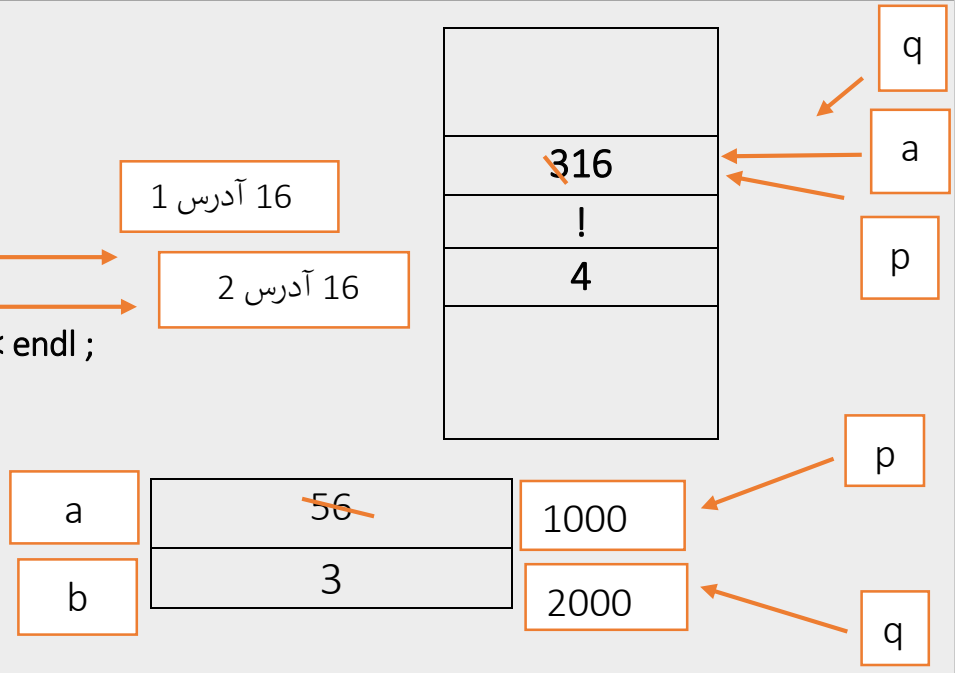

```
Int a [ 5 ] , * p , q ;
a [ 0 ] = a [ 1 ] = a [ 2 ] = 4 ;
p= a ;
q = & a [ 2 ] ;
*(p+2) = a [ 1 ] * a [ 2 ] ;
*q =(*q) + ( * p );
Cout << * p << *q ;
```

4 20

1

خروجی تک برنامه زیر چیست ؟

```
Int a , b , * p , *q ;
a = 3 ;
b =4 ;
p = & b ;
a = ( * p ) * ( * q ) ;
p = & a ;
q = p ;
cout << p << * p << endl ;
cout <<q << * q ;
```



```
Int a , b , * p , * q ;
        a = 5 ;
        b = 3 ;
        p = & a ;
        q = & b ;
         a = b* 2 ;
       cout << p << * p << endl ;
      cout << q << * q ;
```

2

تابعی بنویسید که عددی صحیح را از ورودی گرفته آنگاه حاصل جمع ارقام آن را به خروجی اصلی ارسال کند .

```
Int sum ( int n )
 {
   Int s =0 , m=n ;
   While ( m l = 0 )
  {
    S = s + ( m % 10 ) ;
    m = m /10 ;
 }
    Return s ;
 }
```

3

تابعی بنویسید که عددی صحیح را از ورودی گرفته آنگاه تعداد ارقام آن را به خروجی ارسال کند .

```
int sum ( int n )
{
  int s =0 , m=n , counter=0 ;
  while ( m! = 0 )
  {
   S= s+( m%10;)
   M = m/10 ;
    Counter ++ ;
  }
      Return s ;
   }
```

```
int *sams ( int n )
   {
    Int s=0 , m=n , counter=0 ;
    Int *p;
     While (m!=0)
    {
      s=s+(m%10);
      m=m/10 ;
      counter ++ ;
    }
       P=& s ;
      *(p+1)=counter ;
     Return p ;
    }
```

```
Main ()
{
 int *a , s , counter ;
 int k ;
 cin >>k ;
 a = sams(k) ;
 s=*a ;
    a[0]
 counter=*(a+1) ;
                a[1]
}
```

1

تابع زیر چه عملیاتی را انجام می دهد ؟

جستجوی دودویی بازگشتی Kecursive binary search

```
Int my fanc ( int L , int h , int x )
      {
        Int m ;
        m = ( L+h ) / 2 ;
          if (L>h)
                return 0 ;
         if (x == a[m] )
                return 1 ;
         if ( x< a[m] )
                return my fanc ( L , m_1 , x )
            else ;
                 return my fanc ( m+ 1 , h , x )
          }
```

2

خروجی برنامه زیر را محاسبه کنید .

```
int s ()
{
  Static int q=1 ;
  cout << q++<<"\n " ;
        return q ;
  }
    Main ()
  {
     cout << s () << " \ n " ;
     cout << s () ;
   }
```

1	1
2	2
3	3
4	4

نکته :

قبل از اجرای تابع از جستجوی دودیی استفاده می کنیم که با یک مرتب سازی از کوچک به بزرگ داشته باشیم .

3

خروجی تابع زیر را بدست آورید .

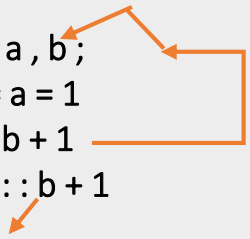
خروجی : 4 3 3 نا مشخص

```
int vote ( int a )
{   static int b ;
Cout<<b ;
b=a+1 ;
cout<<b;
return ( b ) ; }
main ()
{  int a =1 ;
a= vote ( 2 ) ;
a = vote ( a ) ; }
```

4

خروجی برنامه زیر چیست ؟

```
#inculde<iostream.h>
    Main()
  {
    int a , b ;
    a=b=2 ;
   void main ()
  {
   int a , b ;
   a = a = 1
   a = b + 1
   a = : : b + 1
```



Global متغیر

Local نه متغیر

اگر در تابعی متغیری هم نام یک متغیر سراسری تعریف کنیم برای دسترسی به متغییر سراسری از علامت (: :) استفاده می کنیم .

4 4 2 3

نکته:
(Over Loding)

تعریف توابع هم نام

```
int add ( int a , int b )
{
   }

int add ( float c , int d )
{
  }
```

5

مقدار what (5 و 3) چیست ؟
مقدار آن 12 است .

```
int what ( int m , int n )
 {
  If ( m == 1 )   return 3 ;
  If ( n == 1 )    return 2 ;
  If ( m == 0 )    return 4 ;
  Return z * what ( m – 1 , n – 2 )
 }
```