|  |
| --- |
| الگوریتم برنامه ای بنویسید که دو عدد را از ورودی گرفته و حاصل جمع آنها را در خروجی نمایش دهد. |
| 1 شروع  2 را از ورودی بگیر aوbعدد  3 c ← a+b  4 را چاپ کن c  5 پایان |

|  |
| --- |
| الگوریتم برنامه ای بنویسید که دو عدد را از ورودی بگیرد آنگاه بزرگترین آن را در خروجی نمایش دهد . |
| 1 شروع  2 را از ورودی بگیرaوbعدد  3 را چاپ کن a a>b  یا  C ← a ← a>bاگر  C ← bدر غیر اینصورت  4 را چاپ کن c  5پایان  4 پایان |

|  |
| --- |
| الگوریتم برنامه ای ینویسید که سه عدد را از ورودی گرفته آنگاه بزرگترین آن را در خروجی نمایش دهد . |
| 1 شروع  2 را از ورودی بگیر بخوان a,b,c  3 max ← a  4 max→b آنگاهb<max آنگاه  5 max←c آنگاه a>max اگر  6 را چاپ کن max  7 پایان |

|  |
| --- |
| الگوریتم برنامه ای بنویسیدکه صدد عدد صحیح را از ورودی بگیرد آنگاه بزرگترین آن را در خروجی نمایش دهد. |
| 1 شروع  2 را بخوان a  3 max c ← a  4 count←1  5 را بخوان a  6 count++  7 max ←a آنگاه a>max اگر  8 برو به مرحله 5 count<100اگر  9 را چاپ کنmax  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 را بخوان a  3 max←a  4 min ←a  5 count←1  6 را بخوان a  7 count ++  8 max←a آنگاه a>max اگر  9 min←a آنگاه a<max اگر  10 برو به مرحله 6 count<100اگر  11 را چاپ کن max و min  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 را بخوان A[count]  4 count++  5 برو به مرحله 3 count<10اگر  6 max←A[0]  7 count ← 1  8 max←A[count]آنگاه A[count]>max  NM←count  9 count++  10 برو به مرحله 8count<10 اگر  11 0←A[NM] ← B[Cl]←max  12 cl ++  13 برو به مرحله 6 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* |

|  |  |  |
| --- | --- | --- |
| *96* | *23* | *5* |

|  |
| --- |
| 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] minبدست آوردن  4 count ++ minبدست آوردن  5 A[count]اگر  min minبدست آوردن  min ←A[count]آنگاه  6 برو به مرحله 4 count<n اگر minبدست آوردن    7 count ← 0 sk خانه max محاسبه  8 max ←A[0] sk خانه max محاسبه  9 count ++ sk خانه max محاسبه  10 max ←A[count1] آنگاه max<A[count] اگر sk خانه max محاسبه  sk←count  11 برو به مرحله 9 count <n اگر sk خانه max محاسبه  12 B[count 1]←max Aدر min جاگذاری Bدرmax جاگذاری  13 A[sk]←min Aدر min جاگذاری Bدرmax جاگذاری  14 count 1++ تکرار مراحل بالا به تعداد خانه های آرایه  15 برو به مرحله 8 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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 220 | 76 | 12 | 5 | 4 |

B

yes

COUT<N

A

end

Count<n

Count 1++

B[count]=Max

A[sk]=Min

COUNT=0

COUNT1=0

Min =A[0]

S

No

Min>A[count]

yes

Min=A[count]

No

yes

Count<n

c

No

yes

NO

Count=0

Max=A[0]

Max=A[count]

SK=count1

A

Max<A[count]

Count ++

yes

|  |  |
| --- | --- |
| Max=a;  If(b>max)  Max=b;  If(c>max)  Max=c;  Cout<<max;  نکته : هر گاه در قسمت else نیاز به انجام کاری بود می توان آن را حذف کرد . | شرط  IF (شرط)  {  دستورات IF  }  Else  {  دستورات else  } |

|  |
| --- |
| اعداد  int  signed int  unsigned int  Long int  Signed Long int  Unsigned Long int  1: صحیح    Float  double  : 2اعشاری |

flouchart

خواندن

s

End

max

no

yes

c>max

no

Max=b

b>max

Max =a

A,b,c

شرط

start

end

پرینت خروجی

yes

no

محاسبات ریاضی

s

e

Max=c

|  |
| --- |
| نکته : اگر زیر مجموعه if یا elseفقط یک دستور باشد نیاز  {} نیست . |

