

چاپ اعداد زوج بین 1 الی 100

```
#include<iostream.h>
int main()
{
for(int i=2;i<=200;i+=2)
cout<<i<<endl;
return(0);
}
```

n را دریافت کرده و اعداد زوج بین 1 الی n را چاپ کند

```
#include<iostream.h>
int main()
{
int n;
cout<<"Please Enter a Number:";
cin>>n;
for(int i=2;i<n;i+=2)
cout<<i<<endl;
return(0);
}
```

دو عدد m, n را دریافت کرده و اعداد زوج مابین آنها چاپ کند n < m

```
#include<iostream.h>
int main()
{
int n,m;
cout<<"Please Enter two Number:\n";
cin>>n>>m;
cout<<endl;
for(int i=n;i<=m;i++)
if(i%2==0)cout<<i<<endl;
return(0);
}
```

برنامه ای که شعاع دایره را دریافت کرده و مساحت آنرا محاسبه کند

```
#include<iostream.h>
int main()
{
float r;
cout<<"Please Enter r of circle:";
cin>>r;
cout<<"the S="<<3.14*r*r;
return(0);
}
```

2 عدد را دریافت کرده واولی را به توان دومی برساند

```
#include<iostream.h>
main()
{
int n,m;
long int result=1;
cout<<"Please enter (n,m) n<m \n";
cin>>n>>m;
cout<<endl;
for(int i=1;i<m;i++)
result*=n;
cout<<"The result="<<result;
}
```

یک عدد را دریافت کرده و کلیه مقسوم علیه های آنرا چاپ کند.

```
#include<iostream.h>
main()
{
int n;
cout<<"Please Enter a number:";cin>>n;
for(int i=1;i<=n;i++)
{
if(n%i==0)
{cout<<i;cout<<endl;}
}
return (0);
}
```

یک عدد را دریافت کرده و مشخص کند عدد اول است یا خیر

```
#include<iostream.h>
main()
{
int n;
int isavval=1;
cout<<"Please Enter a number:";cin>>n;
if(n==2) cout<<"number "<<n<<" is avval";

for(int i=2;i<n;i++)
{
if(n%i==0){isavval=0; break;}
}

if (isavval==1)
cout<<"the number "<<n<<" is avval";
else
cout<<"the number "<<n<<" isnot avval";
return (0);
}
```

یک عدد را دریافت کرده و مشخص کند عدد کامل است یا خیر

```
#include<iostream.h>
main(){
int n;
int sum=0;
cout<<"Please Enter a number:";cin>>n;

for(int i=1;i<n;i++)
{
if(n%i==0){sum+=i;}
}

if(sum==n)
cout<<"the number "<<n<<" is kamel";
else
cout<<"the number "<<n<<" isnot kamel";
return (0);}
```

سه عدد را دریافت کرده و مینیمم و ماکزیمم آن را چاپ کند؟

```
#include<iostream.h>
main(){
long int a,b,c,max=0,min=0;

cout<<"Please Enter number a:";cin>>a;
cout<<"Please Enter number b:";cin>>b;
cout<<"Please Enter number c:";cin>>c;
cout<<endl;
if(a<b){min=a;max=b;}
else {min=b;max=a;}
if(c>max) max=c;
if(c<min) min=c;

cout<<"the max number is:"<<max<<endl;
cout<<"the min number is:"<<min;

return (0);}
```

n را دریافت کرده و خروجی زیر را چاپ کند.

	1	1	1
	2	4	8
#include<iostream.h>	3	9	27
main(){	4	16	64
long int n;	n → 5	25	125

```
cout<<"Please Enter The number:";cin>>n;
cout<<endl;
cout<<"Tavan1 Tavan2 Tavan3"<<endl;
cout<<"-----"<<endl;

for(int i=0;i<=n;i++)
cout<<" "<<i<<" " <<i*i<<" " <<i*i*i<<endl;

return (0);}
```

n عدد را دریافت کرده و میانگین و ماکزیمم و مینیمم آنرا چاپ کند.

```
#include<iostream.h>
main(){
int n;
long int a,max=0,min=0;
float mod=0;
cout<<"How many number do you want enter:";
cin>>n;
cout<<endl;
for(int i=1;i<=n;i++)
{
cout<<"Please Enter number"<<i<<":";cin>>a;
mod+=a;
if(a>max) max=a;
if(min==0) min=a;
if(a<min) min=a;
}

cout<<"The Max number:"<<max<<endl;
cout<<"The Min number:"<<min<<endl;
cout<<"The Mod number:"<<mod/n<<endl;

return (0);}
```

n را دریافت کرده و فاکتوریل آنرا محاسبه کند.

```
#include<iostream.h>
main(){
int n,fact=1;
cout<<"Please Enter The number:";cin>>n;
cout<<endl;

for(int i=1;i<=n;i++) fact*=i;

cout<<n<<"!="<<fact;
return (0);}
```

n را دریافت کرده و خروجی زیر را چاپ کند.

