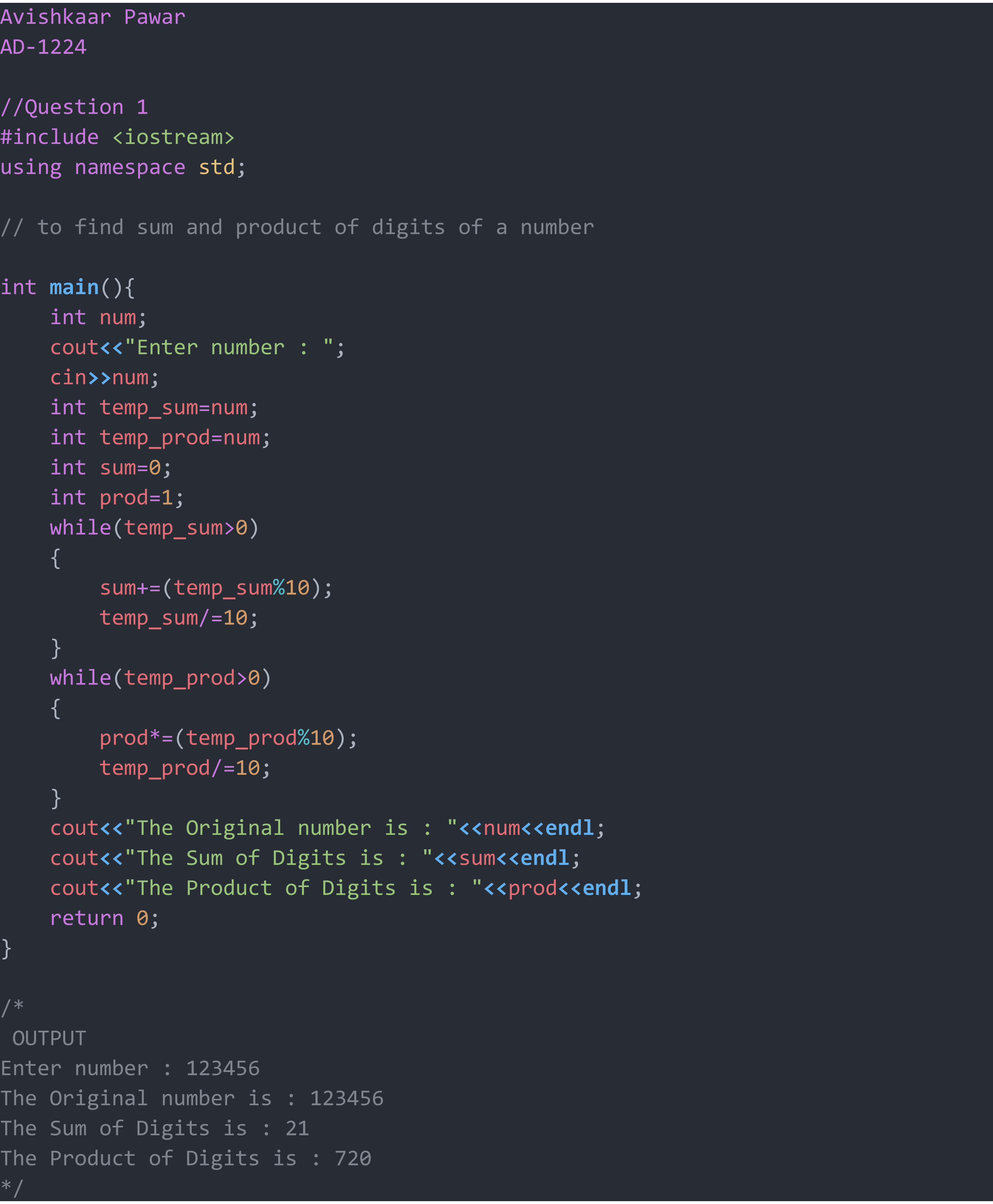
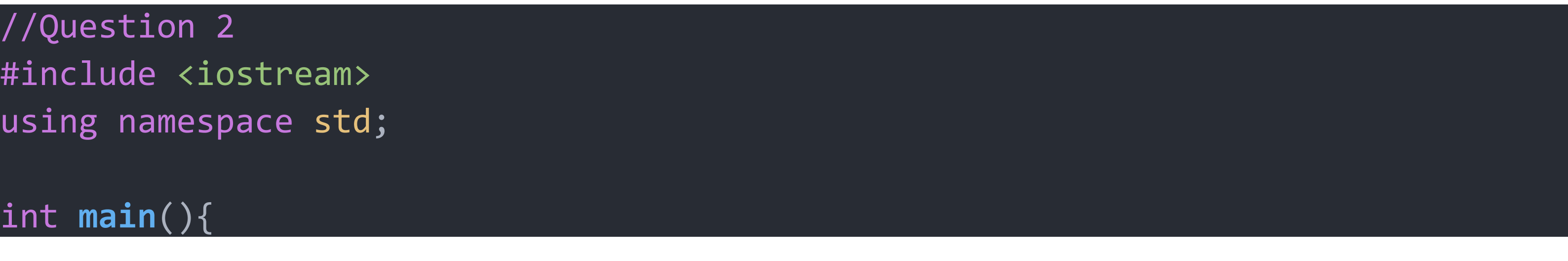
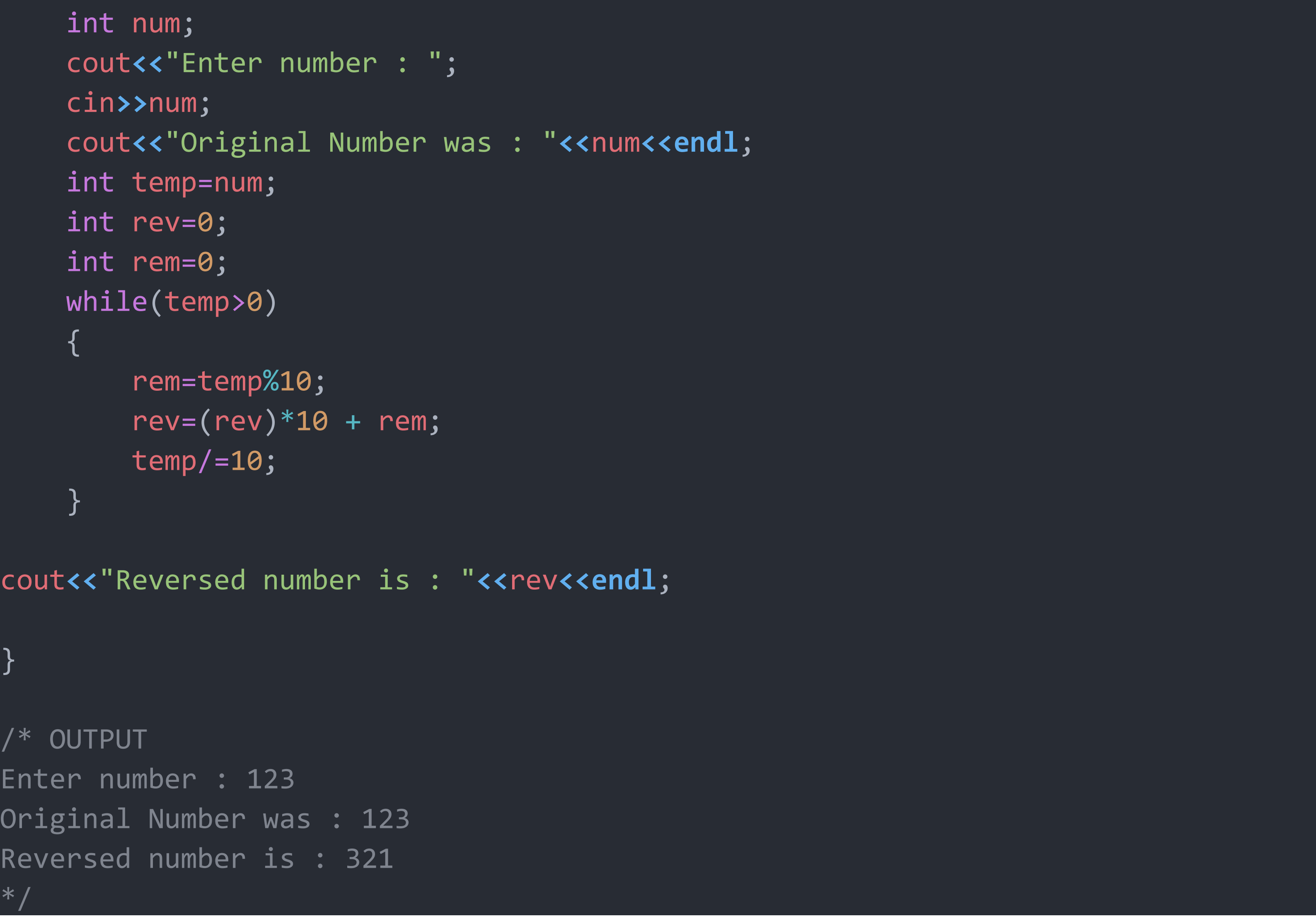


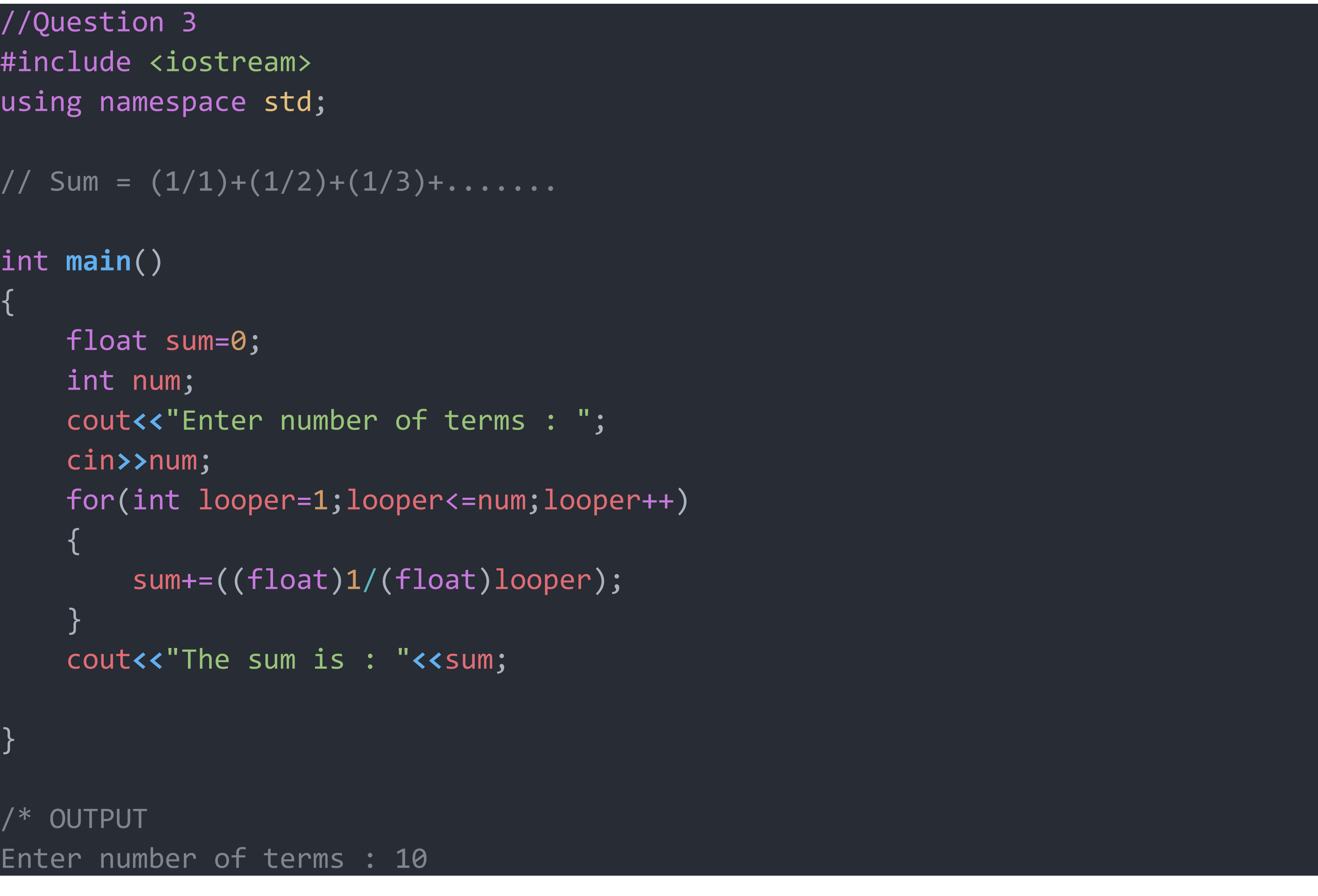
```
Avishkaar Pawar
AD-1224

//Question 1
#include <iostream>
using namespace std;

// to find sum and product of digits of a number

int main(){
    int num;
    cout<<"Enter number : ";
    cin>>num;
    int temp_sum=num;
    int temp_prod=num;
    int sum=0;
    int prod=1;
    while(temp_sum>0)
    {
        sum+=(temp_sum%10);
        temp_sum/=10;
    }
    while(temp_prod>0)
    {
        prod*=(temp_prod%10);
        temp_prod/=10;
    }
    cout<<"The Original number is : "<<num<<endl;
    cout<<"The Sum of Digits is : "<<sum<<endl;
    cout<<"The Product of Digits is : "<<prod<<endl;
    return 0;
}

/*
 OUTPUT
Enter number : 123456
The Original number is : 123456
The Sum of Digits is : 21
The Product of Digits is : 720
*/
```



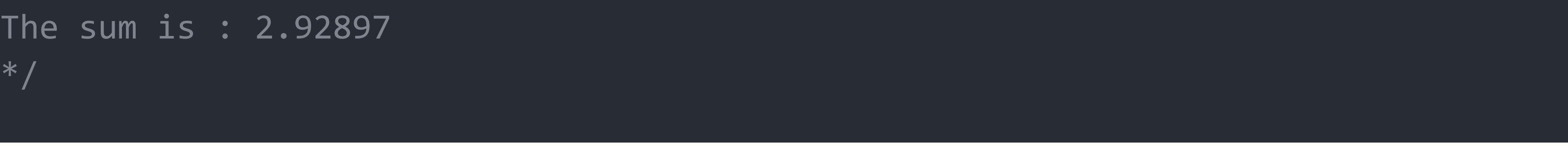
```
//Question 2
#include <iostream>
using namespace std;

int main(){
```



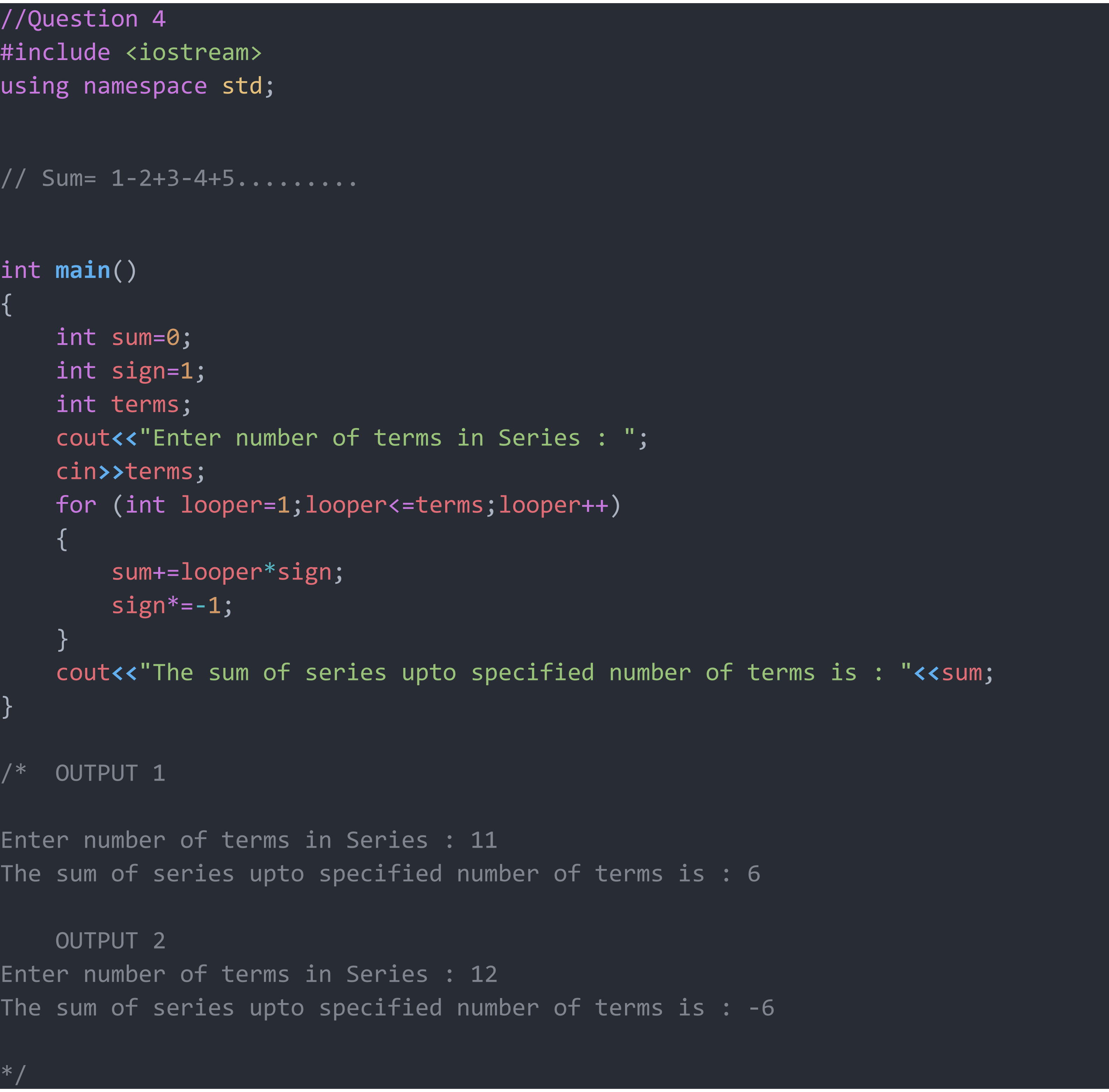
```
    int num;
    cout<<"Enter number : ";
    cin>>num;
    cout<<"Original Number was : "<<num<<endl;
    int temp=num;
    int rev=0;
    int rem=0;
    while(temp>0)
    {
        rem=temp%10;
        rev=(rev)*10 + rem;
        temp/=10;
    }

cout<<"Reversed number is : "<<rev<<endl;

}

/* OUTPUT
Enter number : 123
Original Number was : 123
Reversed number is : 321
*/
```



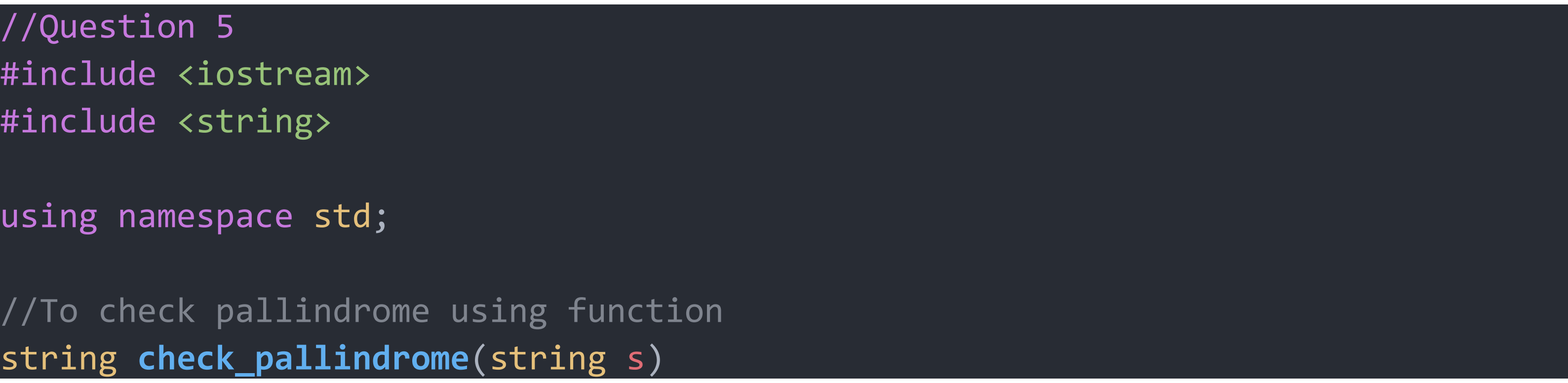
```
//Question 3
#include <iostream>
using namespace std;

// Sum = (1/1)+(1/2)+(1/3)+.......

int main()
{
    float sum=0;
    int num;
    cout<<"Enter number of terms : ";
    cin>>num;
    for(int looper=1;looper<=num;looper++)
    {
        sum+=((float)1/(float)looper);
    }
    cout<<"The sum is : "<<sum;

}

/* OUTPUT
Enter number of terms : 10
```