|  |
| --- |
| 1 : حلقه با تعداد تکرار ثابت  2 : حلقه با تعداد تکرار نامشخص  حلقه  For (j=0 ; j<5 ; i+=6)  {  دستورات حلقه  {  یکبار تکرار می شود .  For ( i=0 ; i<100 ; i\*=2)  {  دستورات حلقه  {  بی نهایت بار تکرار می شود .  For(i=0 ; i<10 ; i++)  For ( i=10 ; i<20 ; i+2)  {  دستورات حلقه  { 5 بار تکرار می شود  For (i==100 ; i>10 ; i\=2 )  {  دستورات حلقه  }  4 بار 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 2

If(a%2==0)

First =a+2;

Else

First = a+1;

مجهول 1

b>a ?

یا

a>b?

if(b>a)

}

Temp=a;

A=b;

B= temp;

{

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  1: با تعداد خانه های مشخص و ثابت  2 : نوع خانه های مشخص و ثابت  ساختمان داده ای است که  نحوه تعریف آرایه نوع خانه های آرایه    اسم آرایه ]تعداد خانه های آرایه [ char ─int    Aآرایه 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 .  1 : sort مرتب سازی  2 : search جستجو  3 : merge ادغام  آرایه |

|  |
| --- |
| 1  برنامه ای بنویسید که عدد صحیح را از ورودی گرفته آنگاه اگر اول بود در خروجی ture در غیر اینصورت false را 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; را بگیر aوb  If(a>b) را کوچکتر قرار بده aهمواره  {  Temp = a;  A=b;  B=temp;  }  For(n=a+1 ; n<b ; n++) انجام بده aوb کارهای زیر را برای اعداد بین  {  Sum=o  For (i=1 ; i<=n ; n++)  If (n%i==0)  Sum =I قرار بده sum را با هم جمع کن ودر nمقسوم علیه های  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];  For(i=0 ; i<n ; i++)  Cout<<A[i] ;  نحوه چاپ کردن خانه های آرایه |

|  |
| --- |
| 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 |

1. 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 ] ; |

search

A [ I ]

Key

پیدا نمی شود

پیدا می شود یکی

بیش از یکی

ی

|  |
| --- |
| 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  0  1  2  3  -7 | Sum  0  1  3  6 | |

حلقه

حلقه با تعداد تکرار نامشخص

While (شرط )

}

دستورات حلقه

{

حلقه با تعداد تکرار مشخص

}

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()   |  |  |  | | --- | --- | --- | | **N**  **4**  **3**  **2**  **1** | **Fact**  **2\*3\*4** | **خروجی**  **24** |   {  Int n , fact=1 ;  Cin >> n ;  While (n>1)  {  Fact=fact\*n  n--;  }  Cout<<fact ;  } |

|  |
| --- |
| 8  برنامه ای بنویسید که دو عدد صحیح mوnرا 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 ;   |  |  |  |  | | --- | --- | --- | --- | | **I**  **─**  **1**  **2**  **3**  **4** | K  ─  5 | T  ─  T\*i=1  1  2  6  24 | خروجی  120  1 |   While(2>1)  {  K=5  For(i=1 ; i<5 ; i++)  T=t\*i ;  Cout <<t\*k ;  Break ;  }  } |

|  |
| --- |
| 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**  **─**  **7**  **6**  **5** | **T**  **I**  **1\*2\*3\*4\*5\*6**  **720\*7**  **1\*2\*3\*4\*5**  **120\*6**  **1\*2\*3\*4**  **24\*5** | **خروجی**  **5040**  **720**  **120** | |

|  |
| --- |
| نوع خروجی اسم تابع (ورودی های تابع )  }  دستورات تابع  }  Returnخروجی  1 : جلوگیری از تکرار 2 : خواناتر شدن برنامه |

|  |
| --- |
| 1  تابعی ینویسید 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 ;  } |

|  |
| --- |
| 6  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 ;  Else return تعریف بازگشت  {  Main()  {  Int x=4 , k  Int y =3 ;  K=plus(x+y)  Cout<<k ; |

|  |
| --- |
| عملیات x+yرا بازگشتی تعریف کنیم .  → x If y==0  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 ; |