	1	1
	2	2
	3	6
	4	24
	n → 5	120

```
#include<iostream.h>
main(){
int n;
long int fact=1;
cout<<"Please Enter The number:";cin>>n;
cout<<endl;
cout<<"Number Factoril "<<endl;
cout<<"----- "<<endl;

for(int j=1;j<=n;j++)
{
for(int i=1;i<=j;i++) fact*=i;

cout<<j<<" " <<fact<<endl;
fact=1;
}

return (0);}
```

←→

n را دریافت کرده و خروجی زیر را چاپ کند.

```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5 ← n

```

```

#include<iostream.h>
main(){
int n;
cout<<"Please Enter a Number:";cin>>n;
cout<<endl;
for(int j=1;j<=n;j++){
{
for(int i=1;i<=j;i++) cout<<" "<<i;
cout<<endl;
}
}
return (0);}

```

←→

n را دریافت کرده و خروجی زیر را چاپ کند.

```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5 ← n

```

```

#include<iostream.h>
main(){
int n;
cout<<"Please Enter a Number:";cin>>n;
cout<<endl;
int j=1,i=1;
while(j<=n)
{
while(i<=j)
{cout<<" "<<i;i+=1;}
cout<<endl;
j+=1;i=1;
}
return (0);}

```

5

←→

n را دریافت کرده و خروجی زیر را چاپ کند.

$$\frac{1^2}{1!} + \frac{2^2}{2!} + \frac{3^2}{3!} + \dots + \frac{n^2}{n!}$$

```

#include<iostream.h>
main(){
long int n,i;
long double fact=1;
double sum=0;
cout<<"PLease enter a number:";cin>>n;

for(int j=1;j<=n;j++){
fact=1;
i=1;
for(int i=1;i<=j;i++){
fact*=i;}
sum+=(j*j)/fact;
}
cout<<"the sum="<<sum<<endl;
return (0);}

```

$ax^2+bx+c=0$

←→

حل معادله درجه 2

```

#include<iostream.h>
#include<math.h>
int main(){
int a,b,c;
long int delta;
long double x1,x2;
cout<<"Please enter number a:";cin>>a;
cout<<"Please enter number b:";cin>>b;
cout<<"Please enter number c:";cin>>c;
delta=(b*b)-(4*a*c);
if (delta<0){
cout<<"there isn't any answer"<<endl;
}
if(delta==0){
cout<<"there is only one answer"<<endl;
cout<<"X="<<-b/(2*a);
}
if(delta>0){
cout<<"there are two answer"<<endl;
cout<<"X1="<<(-b+sqrt(delta))/(2*a);cout<<endl;
cout<<"X2="<<(-b-sqrt(delta))/(2*a);cout<<endl;
}
return(0);
}

```

6

Switch case

```
#include <iostream.h>
int main() {
    cout << "What do you want to do:\n";
    cout << "1. Calculate the area of a square\n";
    cout << "2. Calculate the area of a circle\n";
    cout << "Your choice: ";
    int choice;
    cin >> choice;
    switch (choice) {

        case 1: {
            cout << "Please enter the side length: ";
            double side;
            cin >> side;
            if (side < 0)
                cout << "There can be no squares with "
                << "negative side lengths. Bye.\n";
            else
                cout << "The area is " << side * side
                << ".\n";
            break;}

        case 2: {
            cout << "Please enter the radius: ";
            double radius;
            cin >> radius;
            if (radius < 0)
                cout << "There are no circles with "
                << "negative radiuses. See you.\n";
            else
                cout << "The area is "
                << radius * radius * 3.1415926
                << ".\n";
            break;}

        default:
            cout << "Your selection isn't valid.\n";
            break;}

    }
}
```