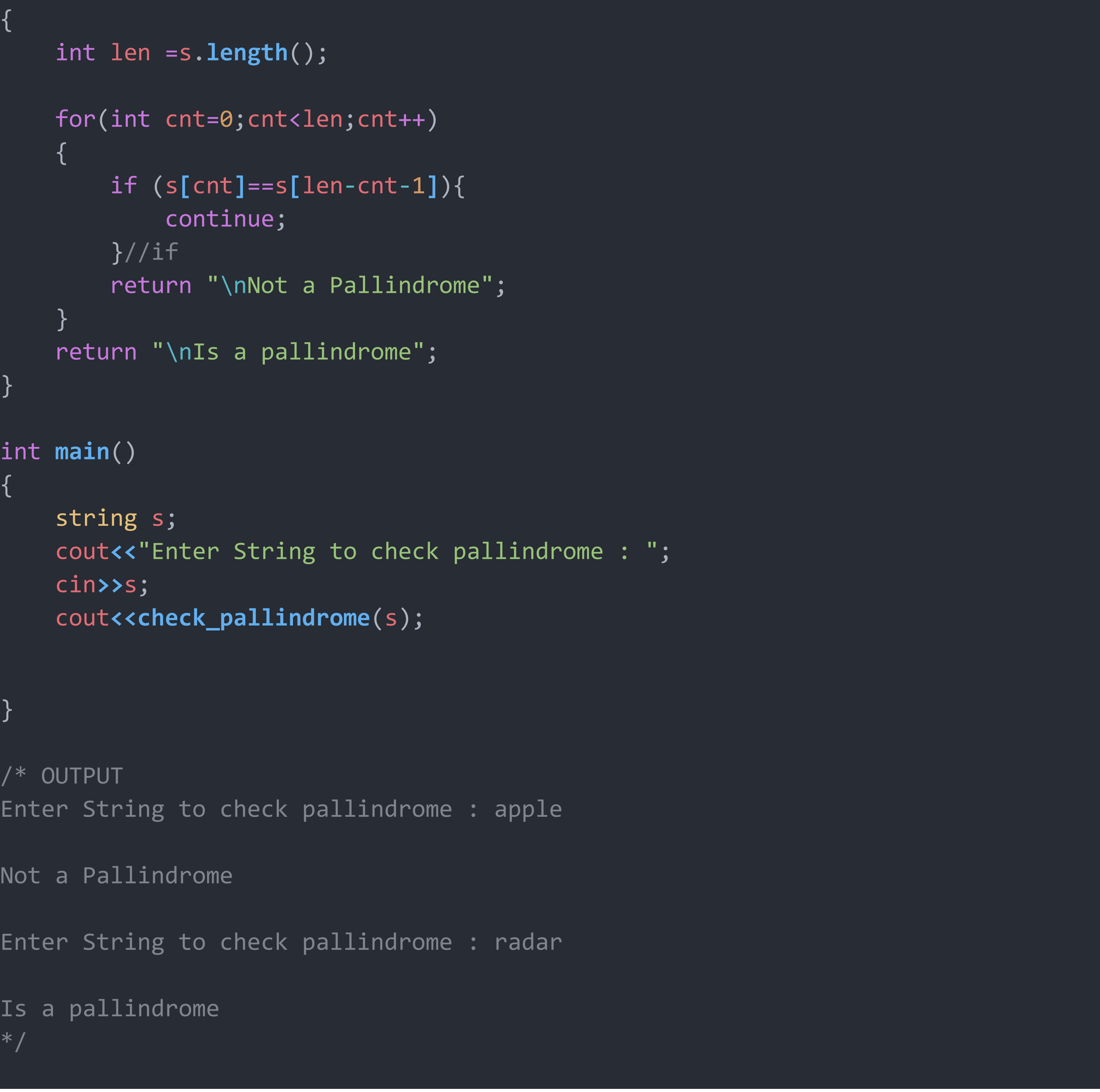
```
The sum is : 2.92897
*/
```



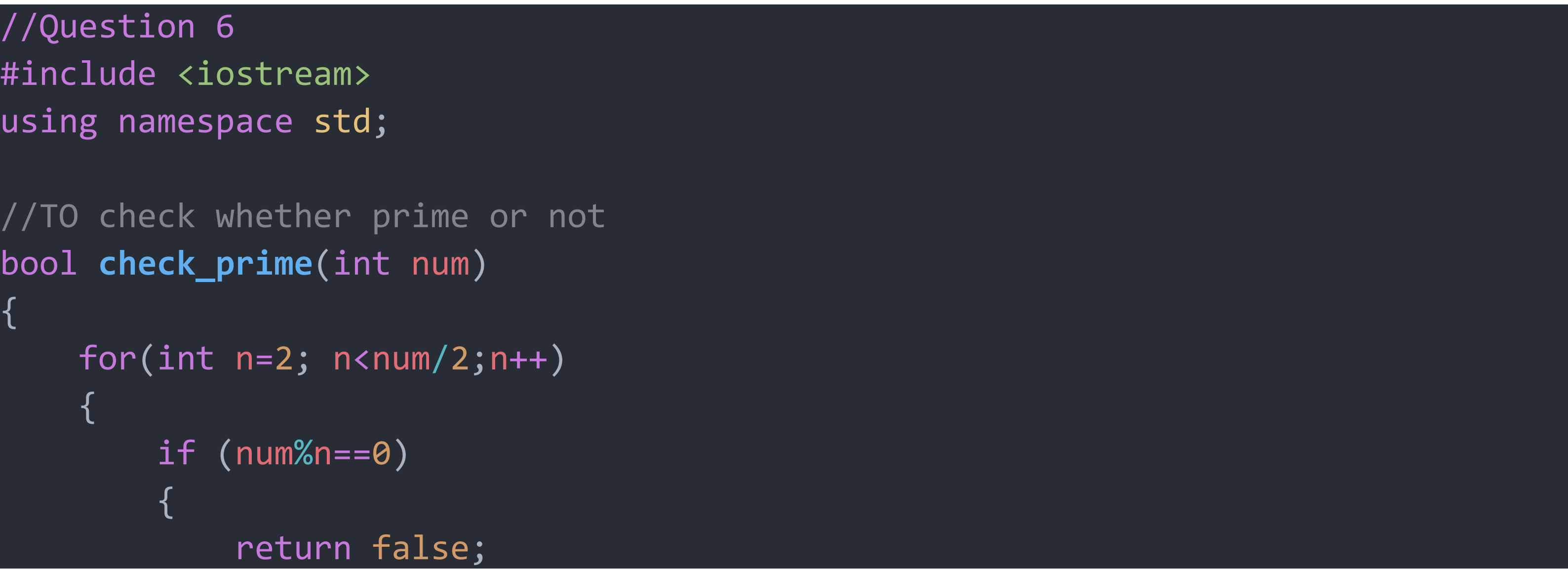
```
//Question 4
#include <iostream>
using namespace std;


// Sum= 1-2+3-4+5.........


int main()
{
    int sum=0;
    int sign=1;
    int terms;
    cout<<"Enter number of terms in Series : ";
    cin>>terms;
    for (int looper=1;looper<=terms;looper++)
    {
        sum+=looper*sign;
        sign*=-1;
    }
    cout<<"The sum of series upto specified number of terms is : "<<sum;
}

/*  OUTPUT 1

Enter number of terms in Series : 11
The sum of series upto specified number of terms is : 6

    OUTPUT 2
Enter number of terms in Series : 12
The sum of series upto specified number of terms is : -6

*/
```



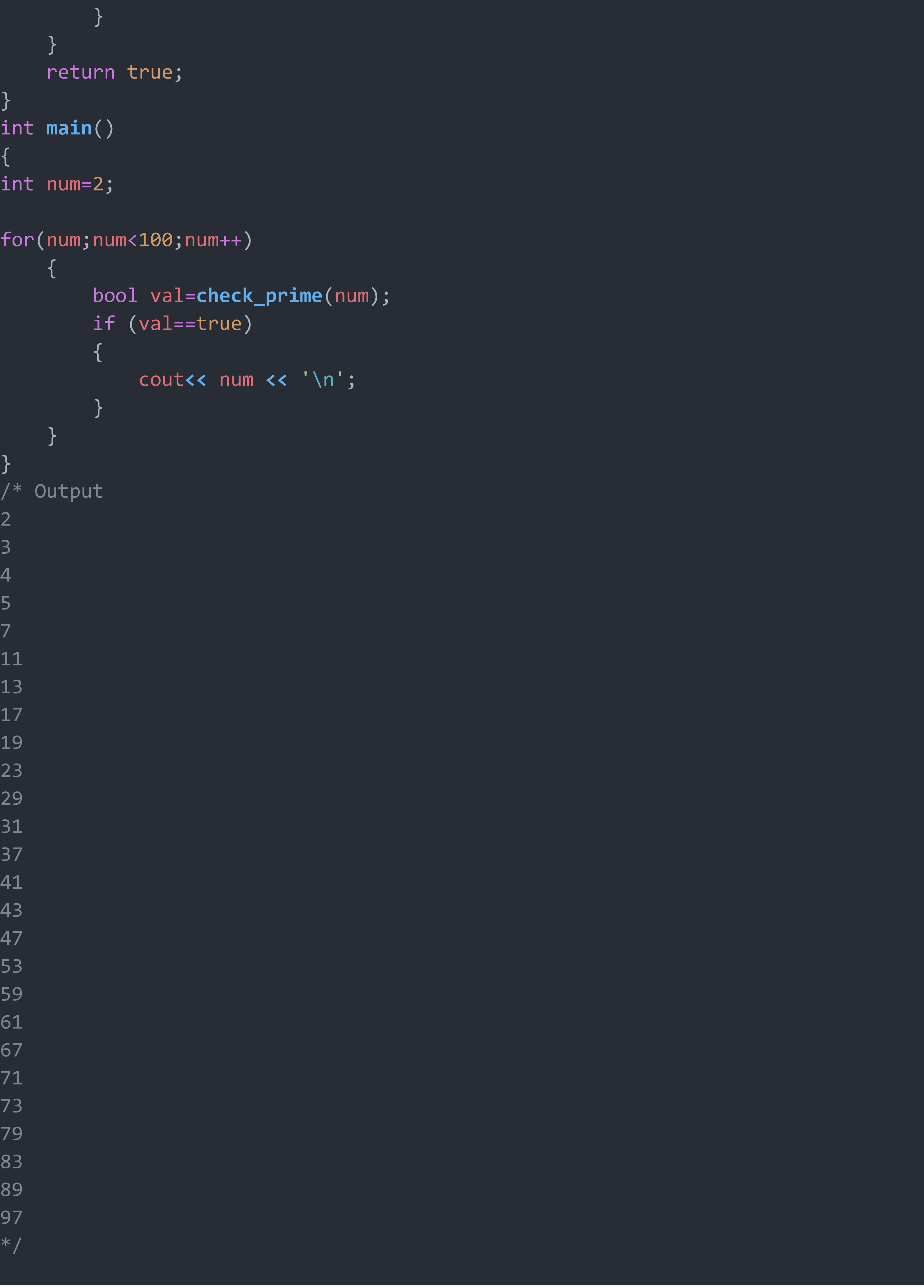
```
//Question 5
#include <iostream>
#include <string>

using namespace std;

//To check pallindrome using function
string check_pallindrome(string s)
```



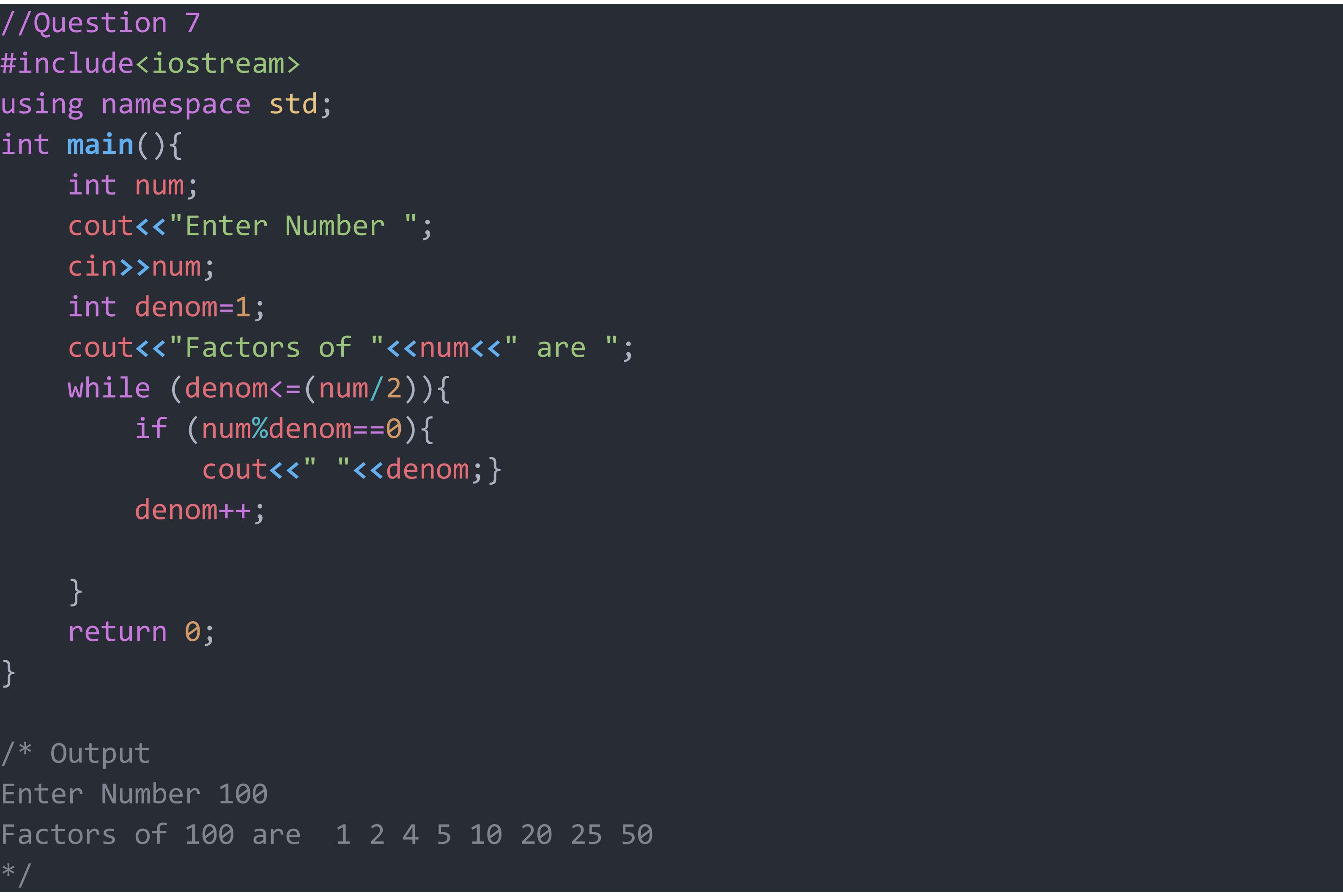
```
{
    int len =s.length();

    for(int cnt=0;cnt<len;cnt++)
    {
        if (s[cnt]==s[len-cnt-1]){
            continue;
        }//if
        return "\nNot a Pallindrome";
    }
    return "\nIs a pallindrome";
}

int main()
{
    string s;
    cout<<"Enter String to check pallindrome : ";
    cin>>s;
    cout<<check_pallindrome(s);


}

/* OUTPUT
Enter String to check pallindrome : apple

Not a Pallindrome

Enter String to check pallindrome : radar

Is a pallindrome
*/
```



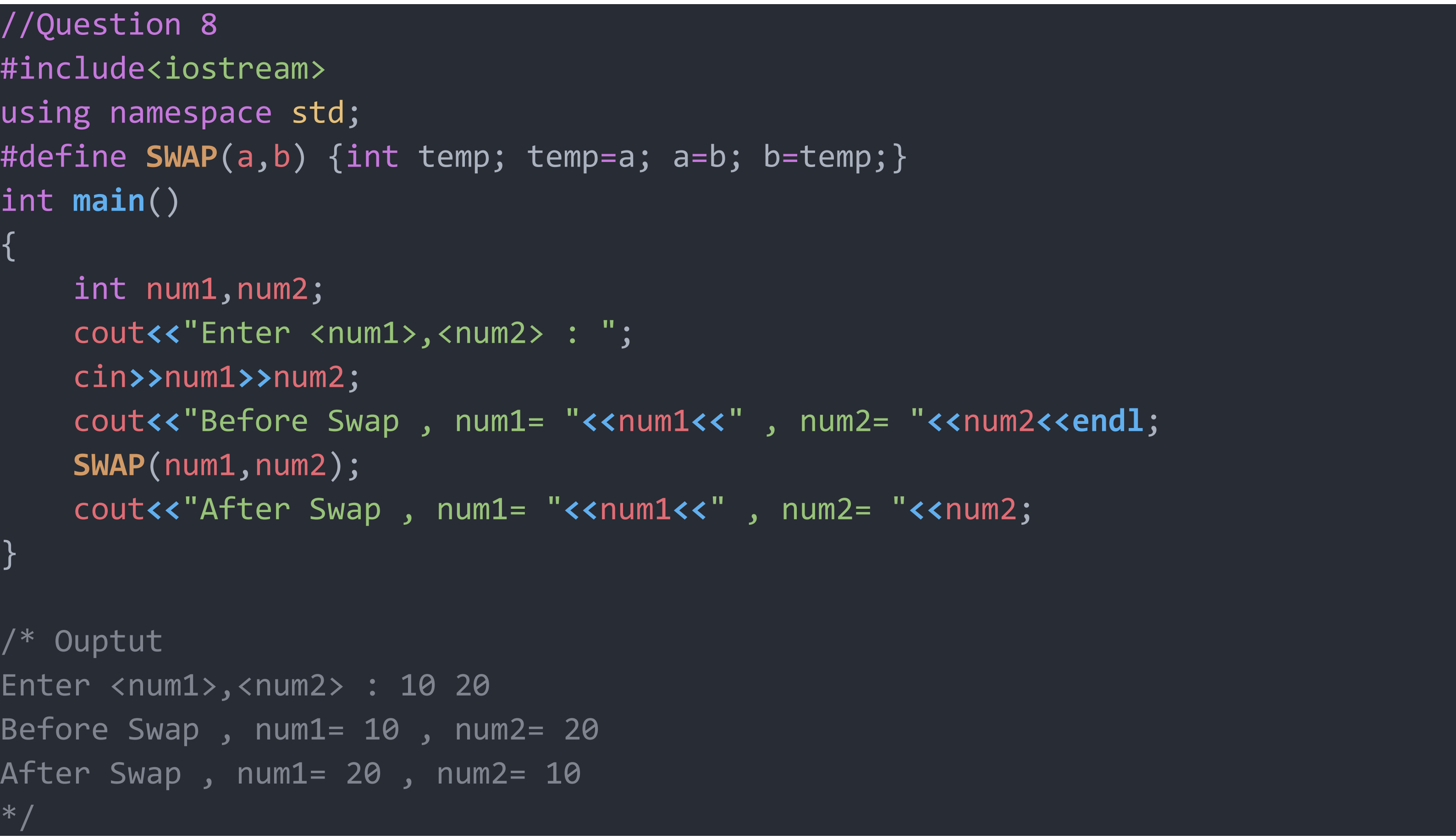
```
//Question 6
#include <iostream>
using namespace std;

//TO check whether prime or not
bool check_prime(int num)
{
    for(int n=2; n<num/2;n++)
    {
        if (num%n==0)
        {
            return false;
```



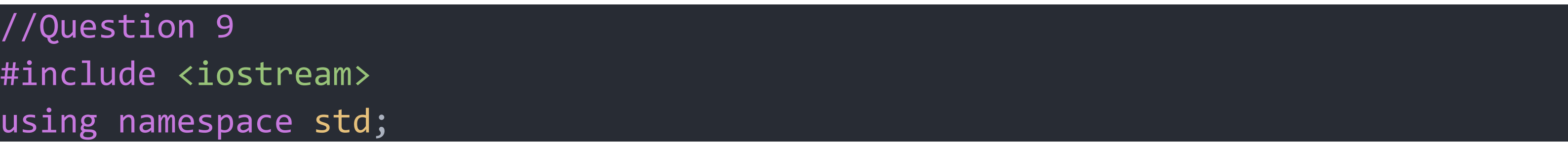
```
        }
    }
    return true;
}
int main()
{
int num=2;

for(num;num<100;num++)
    {
        bool val=check_prime(num);
        if (val==true)
        {
            cout<< num << '\n';
        }
    }
}
/* Output
2
3
4
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
*/
```



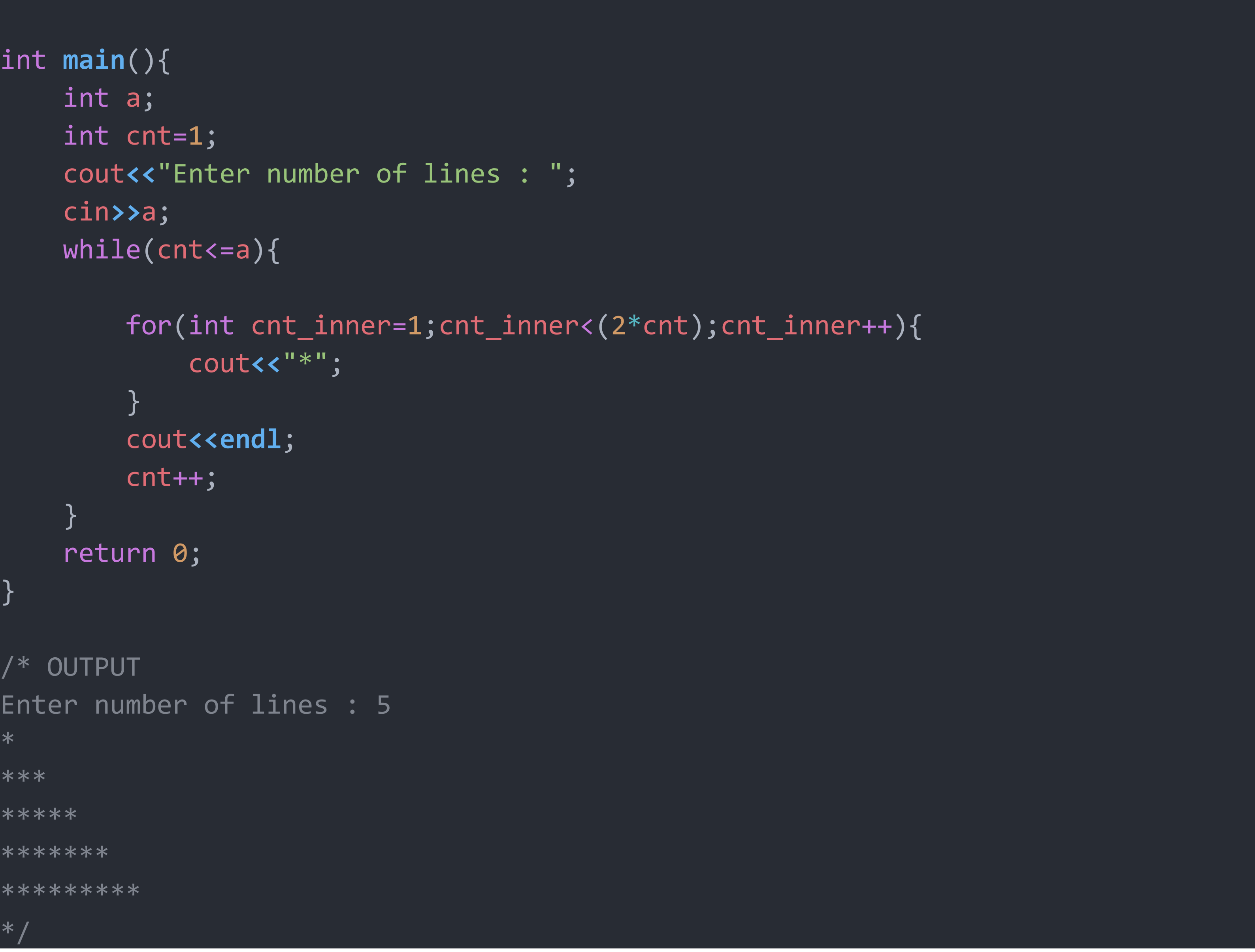
```
//Question 7
#include<iostream>
using namespace std;
int main(){
    int num;
    cout<<"Enter Number ";
    cin>>num;
    int denom=1;
    cout<<"Factors of "<<num<<" are ";
    while (denom<=(num/2)){
        if (num%denom==0){
            cout<<" "<<denom;}
        denom++;

    }
    return 0;
}

/* Output
Enter Number 100
Factors of 100 are  1 2 4 5 10 20 25 50
*/
```



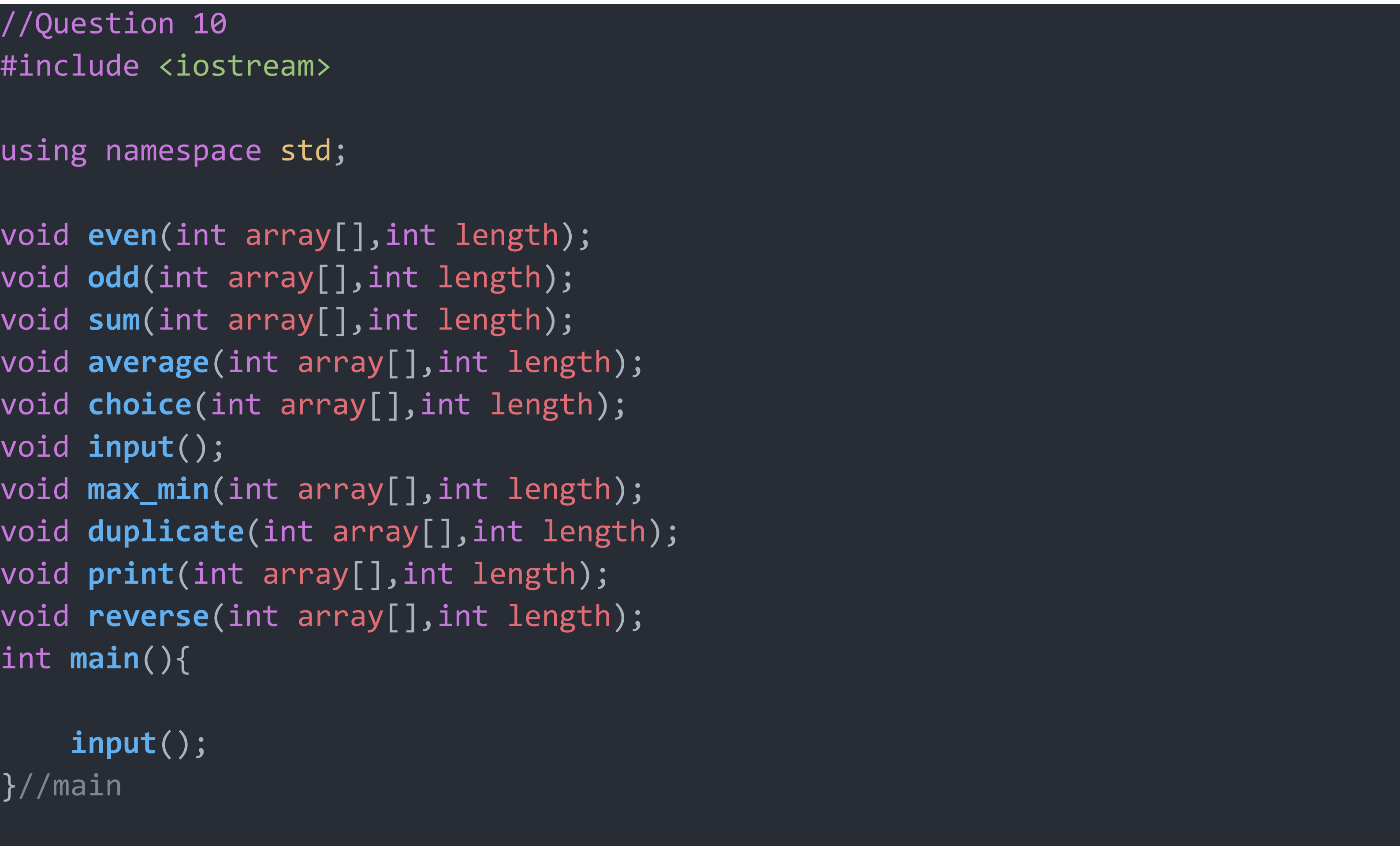
```
//Question 8
#include<iostream>
using namespace std;
#define SWAP(a,b) {int temp; temp=a; a=b; b=temp;}
int main()
{
    int num1,num2;
    cout<<"Enter <num1>,<num2> : ";
    cin>>num1>>num2;
    cout<<"Before Swap , num1= "<<num1<<" , num2= "<<num2<<endl;
    SWAP(num1,num2);
    cout<<"After Swap , num1= "<<num1<<" , num2= "<<num2;
}

/* Ouptut
Enter <num1>,<num2> : 10 20
Before Swap , num1= 10 , num2= 20
After Swap , num1= 20 , num2= 10
*/
```