|  |
| --- |
| Return  ( 12 )  Return p plus  ( 7 , 0 )  + 5  Return p plus ( 6 , 1 )  + 5  P plus ( 5 , 2 )  P plus ( 6 , 1 )  P plus ( 7 , 0 )  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 ;  Kp ( 3 , 4 )  Kp ( 2 , 5 )  Kp ( 4 , 3 )  Return 6  Kp ( 5 , 2 )  Return 5  Return 5  Kp ( 4 , 3 )  Return  Kp ( 4 , 3 )  4 \* 6 ( 5 , 2 )  Return  Kp ( 3 , 4 )  +  Kp (4 , 3 )  5 |

|  |
| --- |
| Main ()  Fan ( 5 , 3 )  Fan ( 2 , 5 )  {  Int x , y , z ;  X = 2 ;  Return 11  Cout << 7  Fan ( 5 , 3 )  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 )  If y = = 1 → return x – 1 شرط بازگشت  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 Y X  C  If ( x < y ) return 0  Else return fan ( x – 1 , y + 1 )  X < y → return0  X y  1+ ( x – y ) Y    2 1+1 → 2 1+3 → 5 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 ;  } |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

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 ; |

.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 17 | 26 | 15 | 12 | 6 | 3 | -1 |

T = -1

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

بدون 1+

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

با 1+

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| -1 | 3 | 6 | 12 | 15 | 26 | 17 |

T=6

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

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 ;  } |

|  |
| --- |
| سراسری قبل از تابع تعریف می شود . Global  تا جای ممکن متغیرهای Global را کمتر تعریف می کنیم چرا چون از اول برنامه تا آخر برنامه حافظه را اشغال می کند .    تعریف متغیر  محلی Local  در خود تابع تعریف می شود . |

|  |
| --- |
| اسم (ورودی ) 1 تعریف کتابخانه  Global uar  →  نوع خروجی  }  Local uar  }  Return خروجی  2 → تعریف تابع شماره 1  شماره 2  شماره 3  و........  بعد از تابع  Main()  {  {  هر جا که تعریف شود همان جا شناخته می شود . |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| اشاره گر ها  Pointer   |  | | --- | |  | | 573 | |  | | 10 |   a  آدرس 1  q  573  آدرس 2  P  223 |

|  |
| --- |
| Int a , \*\*p , q\* ; int a = 10 ;  a =10 ; int \* p ;  q = & a ; p = & a ;  p = & q ;  cout <<a << q << p ;  آدرس 2 آدرس 1 10  Cout << \* q ; a متغیر  10  Cout <<\* p ; q آدرس متغیر  آدرس 1  Cout << \*\* p ; pآدرس آدرس متغیر  10 |

|  |
| --- |
| نکته  اسم اشاره یک اشاره گر است که به خانه صفرم آرایه اشاره می کند . |
| |  | | --- | |  | | 4 | | 4 | | 20 10 4 | |  | |  |     p  q  a  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 ;  q   |  | | --- | |  | | **316** | | **!** | | **4** | |  |   a = 3 ;  b =4 ;  a  p = & b ;  16 آدرس 1  a = ( \* p ) \* ( \* q ) ;  p  p = & a ;  16 آدرس 2  q = p ;  cout << p << \* p << endl ;  cout <<q << \* q ;   |  | | --- | | 56 | | 3 |   p  a  1000  2000  b  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 ؟  Kecursive binary searchجستجوی دودویی بازگشتی |
| 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 ()   |  |  | | --- | --- | | 1 | 1 | | 2 | 2 | | 3 | 3 | | 4 | 4 |   {  Static int q=1 ;  cout << q++<<”\n “ ;  return q ;  }  Main ()  {  cout << s () << “ \ n “ ;  cout << s () ;  } |

|  |
| --- |
| 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 ;   |  | | --- | | **اگر در تابعی متغیری هم نام یک متغیر سراسری تعریف کنیم برای دسترسی به متغییر سراسری از علامت ( : : ) استفاده می کنیم .** |   a=b=2 ;  void main ()  {  int a , b ;  a = a = 1 3 2 4 4  a = b + 1  a = : : b + 1  Global متغیر  Local نه متغیر |

|  |
| --- |
| نکته:  (Over Loding ) |
| تعریف توابع هم نام  int add ( int a , int b )  {  }    int add ( float c , int d )  {  } |

|  |
| --- |
| 5  مقدار ( 5 و 3 ) what 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 )  } |
| 122  What (3 , 5 )  Return  Z \* what  ( 2 , 3 )  Return  Z \* what  ( 1 ,1 )  6  What ( 2 , 3 )  What ( 1 , 1 )  Return  3 |