While

```
#include <iostream.h>
int main() {
    int i = 0;

    while (i < 10) {
        cout << i << " ";
        i = i + 1;}
}
```

0 1 2 3 4 5 6 7 8 9

for

```
#include <iostream.h>
int main() {
    for (int i = 0; i < 10; ++i) {
        cout << i << " ";
    }
}
```

0 1 2 3 4 5 6 7 8 9

do . . . while

```
#include <iostream.h>
int main() {
    int i;
    // Keep asking until the number is greater than 10
    do {
        cout << "Please enter a number greater than "
        << "10:";
        cin >> i;
    } while (i > 10);
    cout << "You entered: " << i << "\n";
}
```

break and continue

```
#include <iostream.h>
int main() {
    cout << "Enter 10 positive numbers, or a "
    <<" negative number to abort.\n";
    // Notice that here we declare i *outside* of the
    // loop. You'll see later why.
    int i;
    // Here we start counting at 1 and not at 0 because
    // otherwise the program would ask for number #0,
    // then for number #1, but we want it to start at
    // 1.
    for (i = 1; i <= 10; ++i) {
        cout << "Enter the number #" << i << ": ";
        int n;
        cin >> n;
        if (n < 0)
            break;
    }
    // If we hadn't declared i outside the loop, we
    // couldn't access it here because it'd be out of
    // scope.
    if (i == 11)
        cout << "You are a real man.\n";
    else
        cout << "You stopped after " << i
        << " numbers, coward!\n";
}
```

// Print all numbers from 0 to 100 except the multipl of 7.

```
#include <iostream.h>
int main() {
    for (int i = 0; i < 100; ++i) {
        // Skip all the multiples of 7
        // Operator % calculates the modulo (remainder)
        // of a division; if (i % 7) is equal to zero
        // this means that i is a multiple of 7.
        if ((i % 7) == 0)
            continue;
        cout << i << " ";
    }
}
```

The other statement, continue, is slightly more complicated but just as useful. It skips all the rest of the statements inside the loop and goes immediately to the next iteration. It is often used to skip the execution if a certain condition is met. Have a look at the following program which prints the numbers from 0 to 100 except the multiples of 7:

9

```
// This program prints out a multiplication table as
// follows:
// 1 2 3
// 2 4 6
// 3 6 9
#include <iostream.h>
int main() {
    for (int y = 1; y <= 3; ++y) {
        for (int x = 1; x <= 3; ++x) {
            cout << x * y << " ";
        }
        cout << "\n";
    }
}
```

```
#include<iostream.h>
int main() {
    cout << "This program calculates the factorial "
    << "of n.\n";
    cout << "Please enter n: ";
    int n;
    cin >> n;
    if (n < 0)
        cout << "n must be non-negative.\n";
    else {
        long double factorial = 1;
        for (int i = 1; i <= n; ++i)
            factorial = factorial * i;
        cout << n << "! = " << factorial << "\n";
    }
}
```

a program asking the user for the upper and the lower bound and then displaying all the numbers between them

```
#include <iostream.h>

int main() {
    cout << "Lower bound: ";
    int lower;
    cin >> lower;
    cout << "Upper bound: ";
    int upper;
    cin >> upper;
    cout << "Step size: ";
    int step;
    cin >> step;
    if (step > 0) {
        for (int i = lower; i < upper; i = i + step)
            cout << i << " ";
    }
    else
        cout << "Step size must be positive.\n";
}
```

Functions

```
#include <iostream.h>
#include <math.h>
//using namespace std;
int tavan2(int n) {
return n * n;
}
int main() {
int num;
do {
cout << "What do you want to know the tavan2 "
<< "of (enter 0 to quit): ";
cin >> num;
cout << "The tavan2 of " << num << " is "
<< tavan2(num) << ".\n";
} while (num != 0);
}
```

The Declaration

```
// A silly function which always returns 3
int three() {
return 3;
}

// Calculates the average of 2 numbers
double average(double n1, double n2) {
return (n1 + n2) / 2;
}
// Returns the string s n times
// For example mult_string("hello", 3) would return
// "hellohellohello".
string mult_string(string s, int n) {
string total;
for (int i = 0; i < n; ++i)
total = total + s;
return total;
}
```