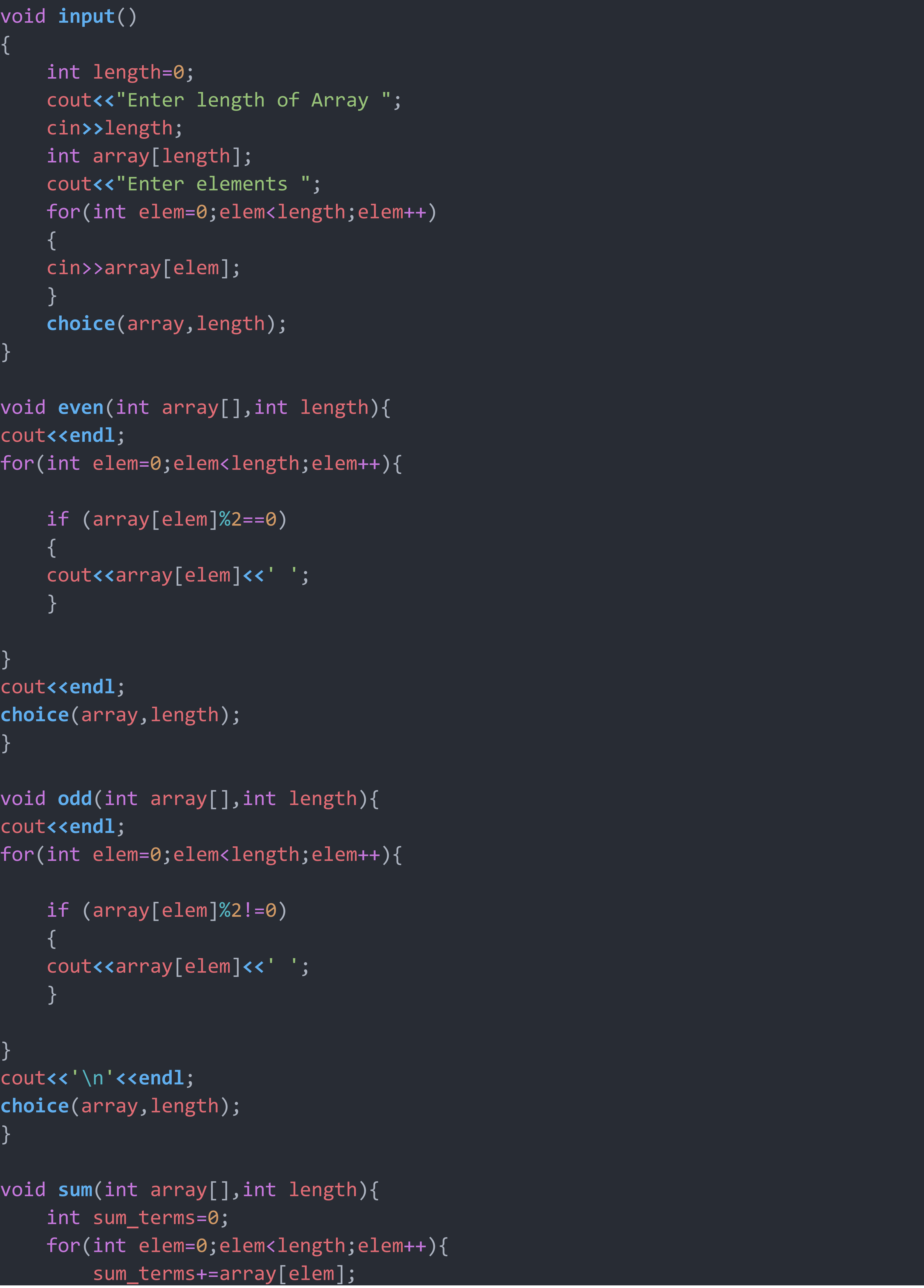
```
//Question 9
#include <iostream>
using namespace std;
```



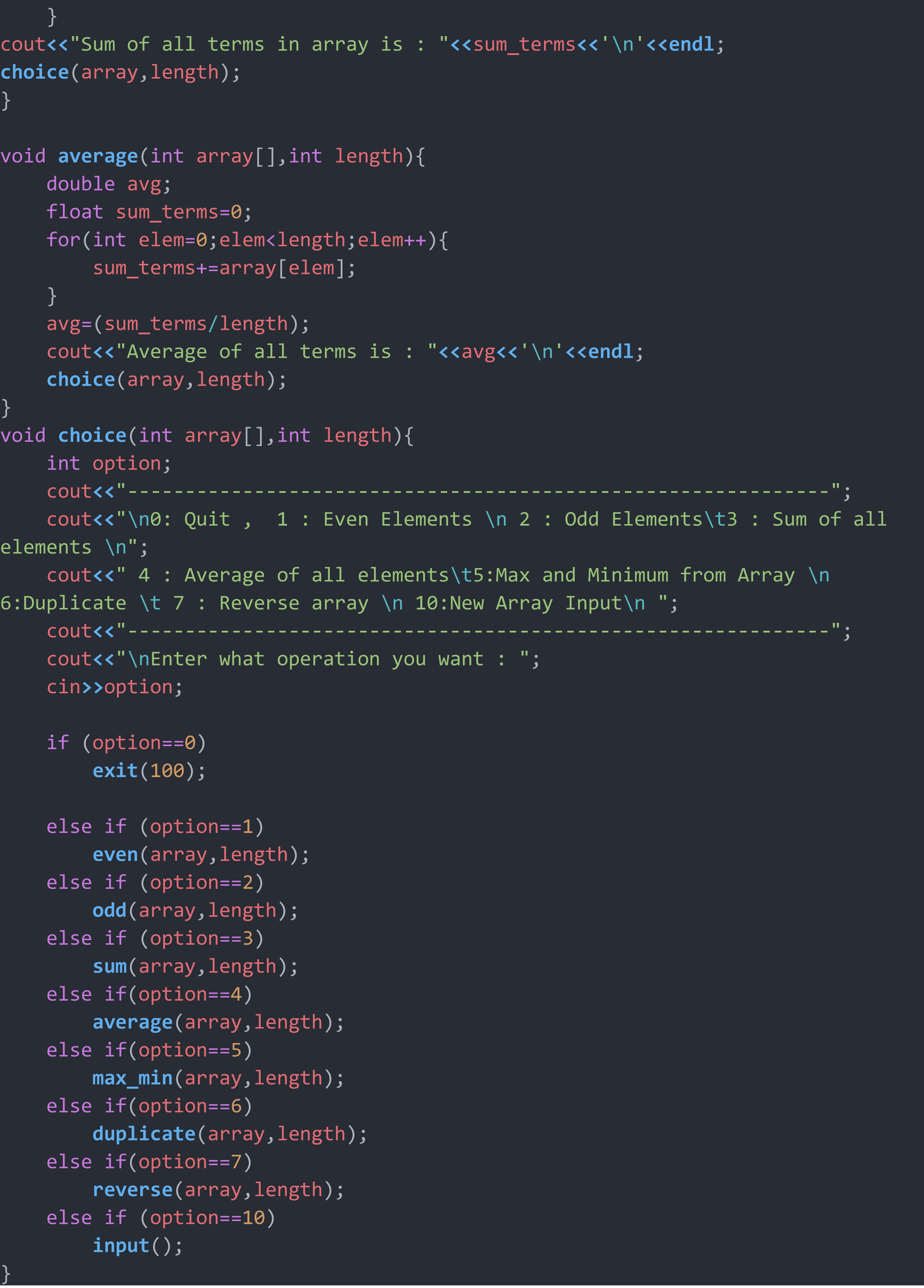
```
int main(){
    int a;
    int cnt=1;
    cout<<"Enter number of lines : ";
    cin>>a;
    while(cnt<=a){

        for(int cnt_inner=1;cnt_inner<(2*cnt);cnt_inner++){
            cout<<"*";
        }
        cout<<endl;
        cnt++;
    }
    return 0;
}

/* OUTPUT
Enter number of lines : 5
*
***
*****
*******
*********
*/
```



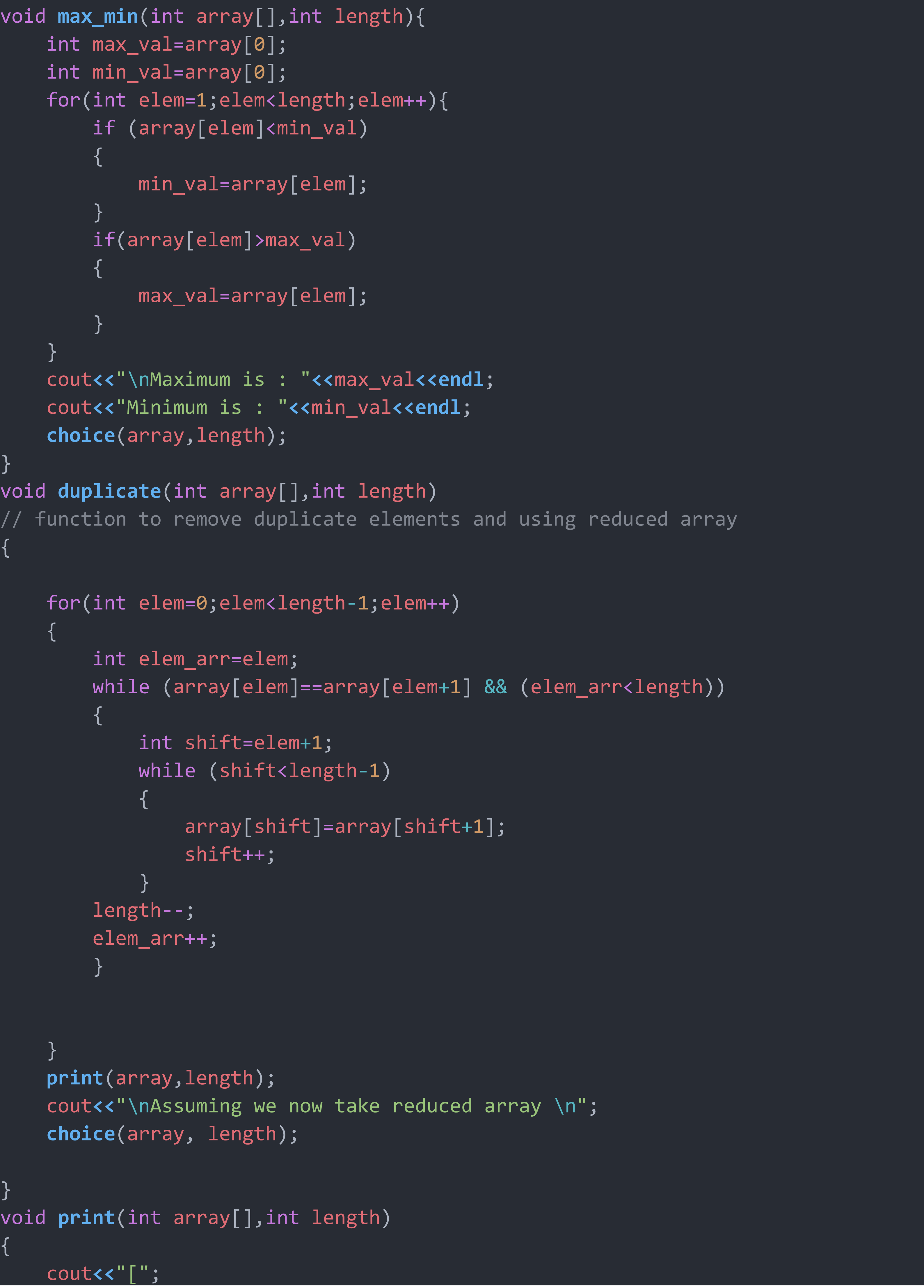
```
//Question 10
#include <iostream>

using namespace std;

void even(int array[],int length);
void odd(int array[],int length);
void sum(int array[],int length);
void average(int array[],int length);
void choice(int array[],int length);
void input();
void max_min(int array[],int length);
void duplicate(int array[],int length);
void print(int array[],int length);
void reverse(int array[],int length);
int main(){

    input();
}//main
```



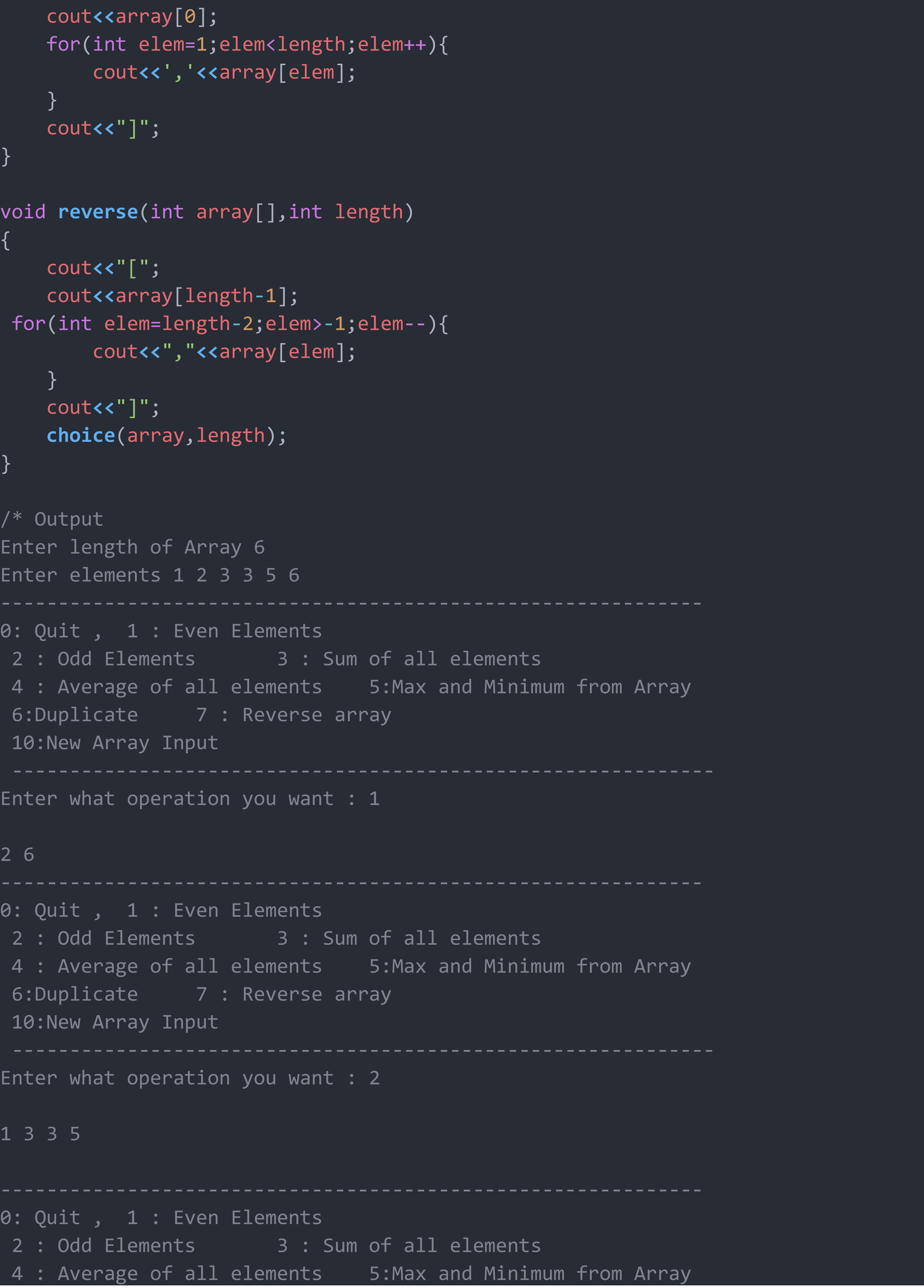
```
void input()
{
    int length=0;
    cout<<"Enter length of Array ";
    cin>>length;
    int array[length];
    cout<<"Enter elements ";
    for(int elem=0;elem<length;elem++)
    {
    cin>>array[elem];
    }
    choice(array,length);
}

void even(int array[],int length){
cout<<endl;
for(int elem=0;elem<length;elem++){

    if (array[elem]%2==0)
    {
    cout<<array[elem]<<' ';
    }

}
cout<<endl;
choice(array,length);
}

void odd(int array[],int length){
cout<<endl;
for(int elem=0;elem<length;elem++){

    if (array[elem]%2!=0)
    {
    cout<<array[elem]<<' ';
    }

}
cout<<'\n'<<endl;
choice(array,length);
}

void sum(int array[],int length){
    int sum_terms=0;
    for(int elem=0;elem<length;elem++){
        sum_terms+=array[elem];
```



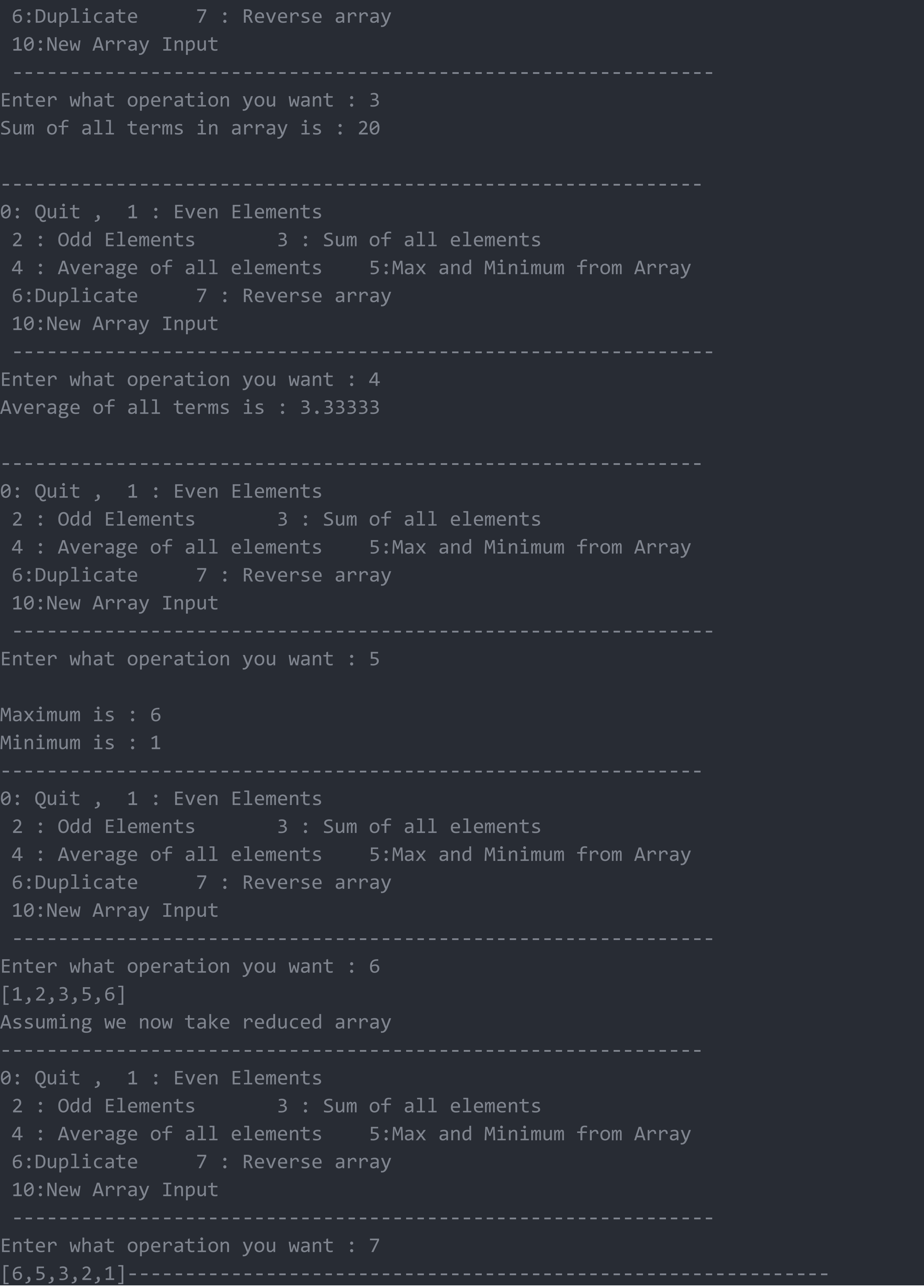
```
    }
cout<<"Sum of all terms in array is : "<<sum_terms<<'\n'<<endl;
choice(array,length);
}

void average(int array[],int length){
    double avg;
    float sum_terms=0;
    for(int elem=0;elem<length;elem++){
        sum_terms+=array[elem];
    }
    avg=(sum_terms/length);
    cout<<"Average of all terms is : "<<avg<<'\n'<<endl;
    choice(array,length);
}
void choice(int array[],int length){
    int option;
    cout<<"-------------------------------------------------------------";
    cout<<"\n0: Quit ,  1 : Even Elements \n 2 : Odd Elements\t3 : Sum of all 
elements \n";
    cout<<" 4 : Average of all elements\t5:Max and Minimum from Array \n 
6:Duplicate \t 7 : Reverse array \n 10:New Array Input\n ";
    cout<<"-------------------------------------------------------------";
    cout<<"\nEnter what operation you want : ";
    cin>>option;

    if (option==0)
        exit(100);

    else if (option==1)
        even(array,length);
    else if (option==2)
        odd(array,length);
    else if (option==3)
        sum(array,length);
    else if(option==4)
        average(array,length);
    else if(option==5)
        max_min(array,length);
    else if(option==6)
        duplicate(array,length);
    else if(option==7)
        reverse(array,length);
    else if (option==10)
        input();
}
```



