

Programs

Prepare a C++ program to input an array and print it in the ascending order

```
#include<iostream.h>
void main()
{
    int i, j, a[10], big, m;

    cout <<"\nEnter the length of the array. \n";
    cin >>m;

    cout <<"\n Enter the elements of the array. \n";
    for(i = 0; i < m; i++)
        cin >>a[i];
```

```
for(i = 0; i < m - 1; i++)
    for(j = i + 1; j < m; j++)
    {
        if (a[i] > a[j])
        {
            big=a[i];
            a[i]=a[j];
            a[j]=big;
        }
    }

cout <<"\n The resulting array in ascending order is: \n";
for(i = 0; i < m; i++)
    cout <<a[i] <<"\n";
}
```

```
#include<iostream.h>
void main()
{
    int i, j, ra, ca;
    float a[10][10], trana[10][10];

    cout <<"\nEnter the size of the matrix.\n";
    cin >>ra >>ca;

    cout <<"\nEnter the elements of the matrix.\n";
    for(i = 0; i < ra; i++)
        for(j = 0; j < ca; j++)
            cin >>a[i][j];

    for(i = 0; i < ra; i++)
        for(j = 0; j < ca; j++)
    {
        trana[j][i] = a[i][j];
    }
}
```

Prepare a C++ program to print the transpose of a given matrix.

```
cout <<"\n The transpose of the matrix is:\n";
for(i = 0; i < ca; i++)
{
    for(j = 0; j < ra; j++)
        cout <<"\t" <trana[i][j];
    cout <<"\n";
}
```

```
#include<iostream.h>
#include<stdlib.h>
void main()
{
    int i,j,ra,ca;
    float a[10][10];
    cout <<"\nEnter the order of the square matrix.\n";
    cin >>ra >>ca;

    cout <<"\nEnter the elements of the matrix.\n";
    for(i = 0; i < ra; i++)
        for(j = 0; j < ca; j++)
            cin >>a[i][j];

    for(i = 0; i < ra; i++)
        for(j = 0; j < ca; j++)
            if(i == j)
            {
                if(a[i][j] != 1)
                {
                    cout <<"\n The given matrix is not an identity matrix.\n";
                    exit(0);
                }
            }
}
```

Prepare a C++ program to check whether the given square matrix of order m is an identity matrix.

```
else
{
    if(a[i][j] != 0)
    {
        cout <<"\n The given matrix is not an identity matrix.\n";
        exit(0);
    }
}
cout <<"\n The given matrix is an identity matrix.\n";
```

Prepare a C++ program to input an array of strings and print it in the alphabetic order.

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

void main()
{
    int i = 0, j = 0, m;
    char word[10][30], dummy[30];

    cout << "\nEnter the length of the array.\n";
    cin >> m;

    cout << "\nEnter the elements of the array.\n";
    for(i = 0; i < m; i++)
        cin >> word[i];
```

```
for(i = 0; i < m - 1; i++)
    for(j = i + 1; j < m; j++)
        if(strcmp(word[i], word[j]) > 0)
    {
        strcpy(dummy, word[i]);
        strcpy(word[i], word[j]);
        strcpy(word[j], dummy);
    }
cout <<"\n Sorted list: \n";
for(i = 0; i < m; i++)
    cout <<word[i] <<"\n";
}
```

Is num a palindrome???

Eg:- 12321

```
void main()
{
int num, rev;
int reverse(int);
cout<<“Enter the number”;
cin>>num;
rev=reverse(num);
if(rev==num)
    cout<<num<<“is a palindrome”
else
    cout<<num<<“is not a palindrome”;
}
```

```
int reverse(int n)
{
int s=0;
while(n)
{
    r=n%10;
    s=10*s+r;
    n=n/10;
}
return(s)
}
```

Recursion

Recursive function is a function that calls itself

```
int main(void)
{
    int n;
    cout<<"\nEnter an integer value: ";
    cin>>n;
    cout<<"\nThe factorial of "<< n << "is," << factorial(n);
    return 0;
}
```

```
long int factorial(int number)
{
    if(number <= 1)
        return 1;
    else
        return (number * factorial(number - 1));
}
```

```
#include<iostream.h>
void main()
{
    int org = 10;
    int change(int);
    cout << "\nThe original value is " << org << ".";
    cout << "\nThe value returned by function is " << change(org) << ".";
    cout << "\nThe value after function execution is " << org << ".";
}

int change(int a)
{
    a = a * a;
    return a;
}
```

Example for pass by value

/*
OUTPUT
The original value is 10.
The value returned by function is
100.
The value after function execution is
10.*/