Calling a Function

The function call is inside the body of the main function, and looks like this:

```
square(num)
But what does this do? It calls the function square, passing
it the value of num. As the function returns the square of n,
it behaves like some kind of variable; we could have
obtained the same effect writing
```

```
int sq = square(num);
cout << sq;
```

There is one thing to think of, though. Whenever we call the function square the code in its body is executed; thus

```
cout << square(num);
cout << square(num);
and
int sq = square(num);
cout << sq;
cout << sq;
```

void Functions

There are also functions with no return value. They are declared just as any other function, but with return type void, like this:

```
void display_help() {
cout << "This program is used to bla bla bla...";
}
```

Of course, you cannot return any value from a void function, but you can use the return statement anyway, just with no arguments. It is used to exit the function. It is not compulsory that you use return though, as it is in the nonvoid functions; if you don't, the function exits when it reaches the end of the body

```
#include <iostream.h>
```

```
// M is a global variable
int m;
int f(int n) {
// Here a *copy* of the argument is modified, *not*
// the object which was passed
n = n + 1;
return n;
}
int g() {
// Here the global variable m is increased
m = m + 1;
return m;
}
int main() {
// This n is local to this function, and *not* the
// same thing as the n in the function f.
int n = 5;
cout << "n = " << n << "\n";
cout << "f(n) = " << f(n) << "\n";
cout << "n = " << n << "\n";
// This is the global m declared at the beginning
// of the program. It is the same m which is
// modified in the function g.
m = 5;
cout << "m = " << m << "\n";
cout << "g() = " << g() << "\n";
cout << "m = " << m << "\n";
}
```

gives the output

```
n = 5
f(n) = 6
n = 5
m = 5
g() = 6
m = 6
```

int main()

You have surely noticed that `int main()` is nothing but a function. It is different from other functions, though. `main` is never called by your code. It is called automatically when the program starts. It is illegal to call it manually. (Something is said to be illegal if it is not allowed by the Standard, or, in other words, if it is an error.)

It is not necessary to put a return statement into `main`. In fact, there are only two values you should return from `main`: either `EXIT_SUCCESS` or `EXIT_FAILURE`. You return `EXIT_SUCCESS` if your program terminated successfully, or `EXIT_FAILURE` if it didn't. (For example, if you write a program to count the words in a text file, it might return `EXIT_FAILURE` if it cannot open the file.) If you do not return anything, the compiler returns `EXIT_SUCCESS` for you.

You are also allowed to return 0 instead of `EXIT_SUCCESS`. If you want to use `EXIT_SUCCESS` and `EXIT_FAILURE` you must `#include <cstdlib.h>`.

Pass-by-Value

In C++ arguments are passed by value. This means that if you pass a function an argument you do not pass it the object itself, only a copy of it. This means that a function cannot modify an argument you gave it. For example, this program

`n` was not modified in the function `f`.

```
#include <iostream.h>
```

```
void f(int n) {  
    n = 4;  
}  
int main() {  
    int m = 1;  
    cout << m << "\n";  
    f(m);  
    cout << m << "\n";  
}
```

Recursive Functions

```
void f() {  
    f();  
}
```

Of course, that function would never exit once it started, but continue running until the computer runs out of memory or something else bad happens. A recursive function should thus always have some condition to terminate. Look at the following program which uses a recursive function to print a sequence of numbers:

```
13 #include <iostream.h>
```

```
void recursive_function(int start, int end) {  
    if (start < end) {  
        cout << start << " ";  
        recursive_function(start + 1, end);  
    }  
    int main() {  
        recursive_function(1, 10);  
    }
```

It output is
1 2 3 4 5 6 7 8 9

This should be quite simple. It starts getting more complicated if we add one line to the function, as follows:

```
#include <iostream.h>
```

```
void recursive_function(int start, int end) {  
    if (start < end) {  
        cout << start << " ";  
        recursive_function(start + 1, end);  
        cout << start << " ";  
    }  
    int main() {  
        recursive_function(1, 10);  
    }
```

Now the output is — hold your breath — this one:
1 2 3 4 5 6 7 8 9 9 8 7 6 5 4 3 2 1

You probably expected something along the lines of 1 1 2 2 3 3 ..., didn't you? If you knew what'd happen, congratulations. Otherwise, read this explanation.