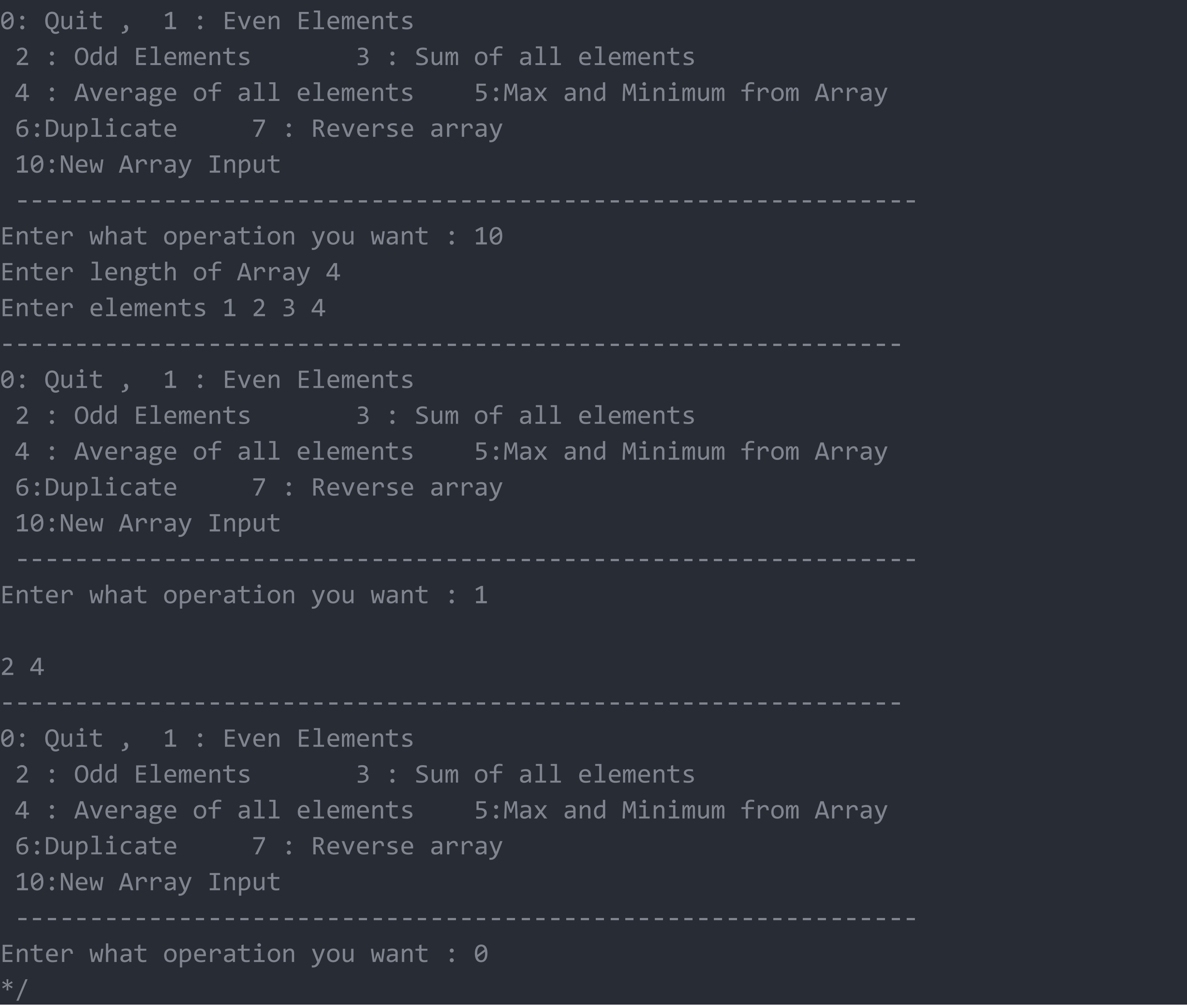
```cpp
void max_min(int array[],int length){
    int max_val=array[0];
    int min_val=array[0];
    for(int elem=1;elem<length;elem++){
        if (array[elem]<min_val)
        {
            min_val=array[elem];
        }
        if(array[elem]>max_val)
        {
            max_val=array[elem];
        }
    }
    cout<<"\nMaximum is : "<<max_val<<endl;
    cout<<"Minimum is : "<<min_val<<endl;
    choice(array,length);
}
void duplicate(int array[],int length)
// function to remove duplicate elements and using reduced array
{

    for(int elem=0;elem<length-1;elem++)
    {
        int elem_arr=elem;
        while (array[elem]==array[elem+1] && (elem_arr<length))
        {
            int shift=elem+1;
            while (shift<length-1)
            {
                array[shift]=array[shift+1];
                shift++;
            }
        length--;
        elem_arr++;
        }


    }
    print(array,length);
    cout<<"\nAssuming we now take reduced array \n";
    choice(array, length);

}
void print(int array[],int length)
{
    cout<<"[";
```



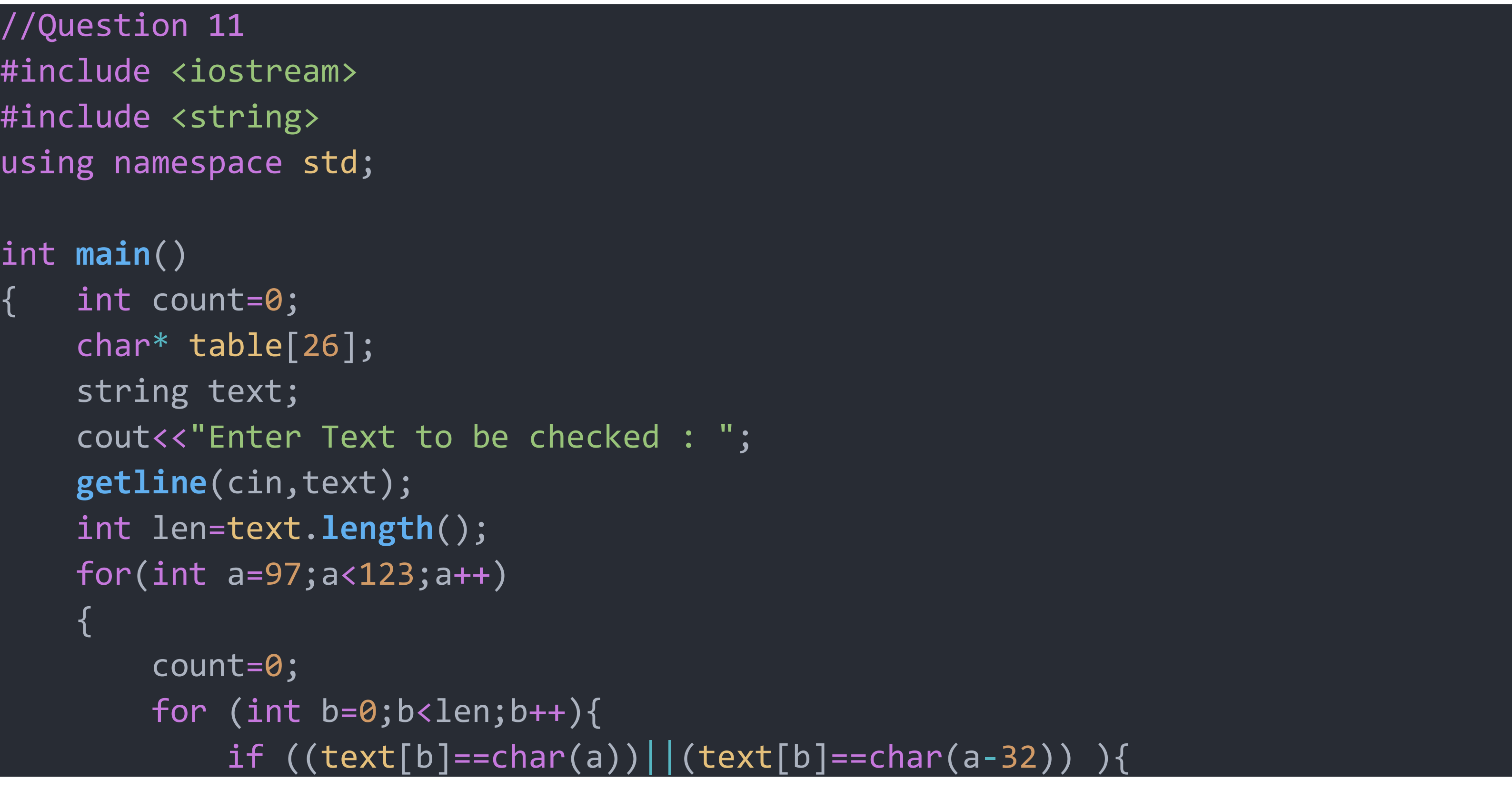
```
    cout<<array[0];
    for(int elem=1;elem<length;elem++){
        cout<<','<<array[elem];
    }
    cout<<"]";
}

void reverse(int array[],int length)
{
    cout<<"[";
    cout<<array[length-1];
 for(int elem=length-2;elem>-1;elem--){
        cout<<","<<array[elem];
    }
    cout<<"]";
    choice(array,length);
}

/* Output
Enter length of Array 6
Enter elements 1 2 3 3 5 6
-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 1

2 6
-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 2

1 3 3 5

-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
```



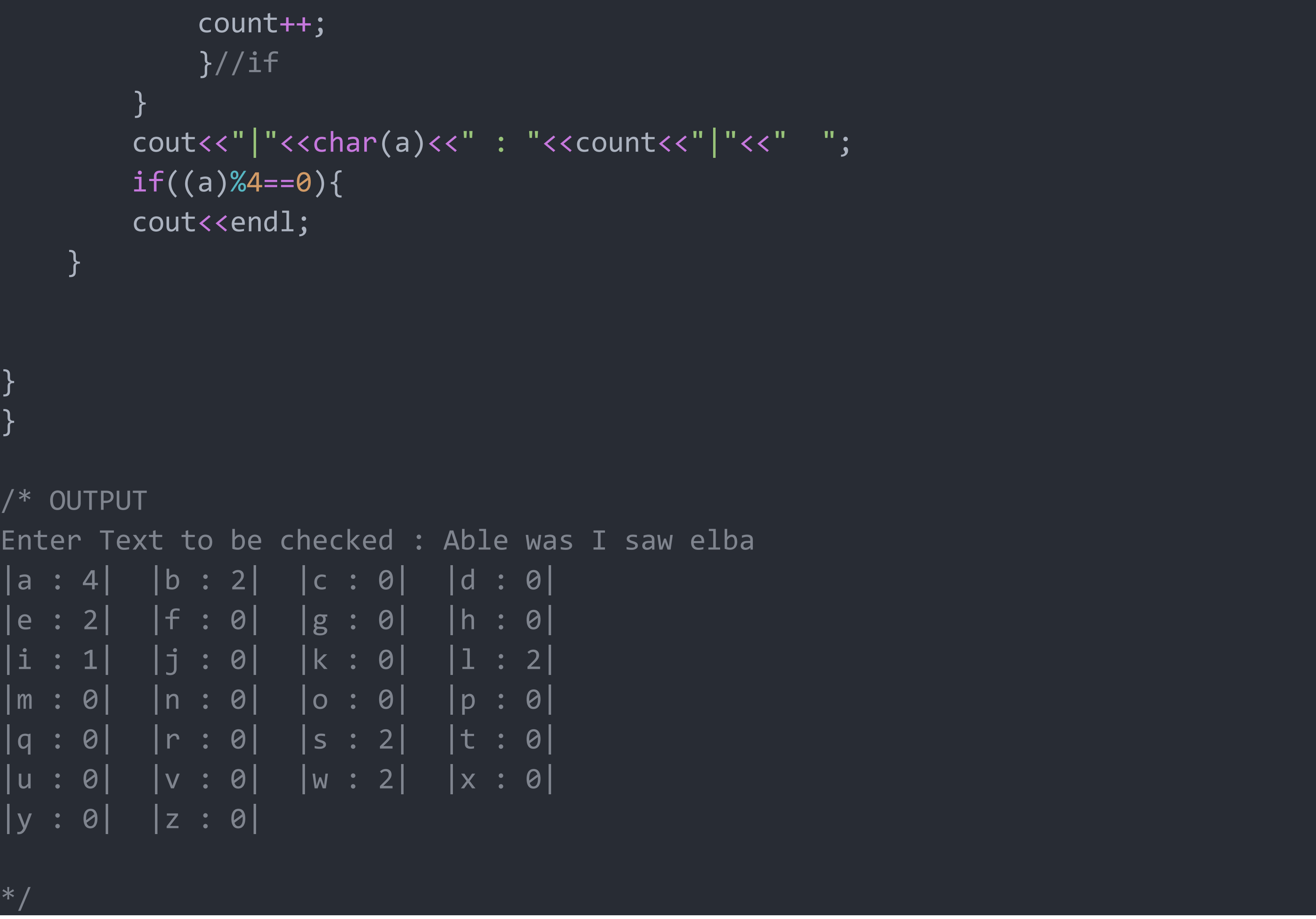
```
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 3
Sum of all terms in array is : 20


-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 4
Average of all terms is : 3.33333


-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 5

Maximum is : 6
Minimum is : 1
-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 6
[1,2,3,5,6]
Assuming we now take reduced array
-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 7
[6,5,3,2,1]-------------------------------------------------------------
```



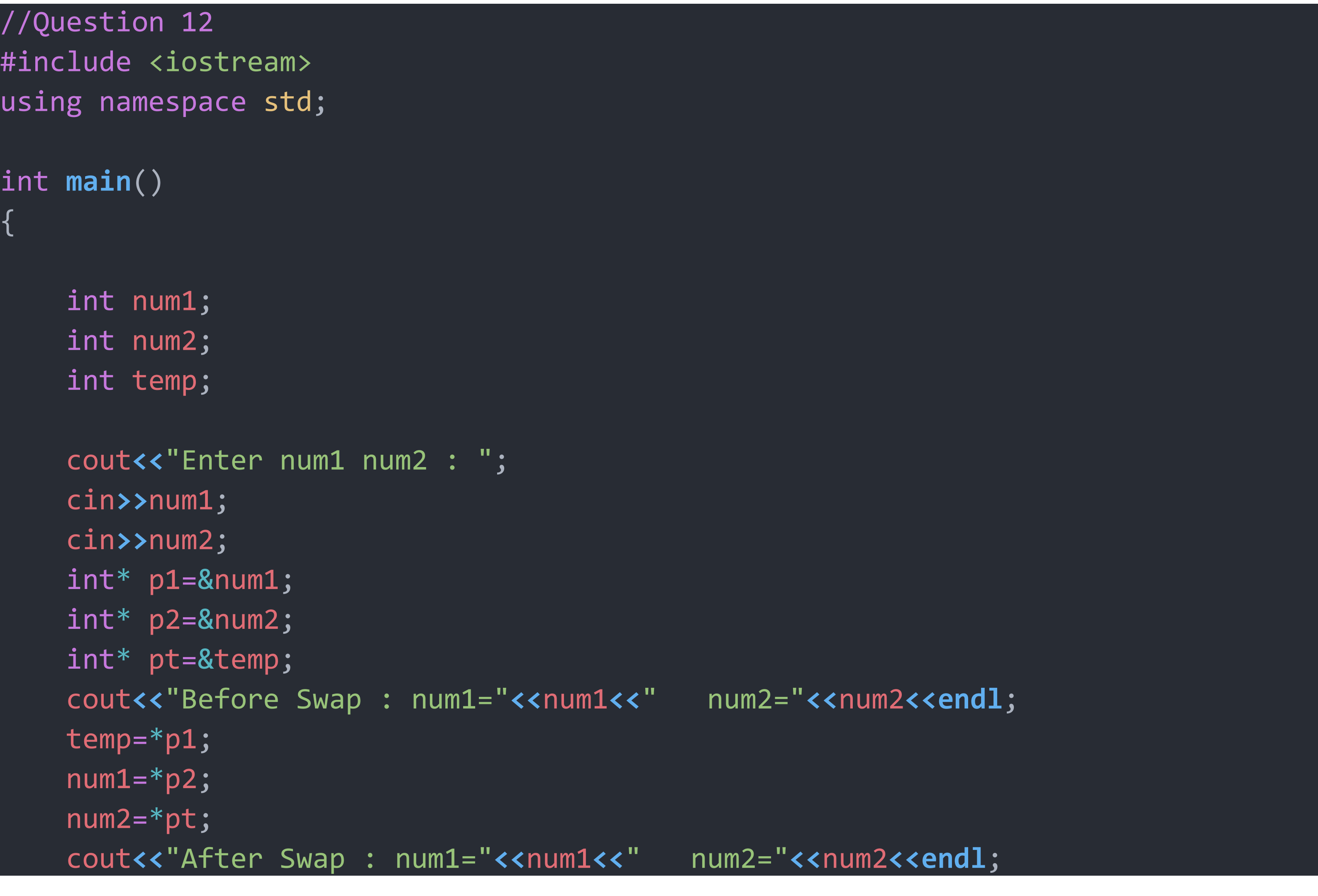
```
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 10
Enter length of Array 4
Enter elements 1 2 3 4
-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 1

2 4
-------------------------------------------------------------
0: Quit ,  1 : Even Elements
 2 : Odd Elements       3 : Sum of all elements
 4 : Average of all elements    5:Max and Minimum from Array
 6:Duplicate     7 : Reverse array
 10:New Array Input
 -------------------------------------------------------------
Enter what operation you want : 0
*/
```



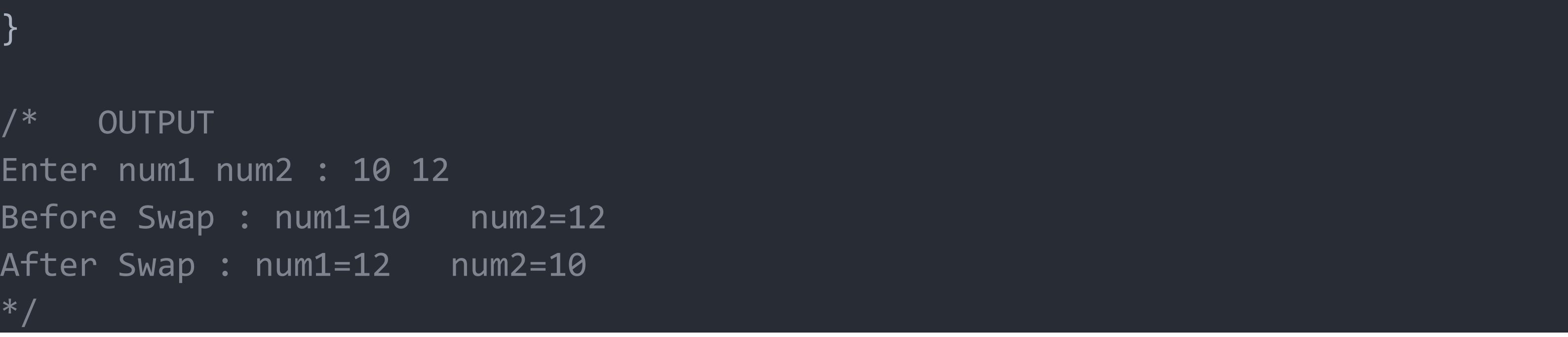
```cpp
//Question 11
#include <iostream>
#include <string>
using namespace std;

int main()
{   int count=0;
    char* table[26];
    string text;
    cout<<"Enter Text to be checked : ";
    getline(cin,text);
    int len=text.length();
    for(int a=97;a<123;a++)
    {
        count=0;
        for (int b=0;b<len;b++){
            if ((text[b]==char(a))||(text[b]==char(a-32)) ){
```



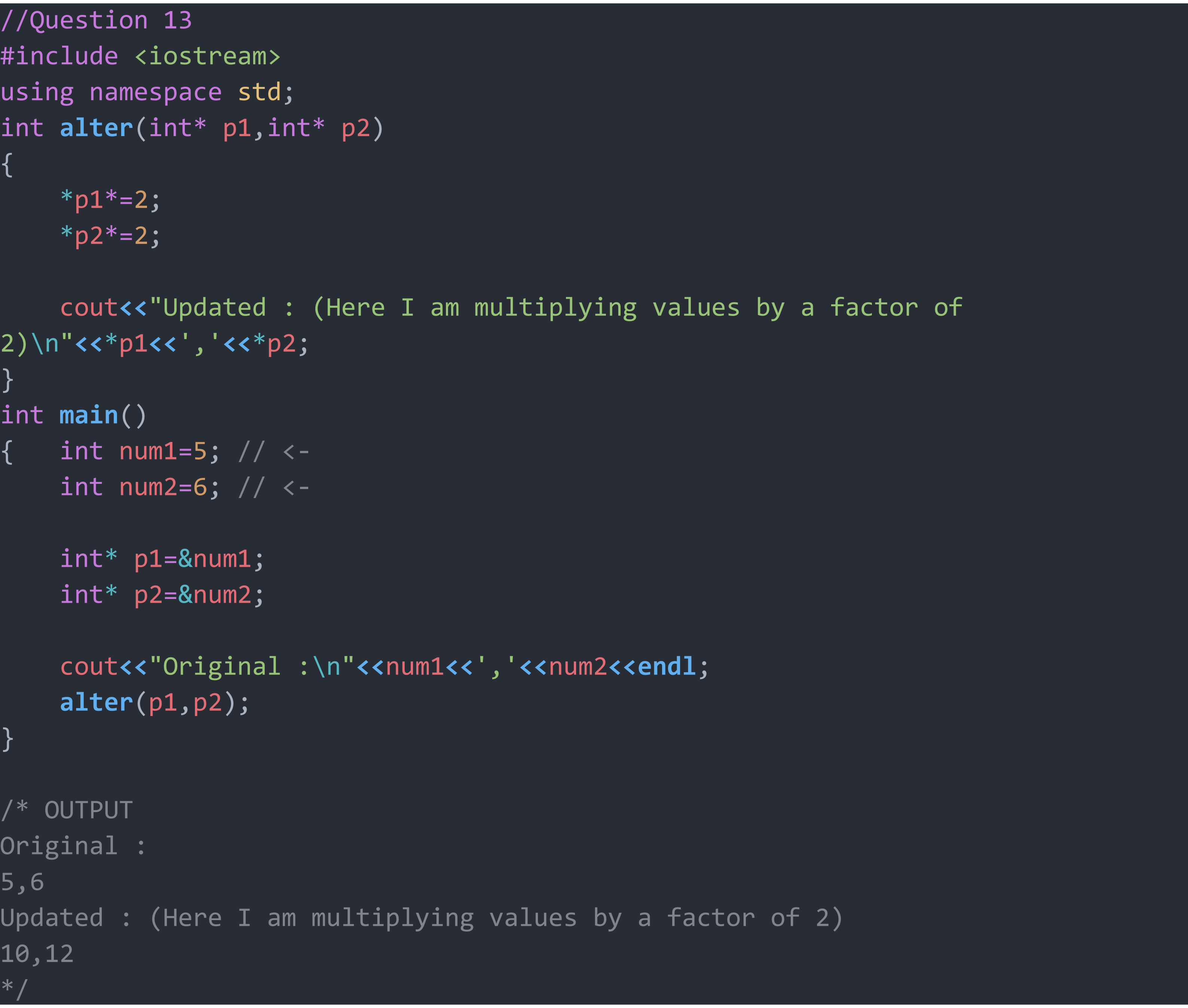
```
            count++;
            }//if
        }
        cout<<"|"<<char(a)<<" : "<<count<<"|"<<"  ";
        if((a)%4==0){
        cout<<endl;
    }


}
}

/* OUTPUT
Enter Text to be checked : Able was I saw elba
|a : 4|  |b : 2|  |c : 0|  |d : 0|
|e : 2|  |f : 0|  |g : 0|  |h : 0|
|i : 1|  |j : 0|  |k : 0|  |l : 2|
|m : 0|  |n : 0|  |o : 0|  |p : 0|
|q : 0|  |r : 0|  |s : 2|  |t : 0|
|u : 0|  |v : 0|  |w : 2|  |x : 0|
|y : 0|  |z : 0|

*/
```



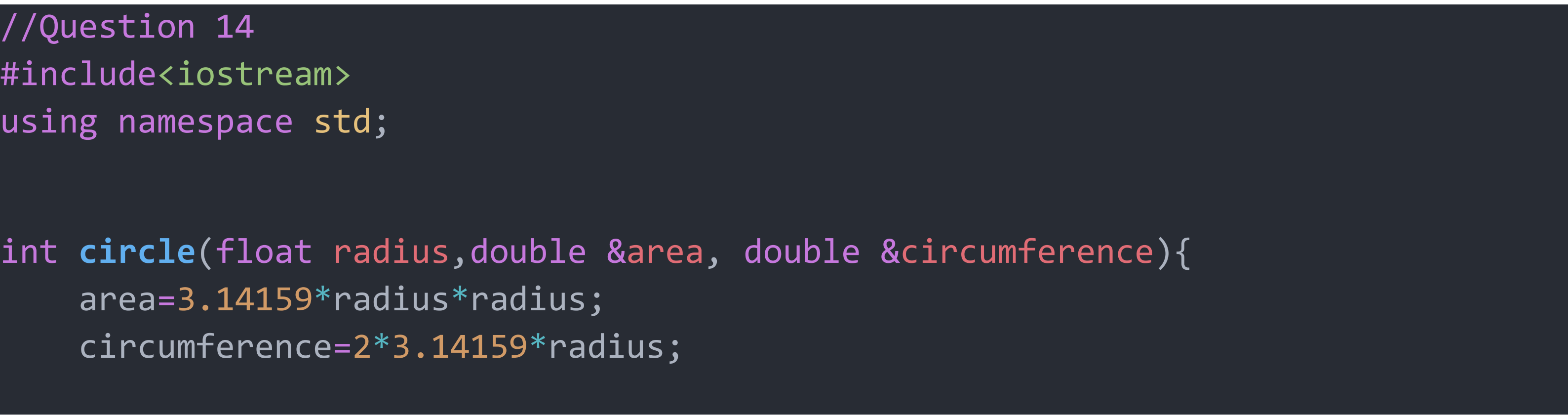
```
//Question 12
#include <iostream>
using namespace std;

int main()
{

    int num1;
    int num2;
    int temp;

    cout<<"Enter num1 num2 : ";
    cin>>num1;
    cin>>num2;
    int* p1=&num1;
    int* p2=&num2;
    int* pt=&temp;
    cout<<"Before Swap : num1="<<num1<<"   num2="<<num2<<endl;
    temp=*p1;
    num1=*p2;
    num2=*pt;
    cout<<"After Swap : num1="<<num1<<"   num2="<<num2<<endl;
```



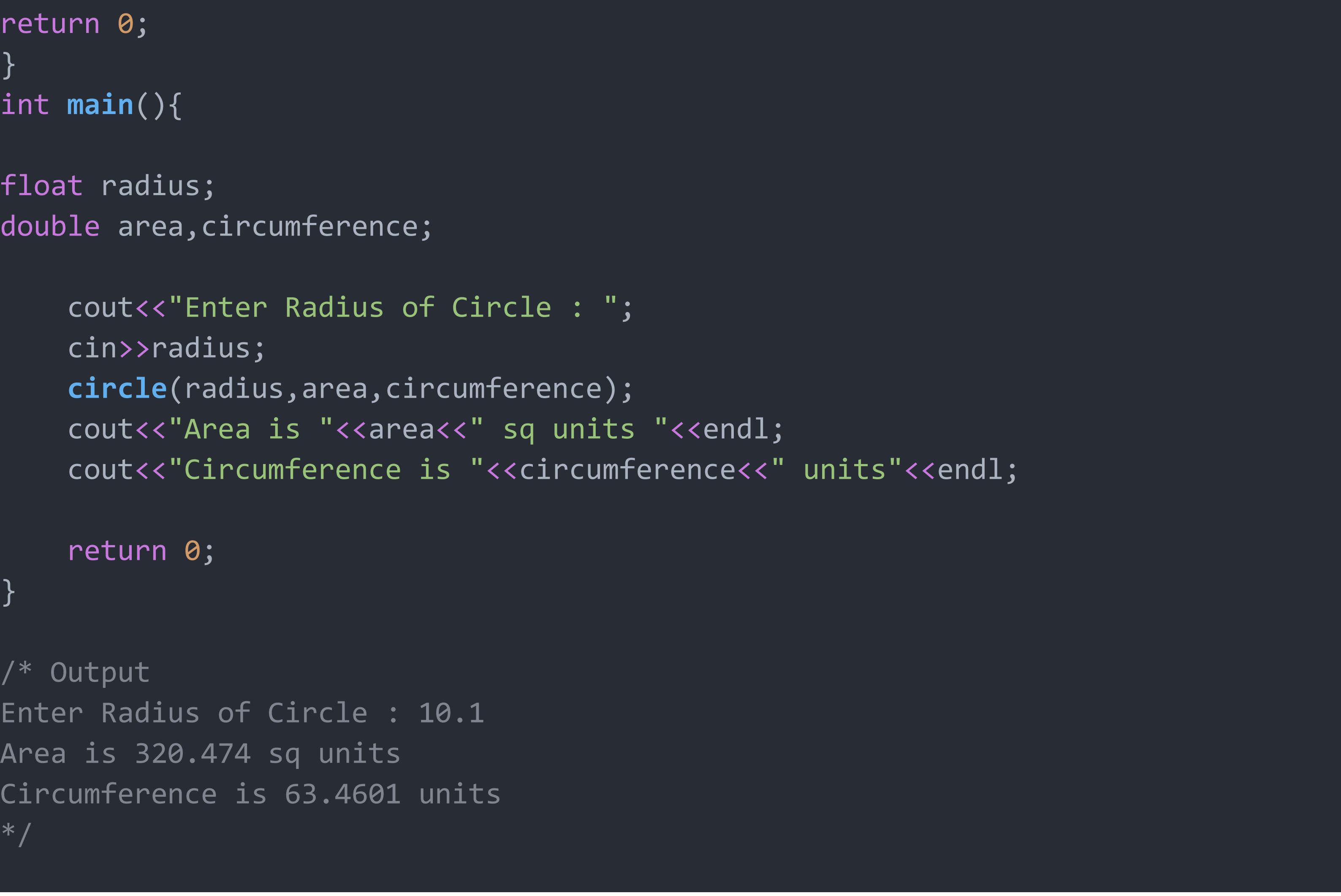
```
}

/*   OUTPUT
Enter num1 num2 : 10 12
Before Swap : num1=10   num2=12
After Swap : num1=12   num2=10
*/
```



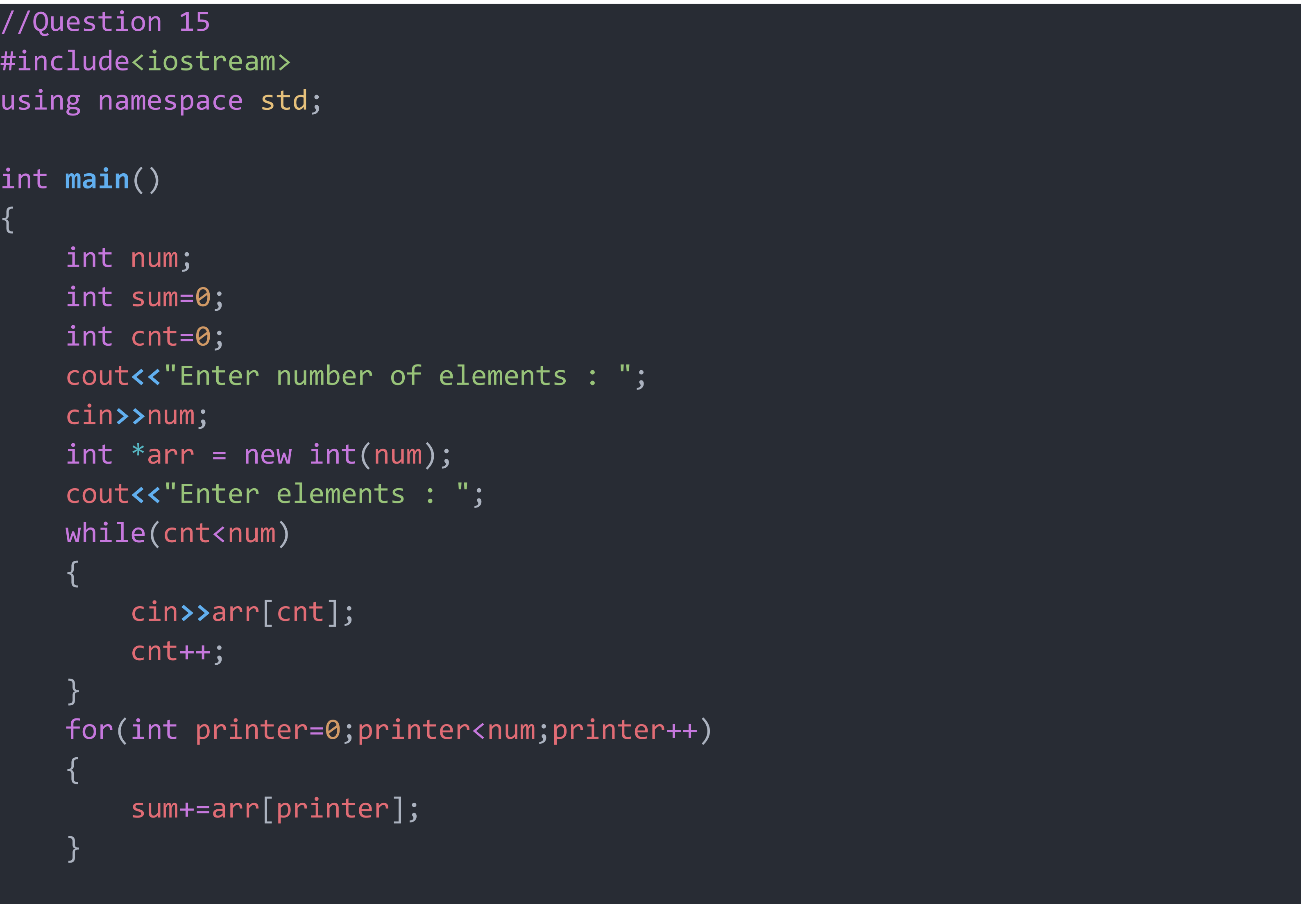
```
//Question 13
#include <iostream>
using namespace std;
int alter(int* p1,int* p2)
{
    *p1*=2;
    *p2*=2;

    cout<<"Updated : (Here I am multiplying values by a factor of
2)\n"<<*p1<<','<<*p2;
}
int main()
{   int num1=5; // <-
    int num2=6; // <-

    int* p1=&num1;
    int* p2=&num2;

    cout<<"Original :\n"<<num1<<','<<num2<<endl;
    alter(p1,p2);
}

/* OUTPUT
Original :
5,6
Updated : (Here I am multiplying values by a factor of 2)
10,12
*/
```



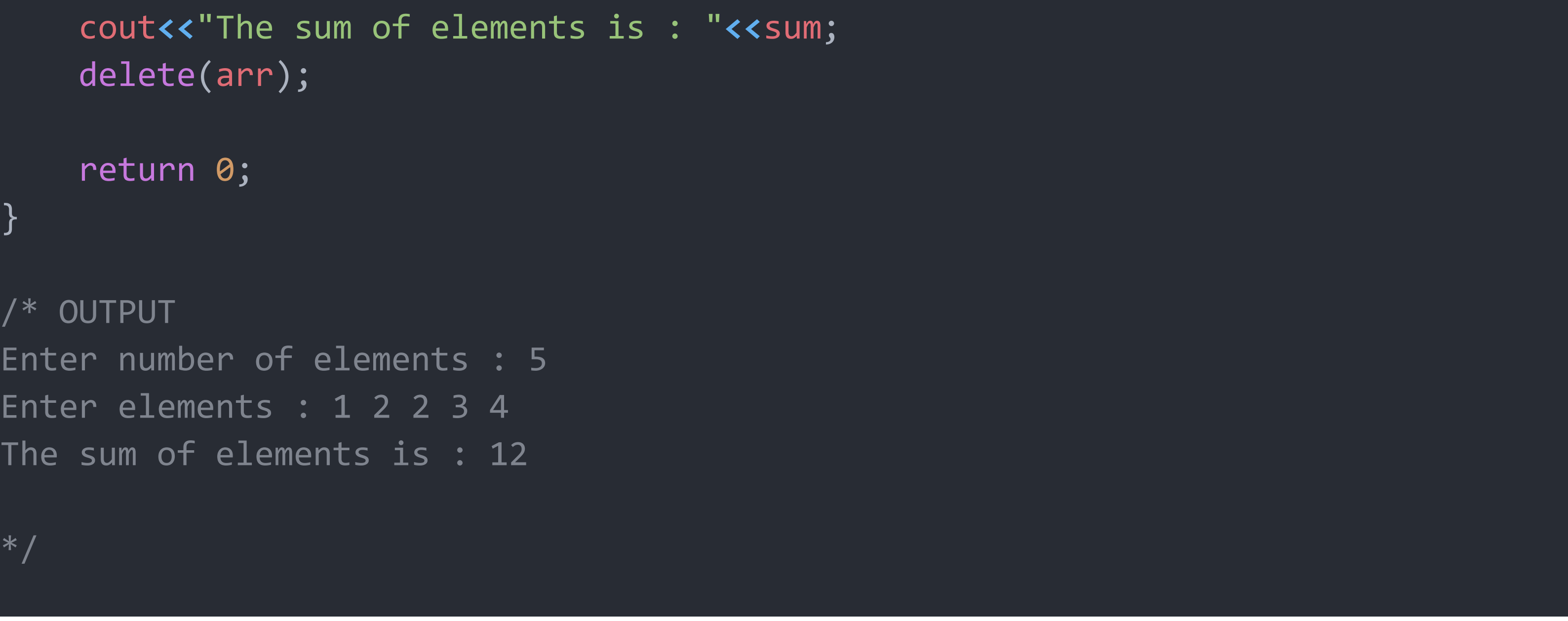
```
//Question 14
#include<iostream>
using namespace std;


int circle(float radius,double &area, double &circumference){
    area=3.14159*radius*radius;
    circumference=2*3.14159*radius;
```



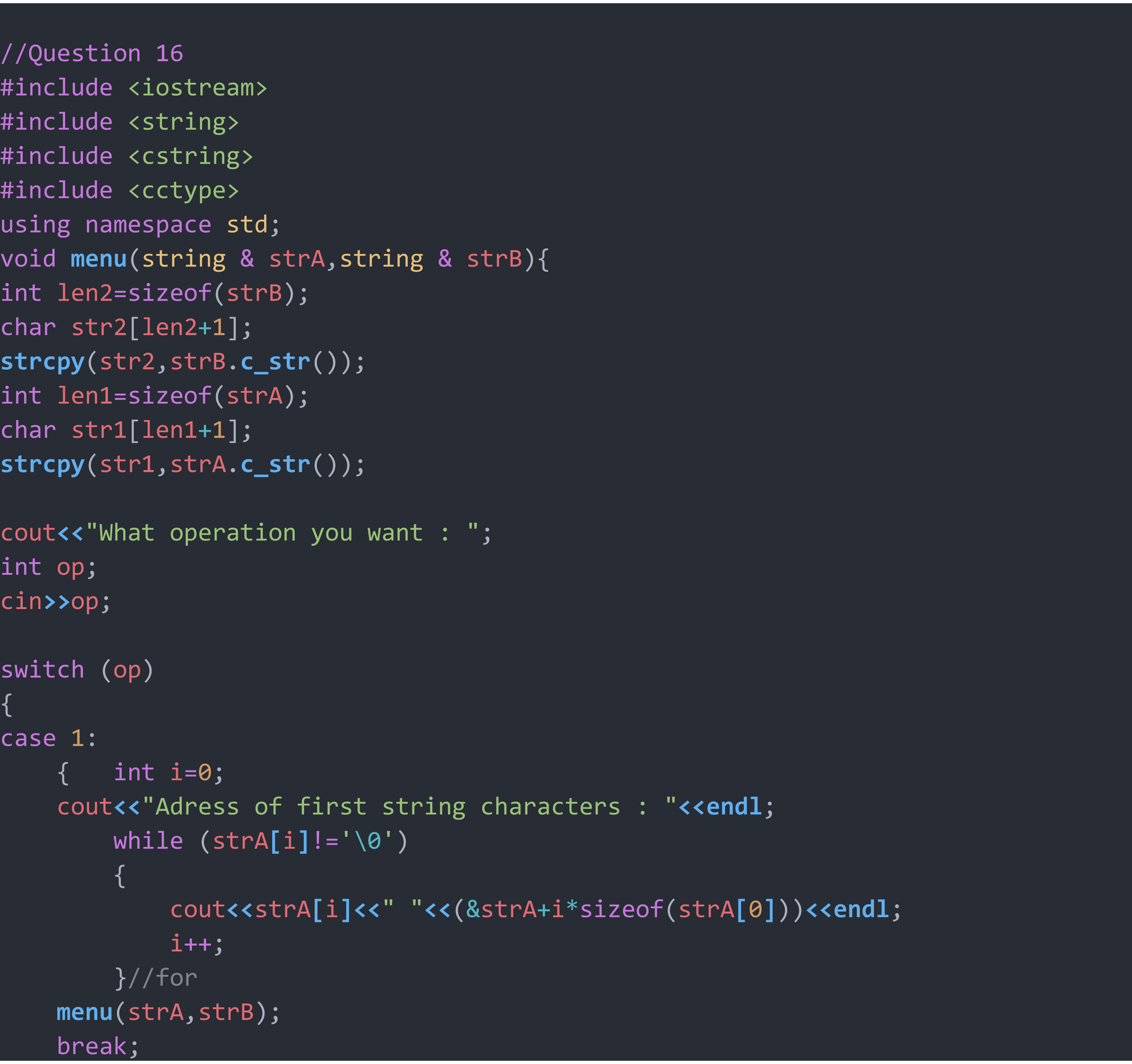
```
return 0;
}
int main(){

float radius;
double area,circumference;

    cout<<"Enter Radius of Circle : ";
    cin>>radius;
    circle(radius,area,circumference);
    cout<<"Area is "<<area<<" sq units "<<endl;
    cout<<"Circumference is "<<circumference<<" units"<<endl;

    return 0;
}

/* Output
Enter Radius of Circle : 10.1
Area is 320.474 sq units
Circumference is 63.4601 units
*/
```



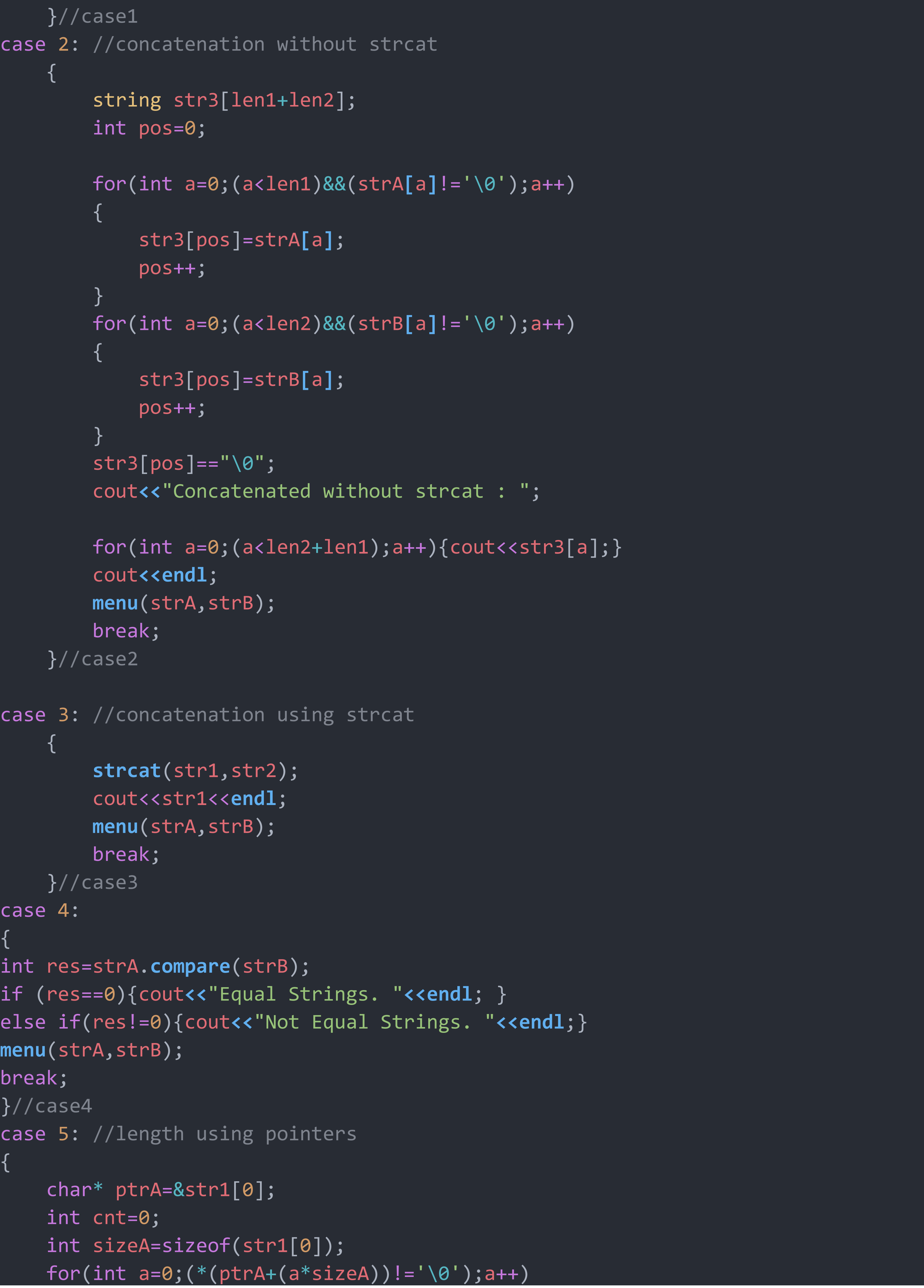
```
//Question 15
#include<iostream>
using namespace std;

int main()
{
    int num;
    int sum=0;
    int cnt=0;
    cout<<"Enter number of elements : ";
    cin>>num;
    int *arr = new int(num);
    cout<<"Enter elements : ";
    while(cnt<num)
    {
        cin>>arr[cnt];
        cnt++;
    }
    for(int printer=0;printer<num;printer++)
    {
        sum+=arr[printer];
    }
```



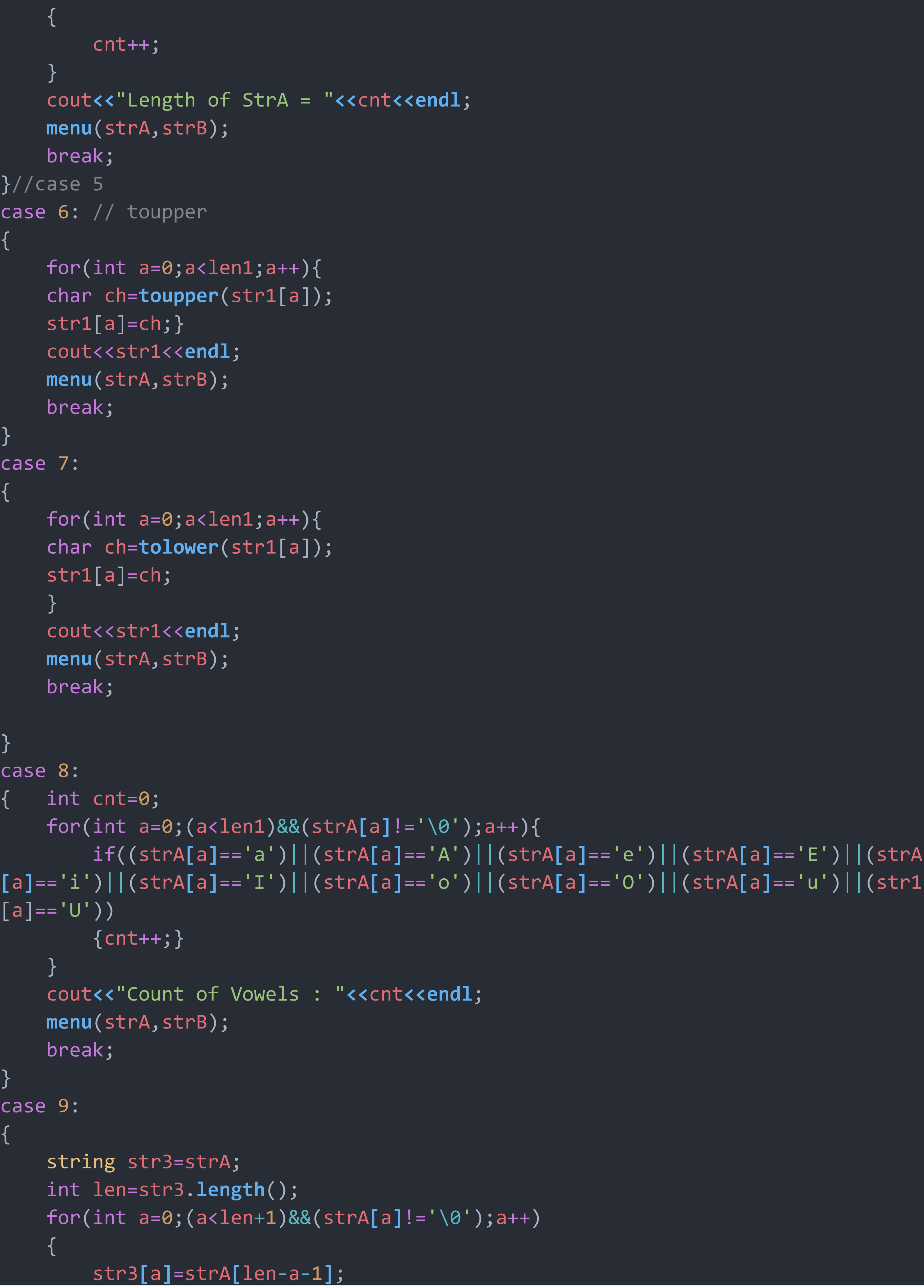
```
    cout<<"The sum of elements is : "<<sum;
    delete(arr);

    return 0;
}

/* OUTPUT
Enter number of elements : 5
Enter elements : 1 2 2 3 4
The sum of elements is : 12

*/
```