In a recursive function, at some time there is more than one instance of the same function. Each of these instances has its own local variables, which do not affect the variables of the other instances. The call `recursive_function(start + 1, end)` inside the function returns only when it has finished executing; thus something along the lines of the following happens:

```
recursive_function(1, 10) is called  
cout << 1;  
recursive_function(2, 10) is called  
cout << 2;  
recursive_function(3, 10) is called  
cout << 3;  
recursive_function(4, 10) is called  
cout << 4;  
...  
recursive_function(10, 10) is called  
The condition is false, thus this does NOTHING  
...
```

Although recursive functions are not too widely used because they are generally error-prone and not very efficient, it is good to know that they exist because some problems can be solved much easier with them than without, especially if speed and stability is not so important.

Functions are used by the dozens, hundreds or even thousands in every "real" program; it is thus easy to come up with examples. They are both used to store repeated code and to split functions which would otherwise be too long to understand and maintain

```
#include <iostream.h>
```

```
// Print a line of '-' to divide one average from the
// other.
```

```
void delimiter() {
for (int i = 0; i < 79; ++i)
cout << "-";
cout << "\n";
}
```

```
// Ask for n numbers and calculate their average.
```

```
void average(int n) {
double sum = 0;
for (int i = 1; i <= n; ++i) {
```

```
// Ask for a number
```

```
cout << "Enter number " << i << " of " << n
<< ": ";
double num;
cin >> num;
```

```
// Add the number to the sum
```

```
sum = sum + num;
}
```

```
// Print the average
```

```
cout << "The average is: " << sum / n << ".\n";
}
```

```
int main() {
```

```
// A for(;;) loop repeats for e
```

```
// to jump out of the function directly
```

```
for (;;) {
cout << "How many numbers do you want to "
<< "calculate the average of (0 to exit): ";
int num;
cin >> num;
```

```
// Jump out of the function if the user entered
```

```
// zero
```

```
if (num == 0)
return 0;
```

```
// Do the average and display a delimiter (line)
```

```
average(num);
delimiter();
}
}
```

تابع محاسبه ماکزیمم دو عدد را بنویسید.

```
#include <iostream.h>
```

```
int getmax(int a ,int b)
```

```
{
if(a>b) return a;
else return b;
}
```

```
int main()
```

```
{
int d,e;
cout<<"please enter number 1:";cin>>d;
cout<<"please enter number 2:";cin>>e;
int max=getmax(d,e);
cout<<"The max between two number
is"<<max;
```

```
cout<<endl;
```

```
return 0;
```

```
}
```

تابع محاسبه ماکزیمم دو عدد را بنویسید.

```
#include <iostream.h>
```

```
void printmax(int a,int b)
```

```
{
if(a>b) cout<<a;
else cout<<b;
}
```

```
int main()
```

```
{
int d,e;
cout<<"please enter number 1:";cin>>d;
cout<<"please enter number 2:";cin>>e;
```

```
cout<<"The max between two number
is";printmax(d,e);
```

```
cout<<endl;
```

```
return 0;
```

```
}
```


تابعی بنویسید که **max** دو عدد دریافتی را محاسبه و چاپ کند

```
#include <iostream.h>
int getmax(int a ,int b)
{
    if(a>b) return a;
    else return b;
}

int main()
{
    int d,e;
    cout<<"please enter number 1:"<<cin>>d;
    cout<<"please enter number 2:"<<cin>>e;
    int max=getmax(d,e);
    cout<<"The max between two number
    is"<<max;

    cout<<endl;
    return 0;
}
```

تابعی بنویسید که **max** دو عدد دریافتی را محاسبه و چاپ کند

Use void

```
#include <iostream.h>

void printmax(int a,int b)
{
    if(a>b) cout<<a;
    else cout<<b;
}

int main()
{
    int d,e;
    cout<<"please enter number 1:"<<cin>>d;
    cout<<"please enter number 2:"<<cin>>e;

    cout<<"The max between two number
    is="<<printmax(d,e);

    cout<<endl;

    return 0;
}
```

```
#include <iostream.h>
#include <math.h>
void moadeleh(int a,int b,int c)
{
    long int x1,x2,delta;
    delta=(b*b)-(4*a*c);
    if (delta<0)
    {
        cout<<"there isn't any answer"<<endl;
    }
    if(delta==0)
    {
        cout<<"there is only one answer"<<endl;
        x1=-b/(2*a);
        cout<<"X="<<x1;
    }
    if(delta>0)
    {
        cout<<"there are two answer"<<endl;
        x1=(-b+sqrt(delta))/(2*a);
        x2=(-b-sqrt(delta))/(2*a);
        cout<<"X1="<<x1;cout<<endl;
        cout<<"X2="<<x2;cout<<endl;
    }
}

int main()
{
    int a,b,c;
    cout<<"Please enter number a:"<<cin>>a;
    cout<<"Please enter number b:"<<cin>>b;
    cout<<"Please enter number c:"<<cin>>c;
    moadeleh(a,b,c);
    return(0);
}
```