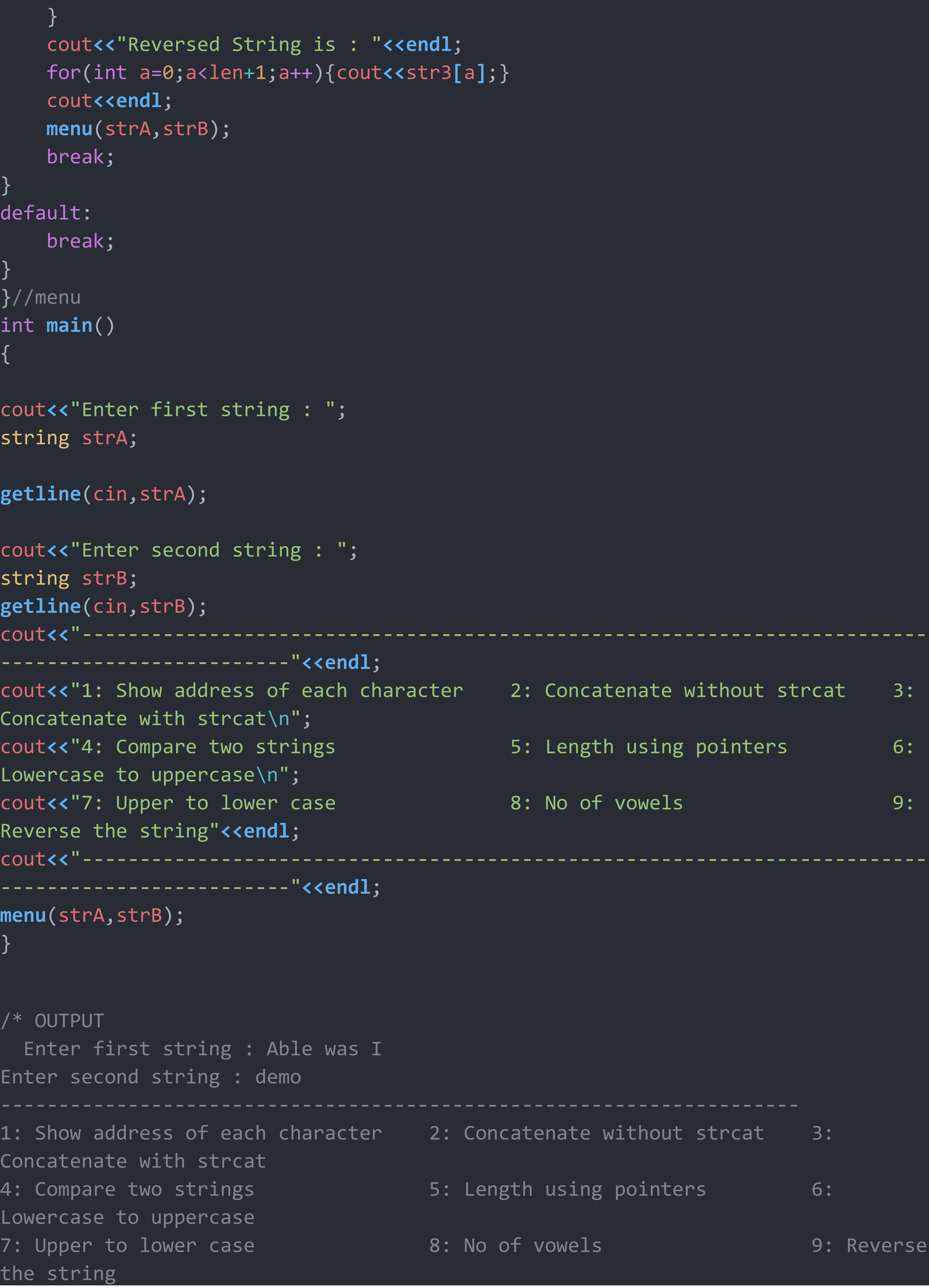
```
//Question 16
#include <iostream>
#include <string>
#include <cstring>
#include <cctype>
using namespace std;
void menu(string & strA,string & strB){
int len2=sizeof(strB);
char str2[len2+1];
strcpy(str2,strB.c_str());
int len1=sizeof(strA);
char str1[len1+1];
strcpy(str1,strA.c_str());

cout<<"What operation you want : ";
int op;
cin>>op;

switch (op)
{
case 1:
    {   int i=0;
    cout<<"Adress of first string characters : "<<endl;
        while (strA[i]!='\0')
        {
            cout<<strA[i]<<" "<<(&strA+i*sizeof(strA[0]))<<endl;
            i++;
        }//for
    menu(strA,strB);
    break;
```



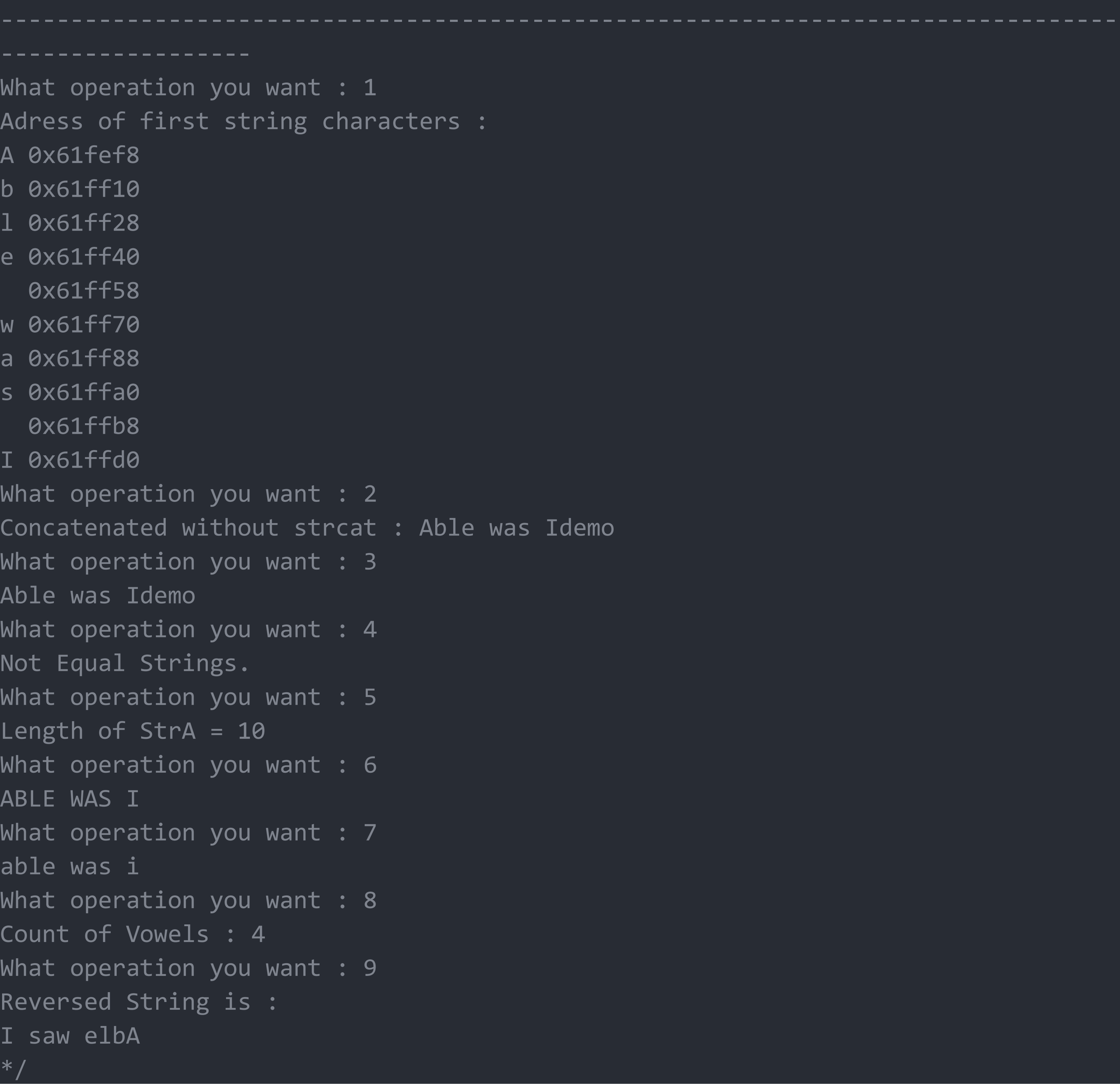
```
    }//case1
case 2: //concatenation without strcat
    {
        string str3[len1+len2];
        int pos=0;

        for(int a=0;(a<len1)&&(strA[a]!='\0');a++)
        {
            str3[pos]=strA[a];
            pos++;
        }
        for(int a=0;(a<len2)&&(strB[a]!='\0');a++)
        {
            str3[pos]=strB[a];
            pos++;
        }
        str3[pos]=="\0";
        cout<<"Concatenated without strcat : ";

        for(int a=0;(a<len2+len1);a++){cout<<str3[a];}
        cout<<endl;
        menu(strA,strB);
        break;
    }//case2

case 3: //concatenation using strcat
    {
        strcat(str1,str2);
        cout<<str1<<endl;
        menu(strA,strB);
        break;
    }//case3
case 4:
{
int res=strA.compare(strB);
if (res==0){cout<<"Equal Strings. "<<endl; }
else if(res!=0){cout<<"Not Equal Strings. "<<endl;}
menu(strA,strB);
break;
}//case4
case 5: //length using pointers
{
    char* ptrA=&str1[0];
    int cnt=0;
    int sizeA=sizeof(str1[0]);
    for(int a=0;(*(ptrA+(a*sizeA))!='\0');a++)
```



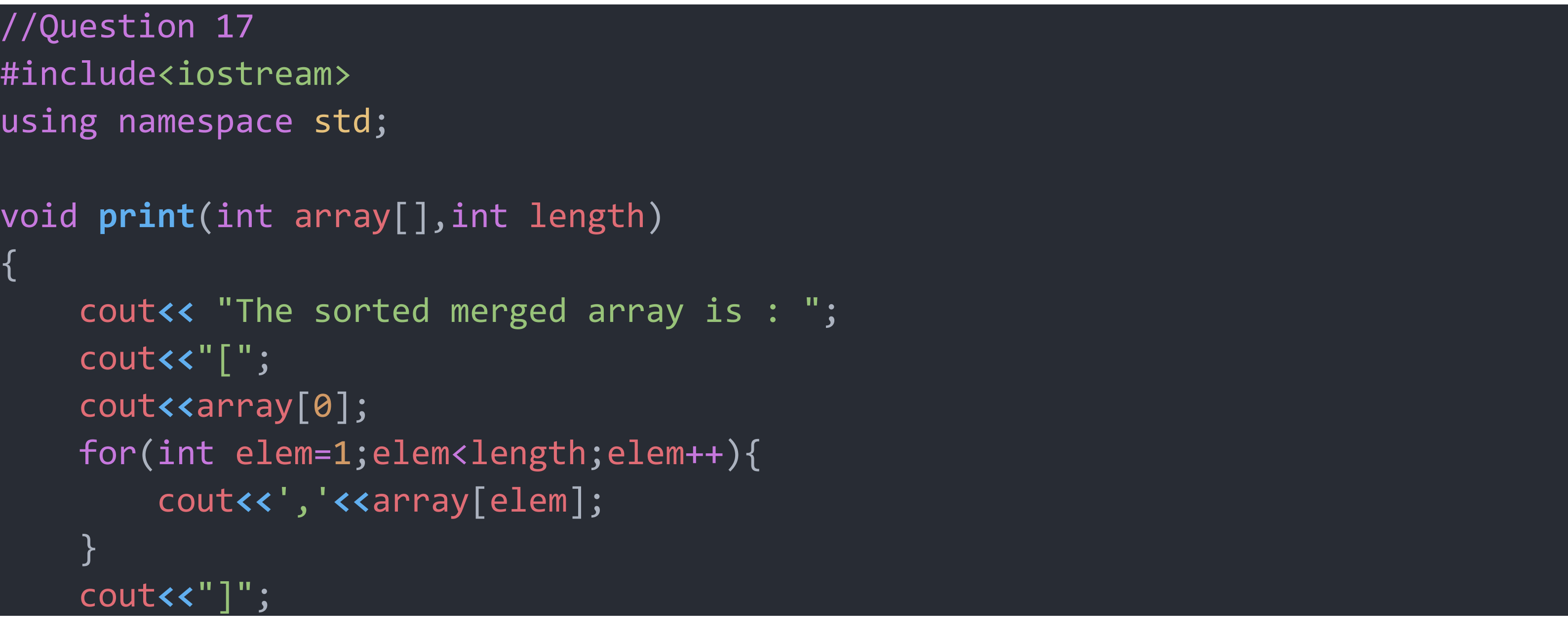
```
    {
        cnt++;
    }
    cout<<"Length of StrA = "<<cnt<<endl;
    menu(strA,strB);
    break;
}//case 5
case 6: // toupper
{
    for(int a=0;a<len1;a++){
    char ch=toupper(str1[a]);
    str1[a]=ch;}
    cout<<str1<<endl;
    menu(strA,strB);
    break;
}
case 7:
{
    for(int a=0;a<len1;a++){
    char ch=tolower(str1[a]);
    str1[a]=ch;
    }
    cout<<str1<<endl;
    menu(strA,strB);
    break;

}
case 8:
{   int cnt=0;
    for(int a=0;(a<len1)&&(strA[a]!='\0');a++){
        if((strA[a]=='a')||(strA[a]=='A')||(strA[a]=='e')||(strA[a]=='E')||(strA
[a]=='i')||(strA[a]=='I')||(strA[a]=='o')||(strA[a]=='O')||(strA[a]=='u')||(str1
[a]=='U'))
        {cnt++;}
    }
    cout<<"Count of Vowels : "<<cnt<<endl;
    menu(strA,strB);
    break;
}
case 9:
{
    string str3=strA;
    int len=str3.length();
    for(int a=0;(a<len+1)&&(strA[a]!='\0');a++)
    {
        str3[a]=strA[len-a-1];
```



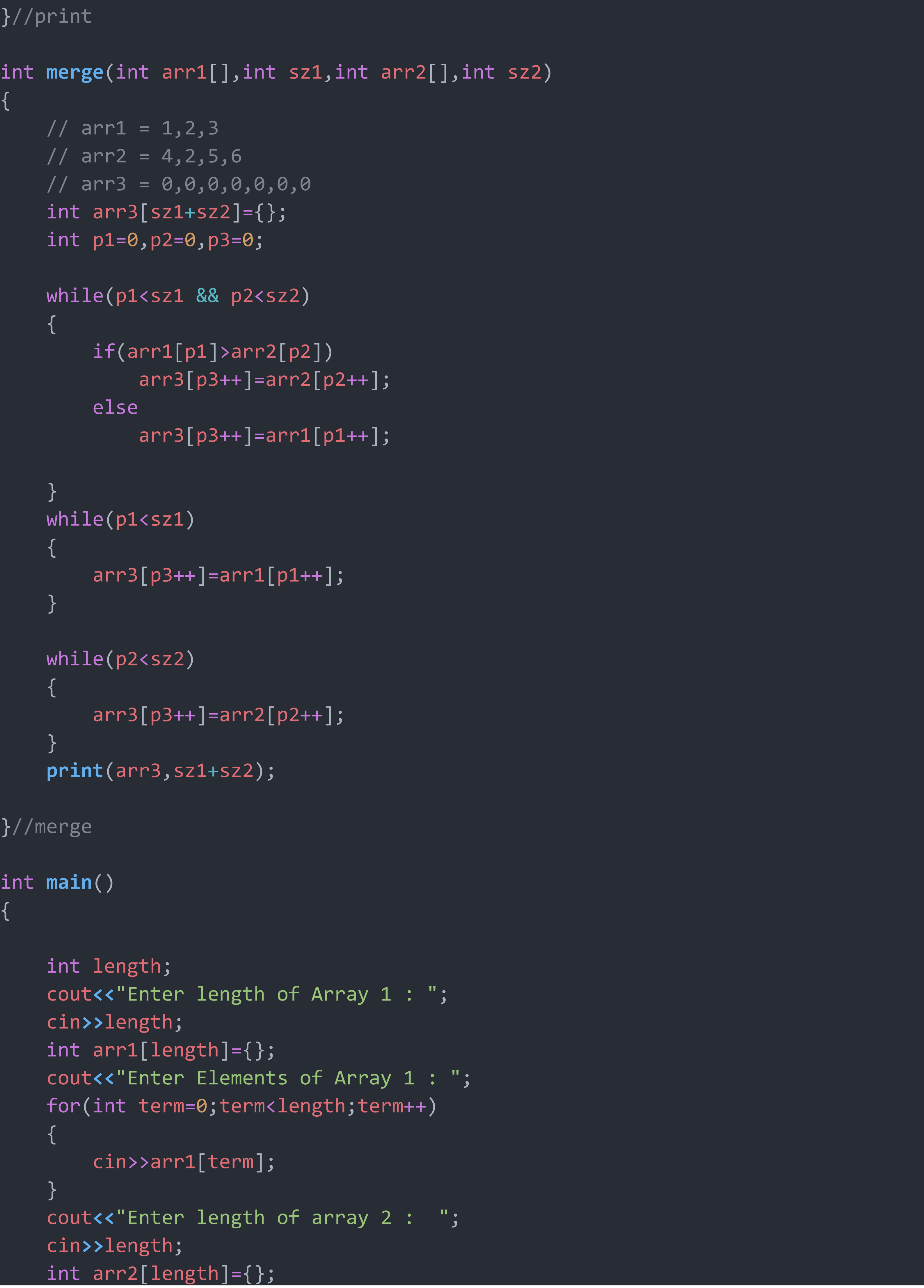
```
    }
    cout<<"Reversed String is : "<<endl;
    for(int a=0;a<len+1;a++){cout<<str3[a];}
    cout<<endl;
    menu(strA,strB);
    break;
}
default:
    break;
}
}//menu
int main()
{

cout<<"Enter first string : ";
string strA;

getline(cin,strA);

cout<<"Enter second string : ";
string strB;
getline(cin,strB);
cout<<"--------------------------------------------------------------------------
-------------------------"<<endl;
cout<<"1: Show address of each character    2: Concatenate without strcat    3:
Concatenate with strcat\n";
cout<<"4: Compare two strings               5: Length using pointers         6:
Lowercase to uppercase\n";
cout<<"7: Upper to lower case               8: No of vowels                  9:
Reverse the string"<<endl;
cout<<"--------------------------------------------------------------------------
-------------------------"<<endl;
menu(strA,strB);
}


/* OUTPUT
  Enter first string : Able was I
Enter second string : demo
----------------------------------------------------------------------
1: Show address of each character    2: Concatenate without strcat    3:
Concatenate with strcat
4: Compare two strings               5: Length using pointers         6:
Lowercase to uppercase
7: Upper to lower case               8: No of vowels                  9: Reverse
the string
```



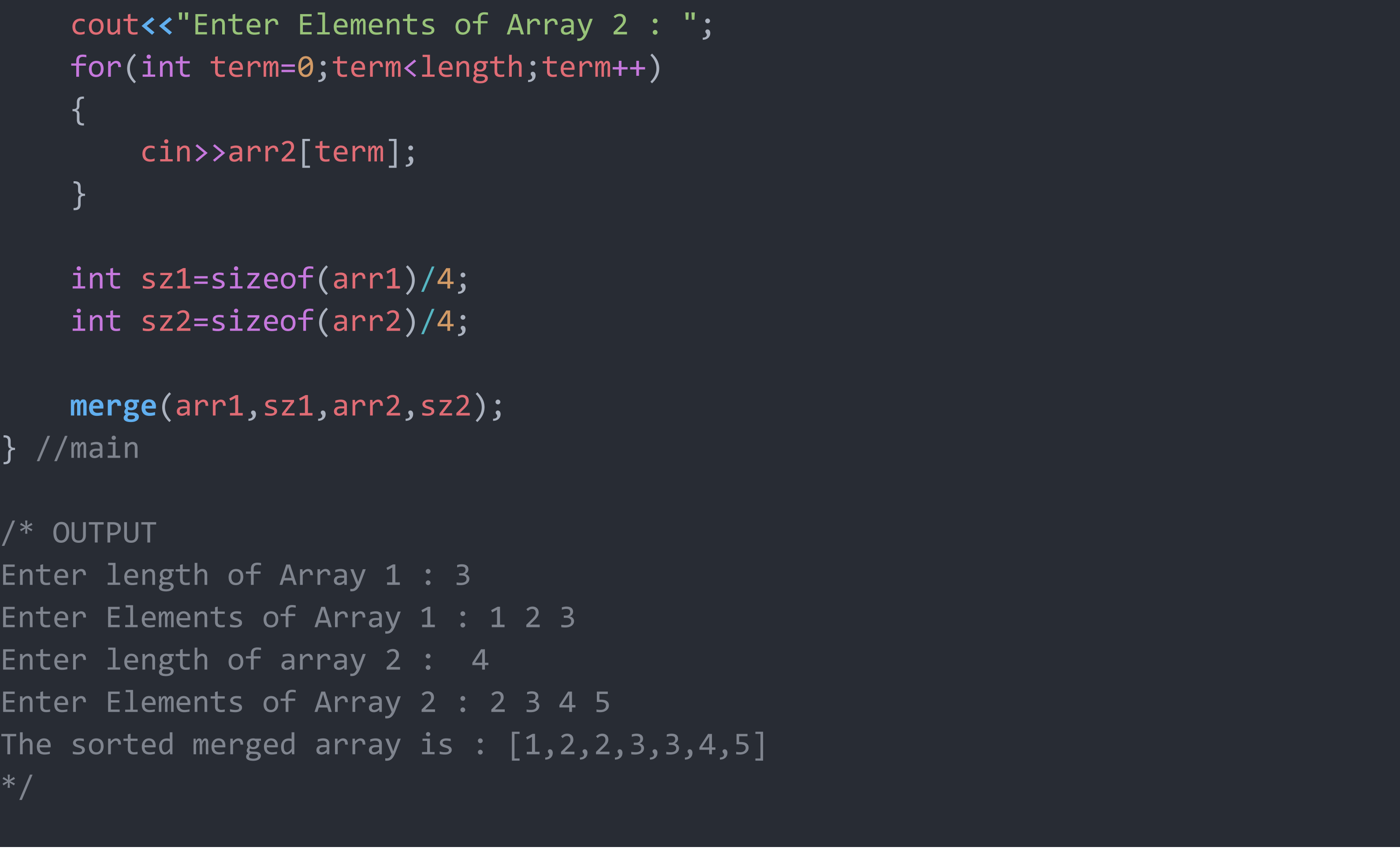
```
--------------------------------------------------------------------------------
------------------
What operation you want : 1
Adress of first string characters :
A 0x61fef8
b 0x61ff10
l 0x61ff28
e 0x61ff40
  0x61ff58
w 0x61ff70
a 0x61ff88
s 0x61ffa0
  0x61ffb8
I 0x61ffd0
What operation you want : 2
Concatenated without strcat : Able was Idemo
What operation you want : 3
Able was Idemo
What operation you want : 4
Not Equal Strings.
What operation you want : 5
Length of StrA = 10
What operation you want : 6
ABLE WAS I
What operation you want : 7
able was i
What operation you want : 8
Count of Vowels : 4
What operation you want : 9
Reversed String is :
I saw elbA
*/
```



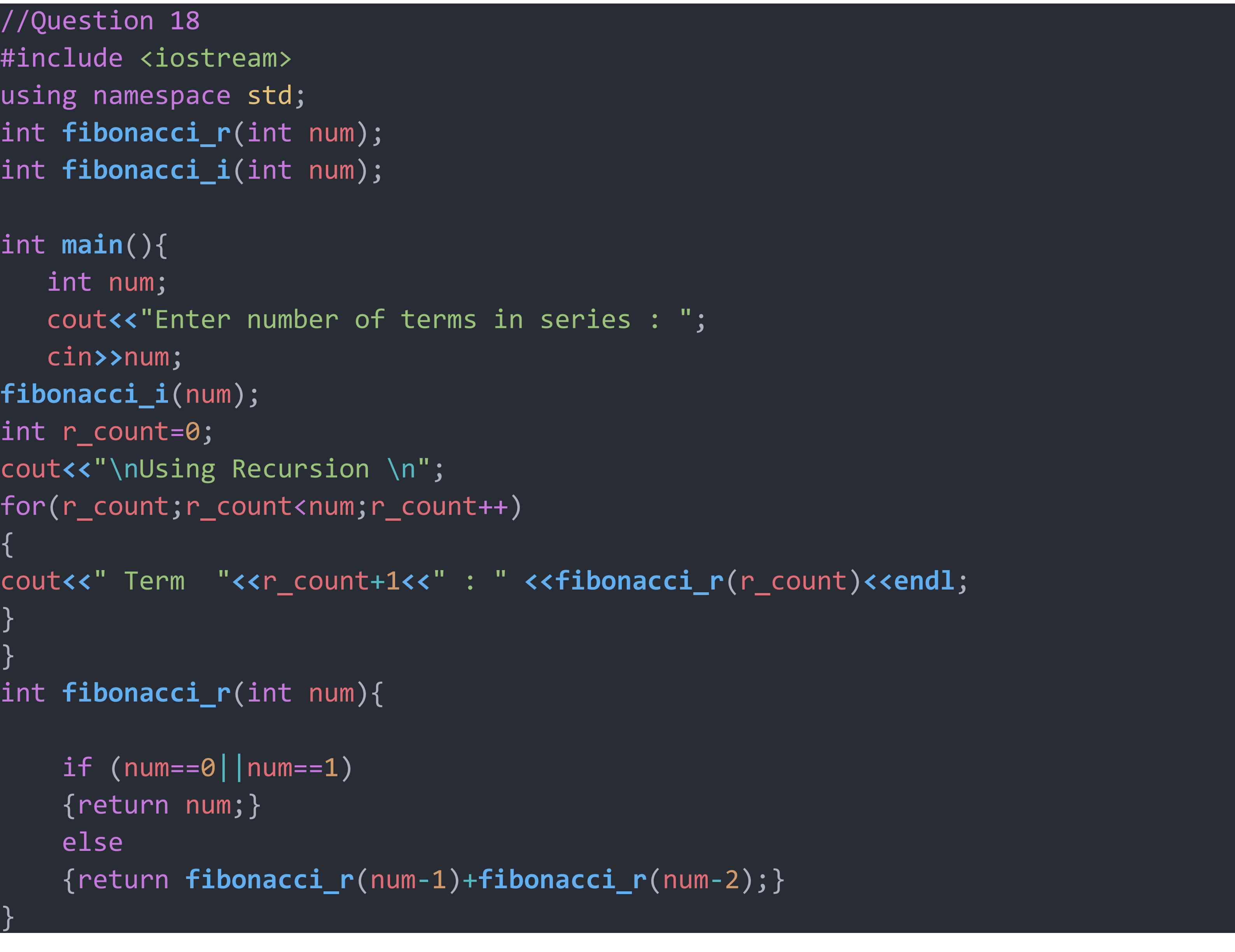
```
//Question 17
#include<iostream>
using namespace std;

void print(int array[],int length)
{
    cout<< "The sorted merged array is : ";
    cout<<"[";
    cout<<array[0];
    for(int elem=1;elem<length;elem++){
        cout<<','<<array[elem];
    }
    cout<<"]";
```



```
}//print

int merge(int arr1[],int sz1,int arr2[],int sz2)
{
    // arr1 = 1,2,3
    // arr2 = 4,2,5,6
    // arr3 = 0,0,0,0,0,0,0
    int arr3[sz1+sz2]={};
    int p1=0,p2=0,p3=0;

    while(p1<sz1 && p2<sz2)
    {
        if(arr1[p1]>arr2[p2])
            arr3[p3++]=arr2[p2++];
        else
            arr3[p3++]=arr1[p1++];

    }
    while(p1<sz1)
    {
        arr3[p3++]=arr1[p1++];
    }

    while(p2<sz2)
    {
        arr3[p3++]=arr2[p2++];
    }
    print(arr3,sz1+sz2);

}//merge

int main()
{

    int length;
    cout<<"Enter length of Array 1 : ";
    cin>>length;
    int arr1[length]={};
    cout<<"Enter Elements of Array 1 : ";
    for(int term=0;term<length;term++)
    {
        cin>>arr1[term];
    }
    cout<<"Enter length of array 2 :  ";
    cin>>length;
    int arr2[length]={};
```

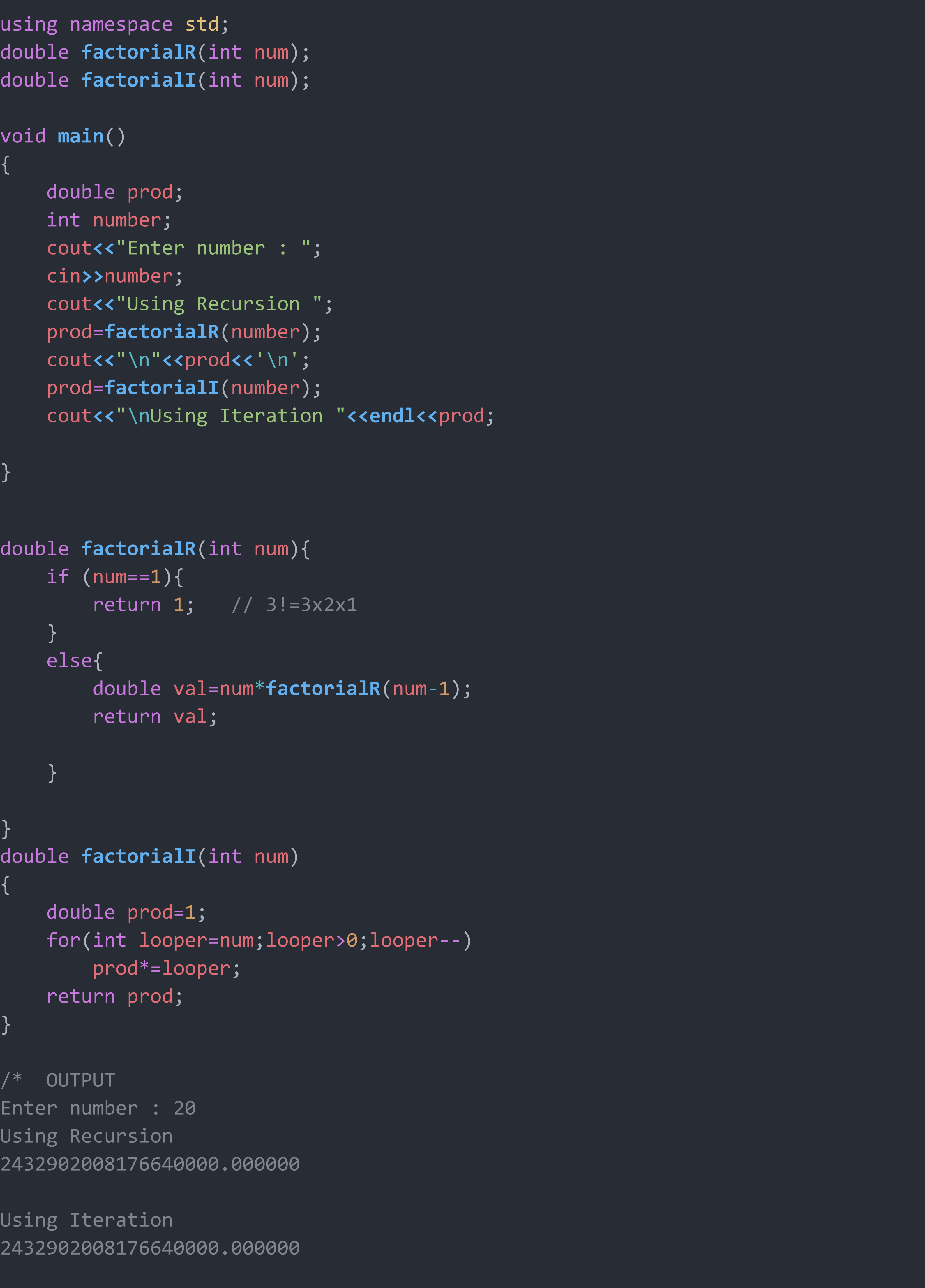


```
    cout<<"Enter Elements of Array 2 : ";
    for(int term=0;term<length;term++)
    {
        cin>>arr2[term];
    }

    int sz1=sizeof(arr1)/4;
    int sz2=sizeof(arr2)/4;

    merge(arr1,sz1,arr2,sz2);
} //main

/* OUTPUT
Enter length of Array 1 : 3
Enter Elements of Array 1 : 1 2 3
Enter length of array 2 :  4
Enter Elements of Array 2 : 2 3 4 5
The sorted merged array is : [1,2,2,3,3,4,5]
*/
```



```
//Question 18
#include <iostream>
using namespace std;
int fibonacci_r(int num);
int fibonacci_i(int num);

int main(){
   int num;
   cout<<"Enter number of terms in series : ";
   cin>>num;
fibonacci_i(num);
int r_count=0;
cout<<"\nUsing Recursion \n";
for(r_count;r_count<num;r_count++)
{
cout<<" Term  "<<r_count+1<<" : " <<fibonacci_r(r_count)<<endl;
}
}
int fibonacci_r(int num){

    if (num==0||num==1)
    {return num;}
    else
    {return fibonacci_r(num-1)+fibonacci_r(num-2);}
}
```

```
int fibonacci_i(int num){
    int num1=0; // 0 1 1 2 3 5 8 13
    int num2=1;
    cout<<"Using Iteration \n";
    cout<<"Term 1 : "<<num1<<endl<<"Term 2 : "<<num2<<endl;
    int cnt=2;
    for (cnt;cnt<num; ){
    int temp1=num1;
    int temp2=num2;

    num1=temp2;
    num2=temp2+temp1;
    cout<<"Term "<<++cnt<<" : "<<num2<<endl;

    }


}

/*
Enter number of terms in series : 7
Using Iteration
Term 1 : 0
Term 2 : 1
Term 3 : 1
Term 4 : 2
Term 5 : 3
Term 6 : 5
Term 7 : 8

Using Recursion
 Term  1 : 0
 Term  2 : 1
 Term  3 : 1
 Term  4 : 2
 Term  5 : 3
 Term  6 : 5
 Term  7 : 8
 */



//Question 19

#include <iostream>
```

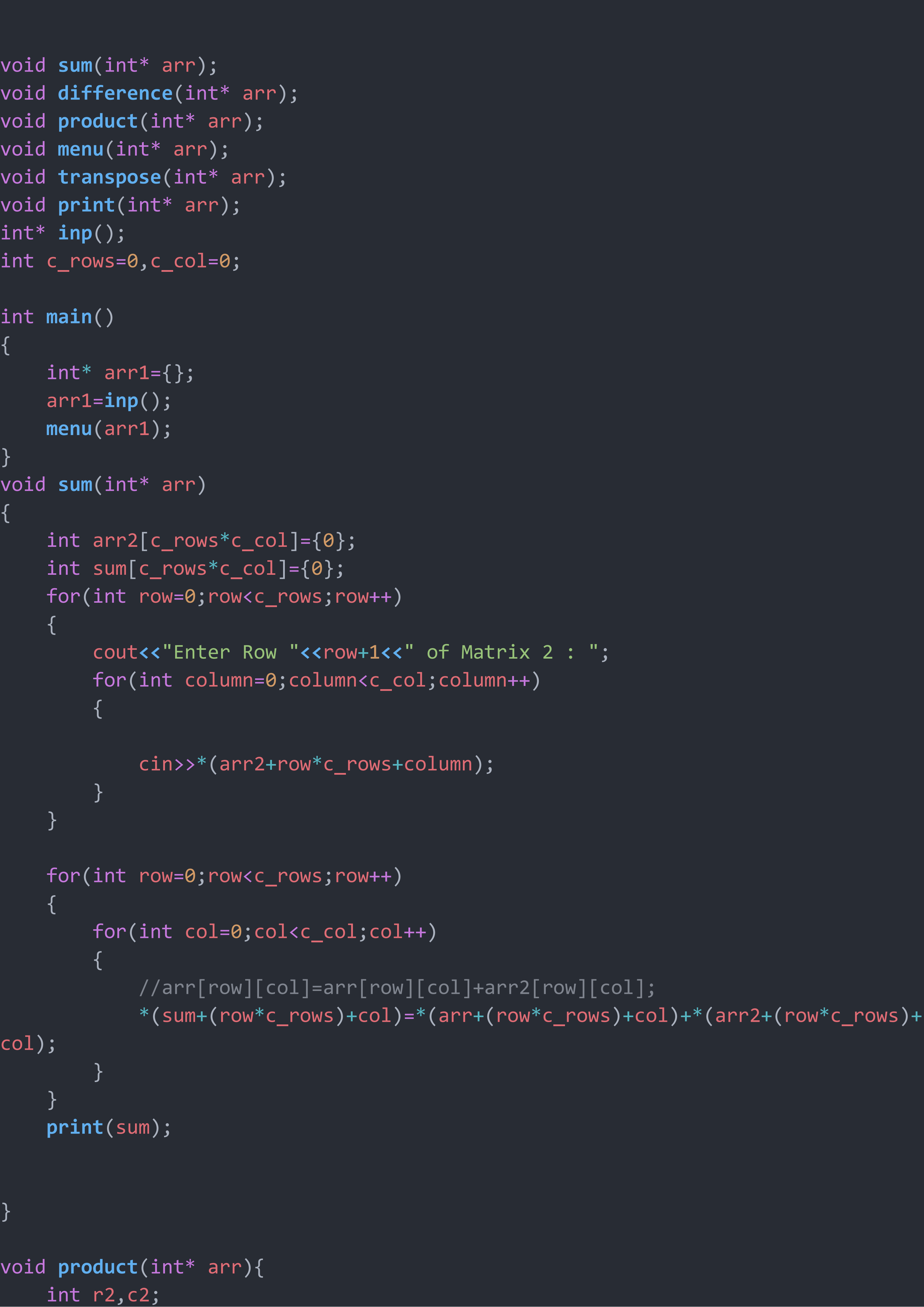


```cpp
using namespace std;
double factorialR(int num);
double factorialI(int num);

void main()
{
    double prod;
    int number;
    cout<<"Enter number : ";
    cin>>number;
    cout<<"Using Recursion ";
    prod=factorialR(number);
    cout<<"\n"<<prod<<'\n';
    prod=factorialI(number);
    cout<<"\nUsing Iteration "<<endl<<prod;

}


double factorialR(int num){
    if (num==1){
        return 1;   // 3!=3x2x1
    }
    else{
        double val=num*factorialR(num-1);
        return val;

    }

}
double factorialI(int num)
{
    double prod=1;
    for(int looper=num;looper>0;looper--)
        prod*=looper;
    return prod;
}

/*  OUTPUT
Enter number : 20
Using Recursion
2432902008176640000.000000

Using Iteration
2432902008176640000.000000
```

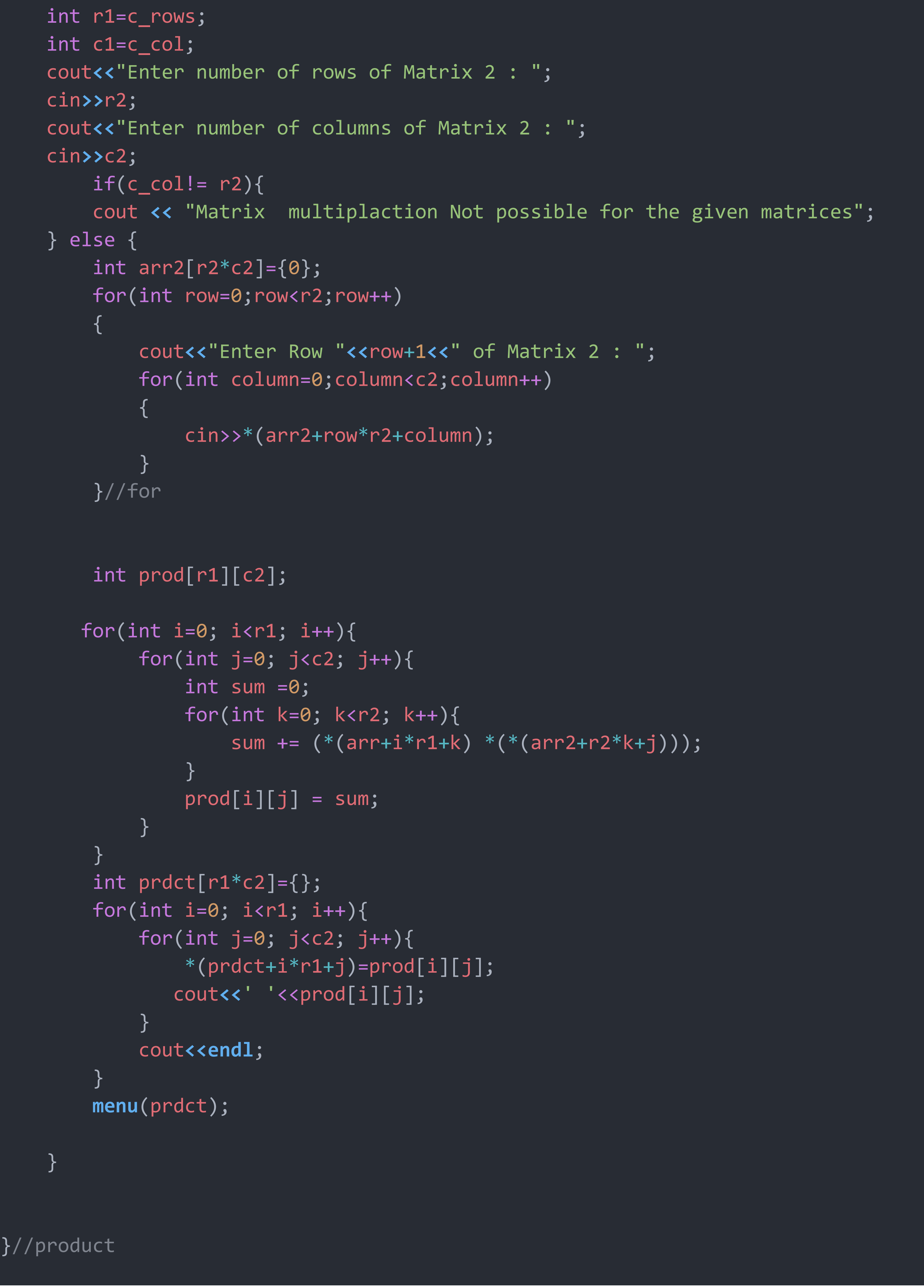
```
*/
```

```
//Question 20
#include <iostream>
using namespace std;

int GCD_r(int num1,int num2)
{
    if(num2==0)
        return num1;
    return GCD_r(num2,(num1%num2));
}

int GCD_i(int num1, int num2)
{
    while(num1%num2!=0 )
    {
        int temp=num1;
        num1=num2;
        num2=(temp%num1);
    }
    return num2;
}

int main()

{
    int num1,num2;
    cout<<"Enter <num1>,<num2> , first bigger then smaller number : ";
    cin>> num1>>num2;
    int gcd_r=GCD_r(num1,num2);
    int gcd_i=GCD_i(num1,num2);
    printf("\nGCD  of %d, %d via recusrion is %d  ",num1,num2,gcd_r);
    printf("\nGCD of %d, %d via iteration is  %d ",num1,num2,gcd_i);
}

/* OUTPUT
Enter <num1>,<num2> , first bigger then smaller number : 117 13

GCD  of 117, 13 via recusrion is 13
GCD of 117, 13 via iteration is  13
*/
```

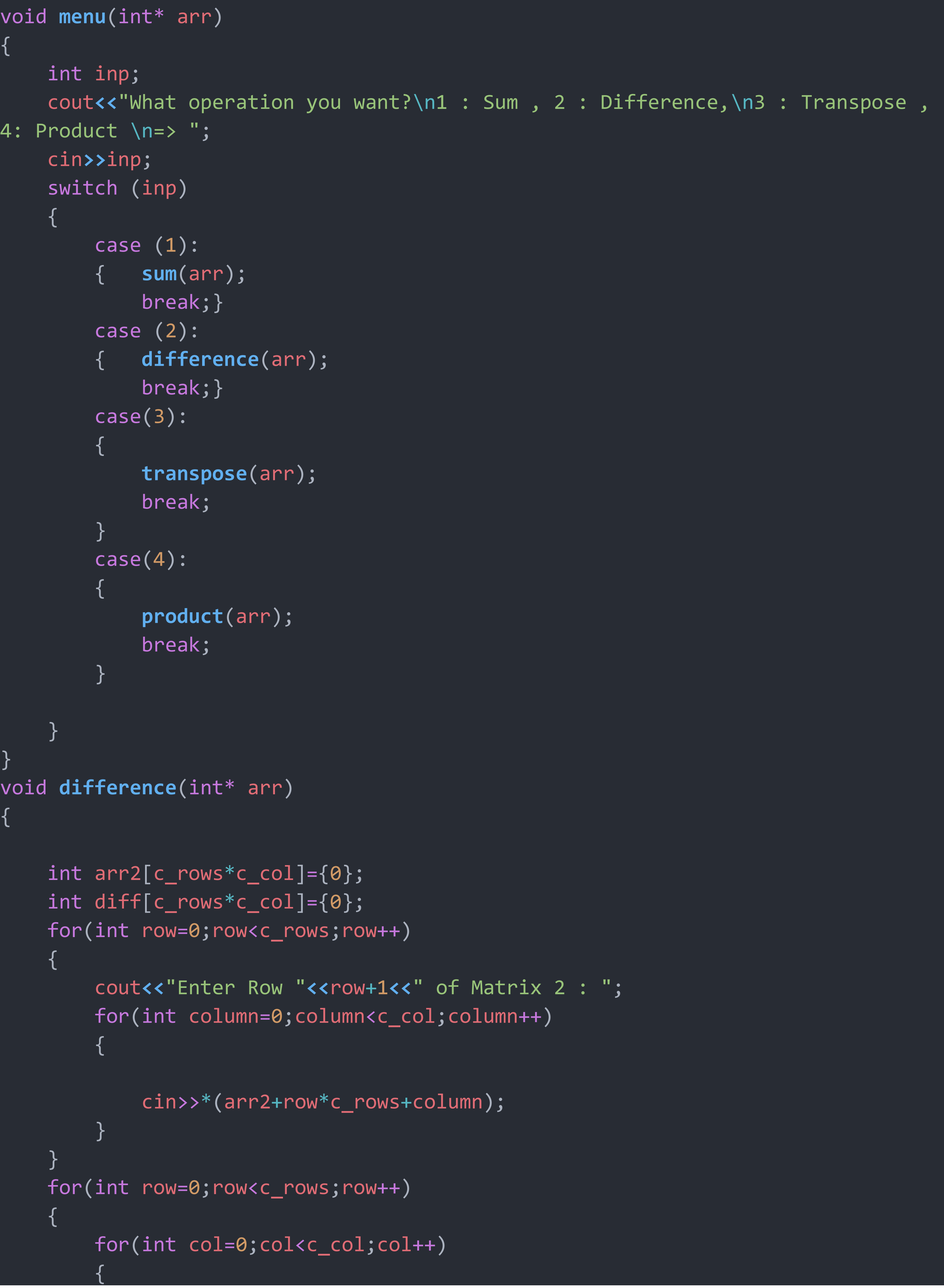
```
//Question 21
#include <iostream>
using namespace std;
```