تابع محاسبه فاکتوریل اعداد بین 1 الی 100 را بنویسید

```
#include<iostream.h>
void getfact(int n){
    long double factorial = 1;
    for (int i = 1; i <= n; ++i)
        factorial *= i;
    cout<<factorial;}

int main(){
    cout << "This program calculates the factorial "
    << "of number.\n";
    cout << "Please enter number: ";
    int n;
    cin >> n;
    if (n < 0)
        cout << "number must be non-negative.\n";
    else {
        cout << n << "! = ";getfact(n);cout<<endl;
    }
}
```

تابعی بنویسید که سه عدد صحیح را دریافت کرده و ماکزیمم آنها را برگرداند

```
#include<iostream.h>
void getmax(int a,int b,int c){
int max=a;
    if (b>max) max=b;
    if (c>max) max=c;
cout<<max;}

int main(){
int a,b,c;
cout<<"This program calculate Max Of 3 number\n";
cout<<"Please enter number 1=";cin>>a;
cout<<"Please enter number 2=";cin>>b;
cout<<"Please enter number 3=";cin>>c;

cout<<"The Max of 3 number=";getmax(a,b,c);
return 0;}
```

تابعی بنویسید که اعداد کامل بین 100 الی 1000 را مشخص کند

```
#include<iostream.h>
void kamel(int n){
int sum=0;
for(int i=1;i<n;i++)
{
if(n%i==0){sum+=i;}
}
if(sum==n)
cout<<"the number "<<n<<" is kamel";
else
cout<<"the number "<<n<<" isnot kamel";
}

main(){
int n;
cout<<"Please Enter a number:";cin>>n;
kamel(n);
return (0);}
```

تابعی بنویسید که عددی را دریافت کرده و اگر کامل بود عدد 1 و در غیر اینصورت عدد صفر را برگرداند

```
#include<iostream.h>
void kamel(int n){
int sum=0;
for(int i=1;i<n;i++)
{
if(n%i==0){sum+=i;}
}
if(sum==n)
cout<<"the number "<<n<<" is kamel";
else
cout<<"the number "<<n<<" isnot kamel";
}

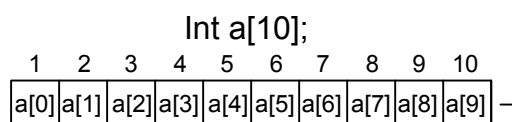
main(){
int n;

cout<<"Please Enter a number:";cin>>n;
kamel(n);

return (0);}
```

آرایه ها Array

Int=16bit Long int=32bit
Float=32bit Double=64bit
Char=8bit



اندیسها در زبان C از صفر شروع میشود.
به هر خانه یک درایه گفته میشود.

برنامه ای بنویسید که ده عدد را از ورودی گرفته و سپس از آخر به اول چاپ کند؟

```
#include <iostream.h>
int main(){
int a[9];
int n;
cout<<"Please enter 10 number=";
for (int i=0;i<10;i++)
{
cin>>n;
a[i]=n;
}
for (int j=9;j>=0;j--)
{
cout<<a[j]<<"\t";
}
return 0;
}
```

برنامه ای بنویسید که ده عدد را از ورودی گرفته و مینیمم و ماکزیمم آنر محاسبه و چاپ کند؟

```
#include <iostream.h>
int main(){
int a[9];
int n,min,max;
min=0;max=0;
cout<<"Please enter numbers:\n";
cin>>n;
a[0]=n;
max=n;min=n;
for (int i=1;i<10;i++)
{
cin>>n;
a[i]=n;
if (a[i]>max)
max=a[i];
if (a[i]<min)
min=a[i];
}
cout<<"the max="<<max;cout<<endl;
cout<<"the min="<<min;
return 0;
}
```

21

```
#include <iostream.h>
int main(){
int a[9];
int n;
cout<<"Please enter 10 number\n";
for (int i=0;i<10;i++)
{
cin>>n;
a[i]=n;
}
for (int j=0;j<10;j++)
{
if (a[j]%2==1) ← odd
cout<<a[j]<<"\t";
}
cout<<endl;
for (int k=0;k<10;k++)
{
if (a[k]%2==0) ← even
cout<<a[k]<<"\t";
}
return 0;
}
```

برنامه ای بنویسید که ده عدد را از ورودی گرفته و سپس ابتدا اعداد فرد و سپس زوج را چاپ کند؟

برنامه ای بنویسید که ده عدد را از ورودی گرفته و مجموع کل آنها را محاسبه و چاپ کند؟

```
#include <iostream.h>
int main(){
int a[9];
int b[9];
int n,m,sum1,sum2,endsum;
sum1=0;sum2=0;
cout<<"Please enter 10 number for first array:\n";
for (int i=0;i<10;i++)
{
cin>>n;
a[i]=n;
sum1 +=a[i];
}
cout<<"Please enter 10 number for second array:\n";
for (int j=0;j<10;j++)
{
cin>>m;
b[j]=m;
sum2 +=b[j];
}
endsum=sum1+sum2;
cout<<endl;
cout<<"The sum of array 1&2="<<endsum;
return 0; }
```

22

Bay

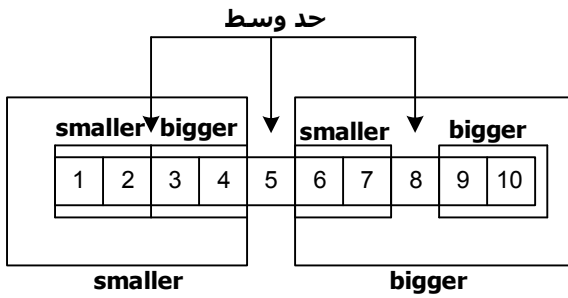
برنامه ای بنویسید که ده عدد را از ورودی گرفته و سپس ابتدا اعداد فرد و سپس زوج را و تعداد اعداد فرد دریافت شده و بعد زوج دریافت شده ، چاپ کند ؟

```
#include <iostream.h>
int main(){
int a[9];
int n,f,z;
f=0;z=0;
cout<<"Please enter 10 number=";<<endl;
for (int i=0;i<10;i++)
{
cin>>n;
a[i]=n;
}
for (int j=0;j<10;j++)
{
if (a[j]%2==1)
{
cout<<a[j]<<"\t";
f++;
}
}
for (int k=0;k<10;k++)
{
if (a[k]%2==0)
{
cout<<a[k]<<"\t";
z++;
}
}
cout<<"fard="<<f<<"\n";
cout<<"zoj="<<z<<"\n";
return 0;
}
```

محاسبه فاکتوریل تا موقعی که عدد صفر وارد شود

```
#include<iostream.h>
int main()
{
int n;
do
{
cout<<"enter 0 to exit"<<"\n";
cout<<"please enter number=";
cin>>n;
int fact=1;
for(int i=1;i<=n;i++)
fact*=i;
cout<<n<<"!="<<fact<<endl;
}
while(n!=0);
return 0;
}
```

مرتب سازی و جستجو
Search and Sort



Binary Search
جستجوی دو دویی

اگر طول آرایه n باشد حداکثر n مقایسه انجام میدهیم، در حقیقت n واحد زمانی عمل مقایسه و جستجو انجام میگردد.

$$\frac{\frac{\frac{n}{2}}{2}}{2} = \text{Log}_2 n$$

در کامپیوتر بصورت پیش فرض لگاریتم در مبنای 2 میباشد

جستجو دودویی
در لیست مرتب شده

```
#include <iostream.h>
#include <conio.h>
int main()
{
int n=10;
int a[10],x;
int dow=0,up=n-1,mid;
for(int i=0;i<n;i++)
{cout<<"Enter number"<<i<<"=";
cin>>a[i];}
cout<<"Enter your search number=";cin>>x;
while(dow<=up)
{mid=(up+dow)/2;
if(a[mid]==x)
{cout<<"a["<<mid<<"]="<<a[mid];
break;}
if(a[mid]>x)
up=mid-1;
else
dow=mid+1;
}
if(dow>up)
cout<<"not find!";
getch();
return 0;
}
```

1	5	2	4	3	7	8	8	9	6
---	---	---	---	---	---	---	---	---	---

a[0]=min

برای مرتب کردن عدد اول را مینیمم میکنیم و با بقیه اعداد مقایسه میشود

1	5	2	4	3	7	8	8	9	6
---	---	---	---	---	---	---	---	---	---

محاسبه زمان برای مرتب سازی (روش حبابی)

اولین بار =	n بار مقایسه
دولین بار =	n-1 بار مقایسه
سولین بار =	n-2 بار مقایسه
⋮	⋮
⋮	⋮
آخرین بار =	1 بار مقایسه

$$\frac{n(n+1)}{2} = n+(n-1)+.....+1$$

مرتب سازی یک لیست ده عضوی

```
#include<iostream.h>
int main()
{
int a[10];
cout<<"enter 10 number"<<"\n";
for(int i=0;i<10;i++)
{cout<<"Enter number"<<i<<"=";
cin>>a[i];}
for(i=0;i<10;i++)
{
for(int j=i+1;j<10;j++)
{
if(a[j]<a[i])
{
int temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
for(i=0;i<10;i++)
cout<<a[i]<<"\t";
return 0;}
}
```

جستجو دودویی در لیست مرتب نشده

```
#include<iostream.h>
int main()
{
int a[10];
cout<<"enter 10 number"<<"\n";
for(int i=0;i<10;i++)
{cout<<"a["<<i<<"]="";
cin>>a[i];}
for(i=0;i<10;i++)
{
for(int j=i+1;j<10;j++)
{
if(a[j]<a[i])
{
int temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
cout<<"The list after sorting"<<endl;
for(i=0;i<10;i++)
cout<<"a["<<i<<"]="<<a[i]<<"\n";
int n=10;
int x;
int dow=0,up=n-1,mid;

cout<<"Enter your search number=";cin>>x;
while(dow<=up)
{mid=(up+dow)/2;
if(a[mid]==x)
{cout<<"a["<<mid<<"]="<<a[mid];
break;}
if(a[mid]>x)
up=mid-1;
else
dow=mid+1;
}
if(dow>up)
cout<<"not find!";
return 0;
}
```

دریافت و چاپ آرایه های دوبعدی

```
#include<iostream.h>
int main()
{
int a[3][3];
cout<<"The inserting array"<<"\n";
for(int i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
cout<<"a["<<i<<"]="<<["<<j<<"]="";
cin>>a[i][j];
}
}
cout<<"The printing array"<<"\n";
for( i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
cout<<"a["<<i<<"]="<<["<<j<<"]="<<a[i][j];
cout<<endl;
}
}
return 0;
}
```

برنامه محاسبه حاصلضرب دو ماتریس 3*3 و 3*2 و 2*3

```
#include<iostream.h>
int main()
{
int a[3][2],b[2][3],c[3][3];
cout<<"The inserting array A"<<"\n";
for(int i=0;i<3;i++)
{
for(int j=0;j<2;j++)
{
cout<<"a["<<i<<"]"<<"["<<j<<"]="";
cin>>a[i][j];
}
}
cout<<"The inserting array B"<<"\n";
for(i=0;i<2;i++)
{
for(int j=0;j<3;j++)
{
cout<<"b["<<i<<"]"<<"["<<j<<"]="";
cin>>b[i][j];
}
}
cout<<"The computing array C"<<"\n";
for(i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
int sum=0;
for(int k=0;k<2;k++)
{sum+=a[i][k]*b[k][j];}
c[i][j]=sum;
}
}
cout<<"The printing array C"<<"\n";
for( i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
cout<<"c["<<i<<"]"<<"["<<j<<"]="<<c[i][j];
cout<<endl;
}
}
return 0;
}
```

ماتریس تنک : Spare

ماتریسی که درایه های صفر آن نسبت به غیر صفر آن بیشتر است حاصلضرب دو ماتریس تنک همیشه تنک نیست

$$\begin{bmatrix} 5 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

ذخیره سازی ماتریس تنک:

$$\begin{bmatrix} 5 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \longrightarrow \begin{array}{cc} \text{مقدار} & \text{ستون} & \text{سطر} \\ \begin{bmatrix} 1 & 1 & 5 \\ 1 & 2 & 2 \end{bmatrix} \end{array}$$