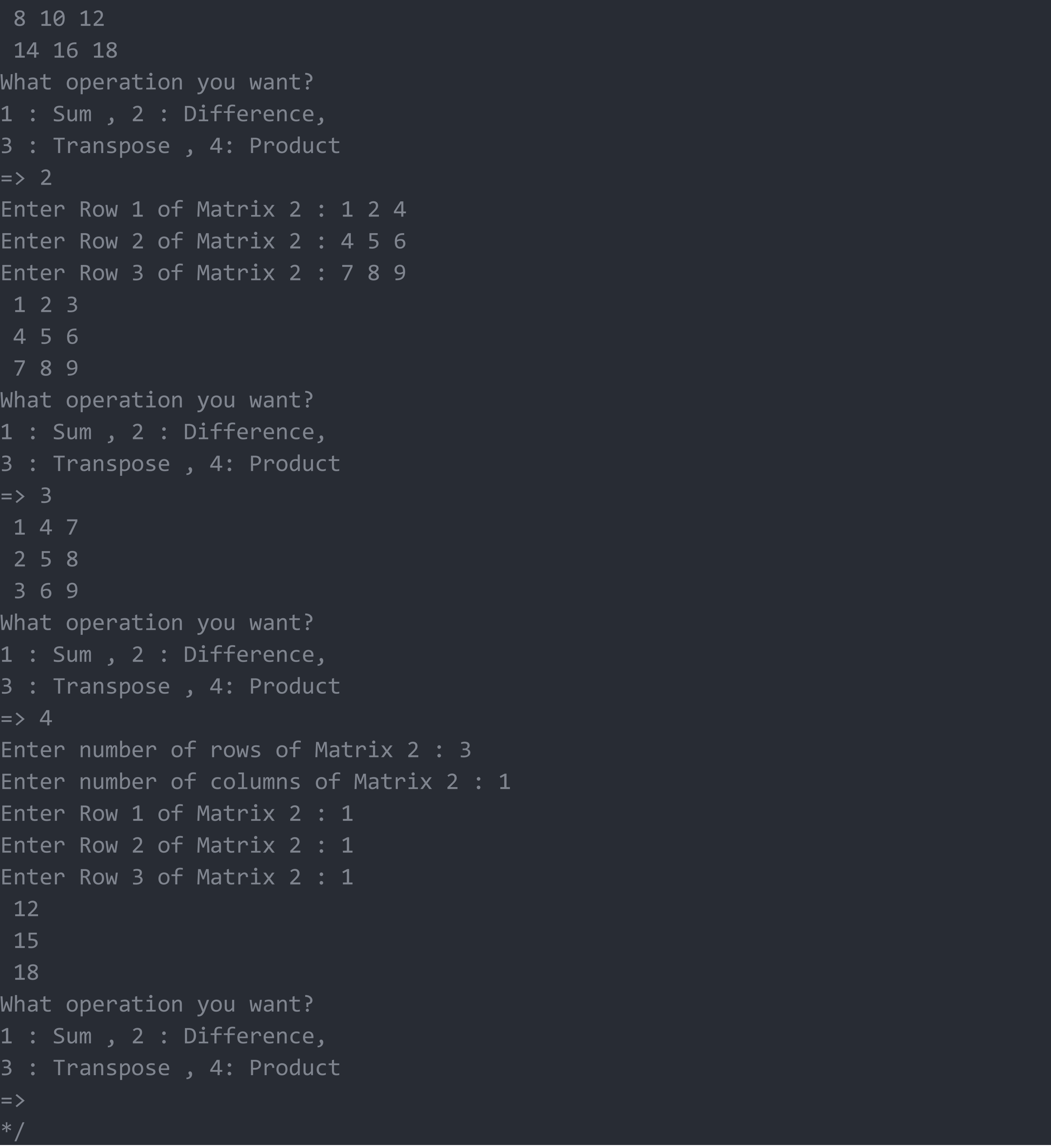
```
void sum(int* arr);
void difference(int* arr);
void product(int* arr);
void menu(int* arr);
void transpose(int* arr);
void print(int* arr);
int* inp();
int c_rows=0,c_col=0;

int main()
{
    int* arr1={};
    arr1=inp();
    menu(arr1);
}
void sum(int* arr)
{
    int arr2[c_rows*c_col]={0};
    int sum[c_rows*c_col]={0};
    for(int row=0;row<c_rows;row++)
    {
        cout<<"Enter Row "<<row+1<<" of Matrix 2 : ";
        for(int column=0;column<c_col;column++)
        {

            cin>>*(arr2+row*c_rows+column);
        }
    }

    for(int row=0;row<c_rows;row++)
    {
        for(int col=0;col<c_col;col++)
        {
            //arr[row][col]=arr[row][col]+arr2[row][col];
            *(sum+(row*c_rows)+col)=*(arr+(row*c_rows)+col)+*(arr2+(row*c_rows)+
col);
        }
    }
    print(sum);


}

void product(int* arr){
    int r2,c2;
```



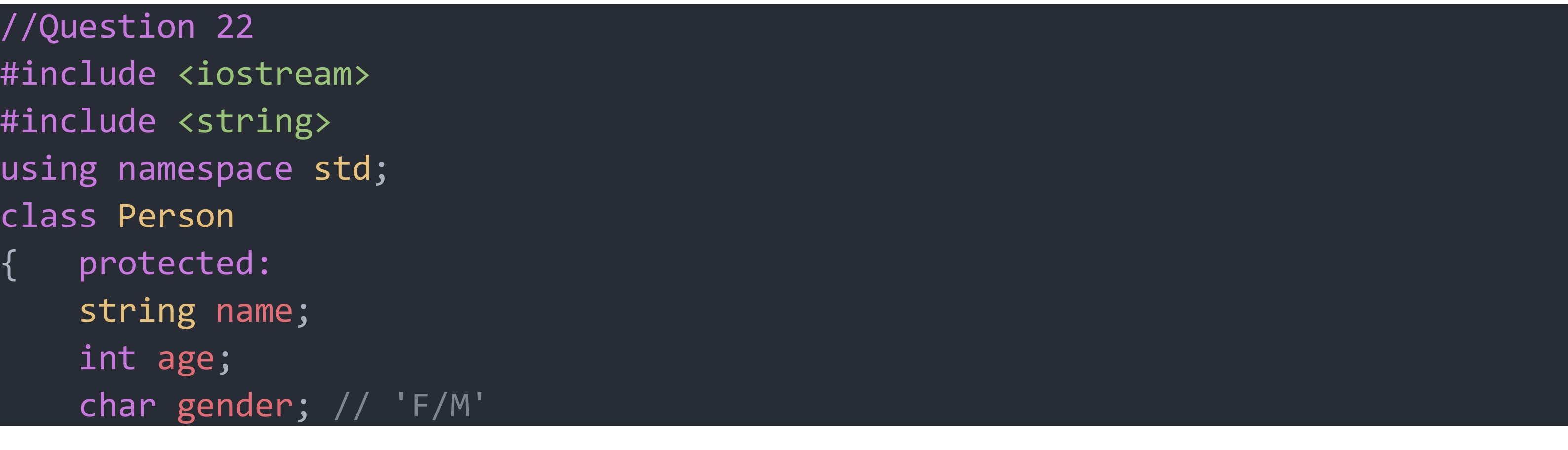
```
    int r1=c_rows;
    int c1=c_col;
    cout<<"Enter number of rows of Matrix 2 : ";
    cin>>r2;
    cout<<"Enter number of columns of Matrix 2 : ";
    cin>>c2;
        if(c_col!= r2){
        cout << "Matrix  multiplaction Not possible for the given matrices";
    } else {
        int arr2[r2*c2]={0};
        for(int row=0;row<r2;row++)
        {
            cout<<"Enter Row "<<row+1<<" of Matrix 2 : ";
            for(int column=0;column<c2;column++)
            {
                cin>>*(arr2+row*r2+column);
            }
        }//for


        int prod[r1][c2];

       for(int i=0; i<r1; i++){
            for(int j=0; j<c2; j++){
                int sum =0;
                for(int k=0; k<r2; k++){
                    sum += (*(arr+i*r1+k) *(*(arr2+r2*k+j)));
                }
                prod[i][j] = sum;
            }
        }
        int prdct[r1*c2]={};
        for(int i=0; i<r1; i++){
            for(int j=0; j<c2; j++){
                *(prdct+i*r1+j)=prod[i][j];
               cout<<' '<<prod[i][j];
            }
            cout<<endl;
        }
        menu(prdct);

    }


}//product
```



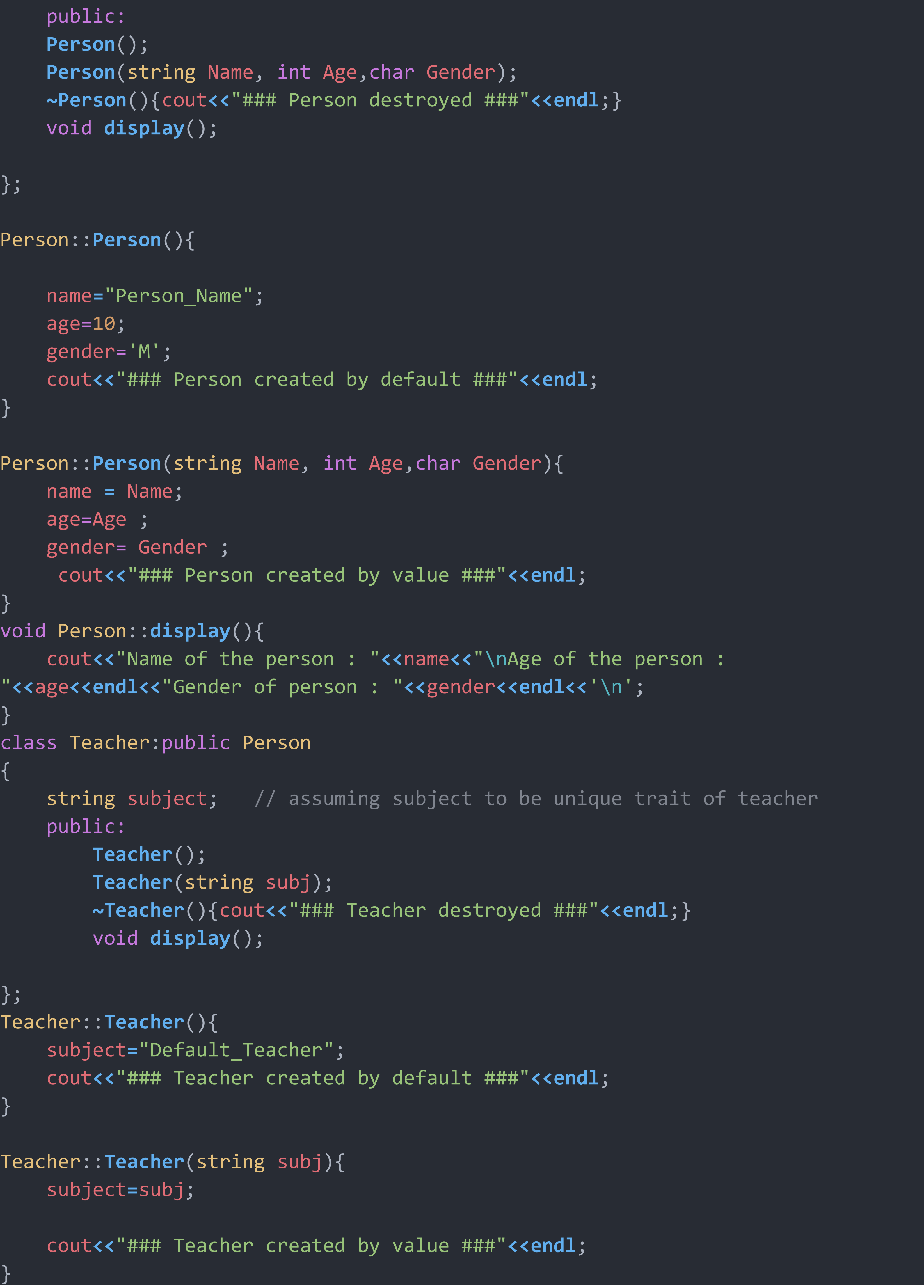
```
void menu(int* arr)
{
    int inp;
    cout<<"What operation you want?\n1 : Sum , 2 : Difference,\n3 : Transpose , 4: Product \n=> ";
    cin>>inp;
    switch (inp)
    {
        case (1):
        {   sum(arr);
            break;}
        case (2):
        {   difference(arr);
            break;}
        case(3):
        {
            transpose(arr);
            break;
        }
        case(4):
        {
            product(arr);
            break;
        }

    }
}
void difference(int* arr)
{

    int arr2[c_rows*c_col]={0};
    int diff[c_rows*c_col]={0};
    for(int row=0;row<c_rows;row++)
    {
        cout<<"Enter Row "<<row+1<<" of Matrix 2 : ";
        for(int column=0;column<c_col;column++)
        {

            cin>>*(arr2+row*c_rows+column);
        }
    }
    for(int row=0;row<c_rows;row++)
    {
        for(int col=0;col<c_col;col++)
        {
            *(diff+(row*c_rows)+col)=*(arr+(row*c_rows)+col)-
*(arr2+(row*c_rows)+col);
        }
    }
    print(diff);

}
void transpose(int* arr)
{
 int temp=0;
 for(int row=0;row<c_rows;row++)
 {
    for (int col=0;col<c_col;col++)


    {
        if(row<col)
        {
        temp=*(arr+(row*c_rows)+col);
        *(arr+(row*c_rows)+col)=*(arr+(col*c_col)+row);
        *(arr+(col*c_col)+row)=temp;
        }
    }

 }

 print(arr);

}



int* inp()

{

    cout<<"Enter no of rows in Matrix : ";
    cin>>c_rows;
    cout<<"Enter no of columns in Matrix : ";
    cin>>c_col;
    int arr[c_col*c_rows]={0};

    for(int row=0;row<c_rows;row++)
    {
        cout<<"Enter Row "<<row+1<<" : ";
        for(int column=0;column<c_col;column++)
        {
            cin>>*(arr+row*c_rows+column);
        }
    }


    cout<<"Entered Matrix is\n";
    print(&arr[0]);
    int* array=NULL;
    array=new int [c_rows*c_col];
    return (array);


}

void print(int* arr)
{

    for(int row=0;row<c_rows;row++){
        for(int col=0;col<c_col;col++)
        {
        cout<<' '<<*(arr+(row*c_rows)+col);
        }
        cout<<endl;
    }
    menu(arr);
}


/* OUTPUT
Enter no of rows in Matrix : 3
Enter no of columns in Matrix : 3
Enter Row 1 : 1 2 3
Enter Row 2 : 4 5 6
Enter Row 3 : 7 8 9
Entered Matrix is
 1 2 3
 4 5 6
 7 8 9
What operation you want?
1 : Sum , 2 : Difference,
3 : Transpose , 4: Product
=> 1
Enter Row 1 of Matrix 2 : 1 2 4
Enter Row 2 of Matrix 2 : 4 5 6
Enter Row 3 of Matrix 2 : 7 8 9
 2 4 7
```



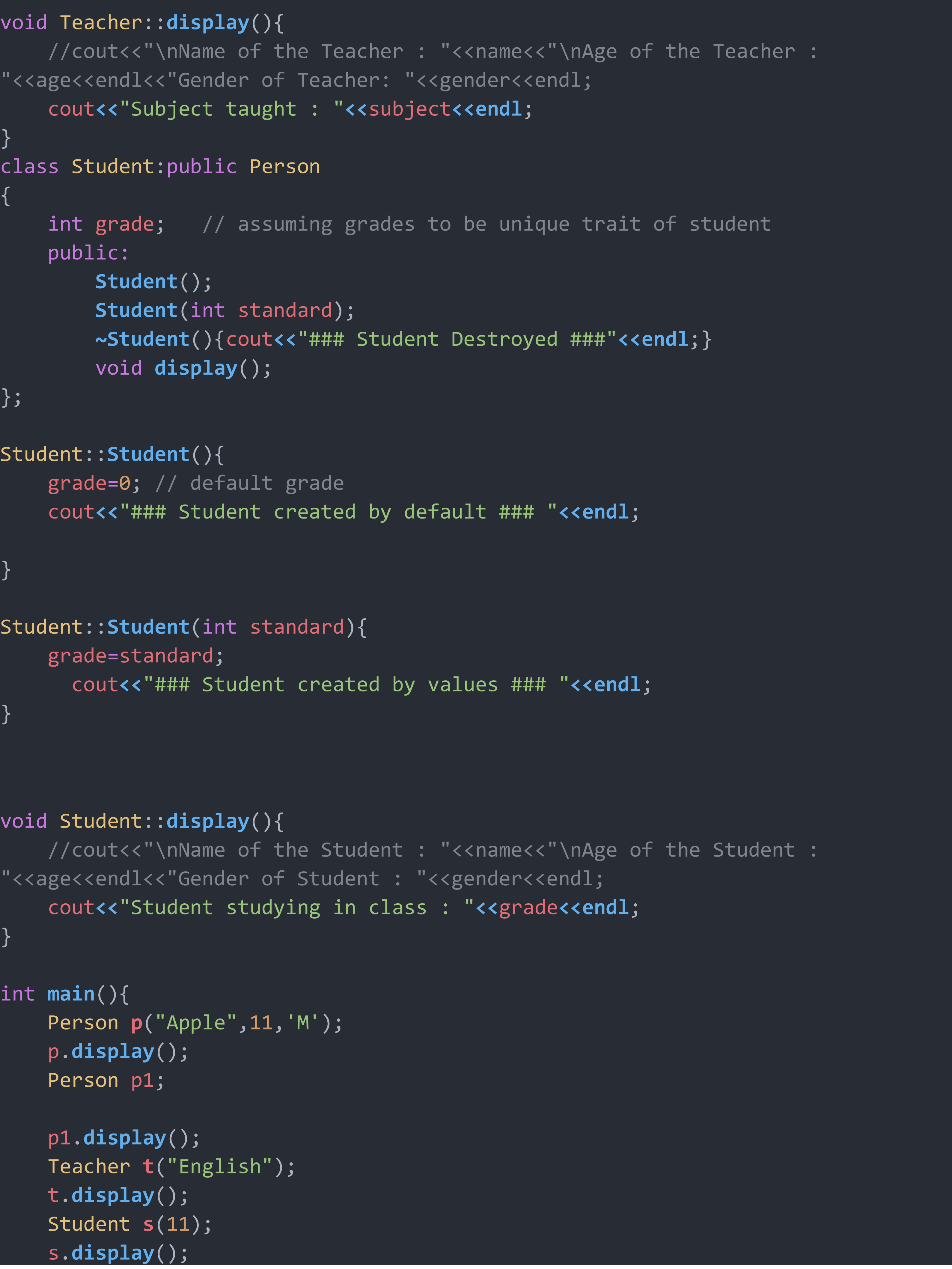
```
 8 10 12
 14 16 18
What operation you want?
1 : Sum , 2 : Difference,
3 : Transpose , 4: Product
=> 2
Enter Row 1 of Matrix 2 : 1 2 4
Enter Row 2 of Matrix 2 : 4 5 6
Enter Row 3 of Matrix 2 : 7 8 9
 1 2 3
 4 5 6
 7 8 9
What operation you want?
1 : Sum , 2 : Difference,
3 : Transpose , 4: Product
=> 3
 1 4 7
 2 5 8
 3 6 9
What operation you want?
1 : Sum , 2 : Difference,
3 : Transpose , 4: Product
=> 4
Enter number of rows of Matrix 2 : 3
Enter number of columns of Matrix 2 : 1
Enter Row 1 of Matrix 2 : 1
Enter Row 2 of Matrix 2 : 1
Enter Row 3 of Matrix 2 : 1
 12
 15
 18
What operation you want?
1 : Sum , 2 : Difference,
3 : Transpose , 4: Product
=>
*/
```



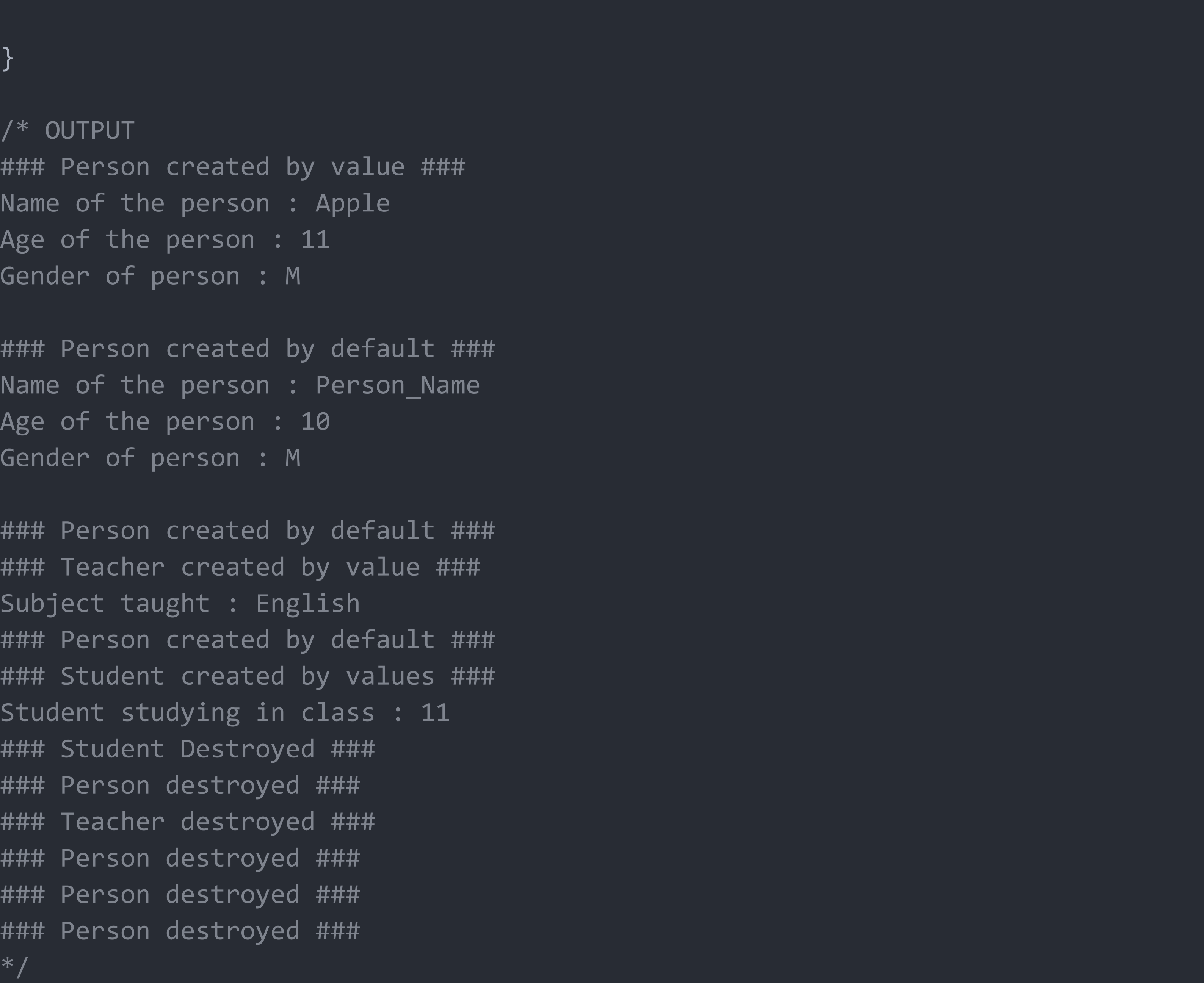
```
//Question 22
#include <iostream>
#include <string>
using namespace std;
class Person
{   protected:
    string name;
    int age;
    char gender; // 'F/M'
```



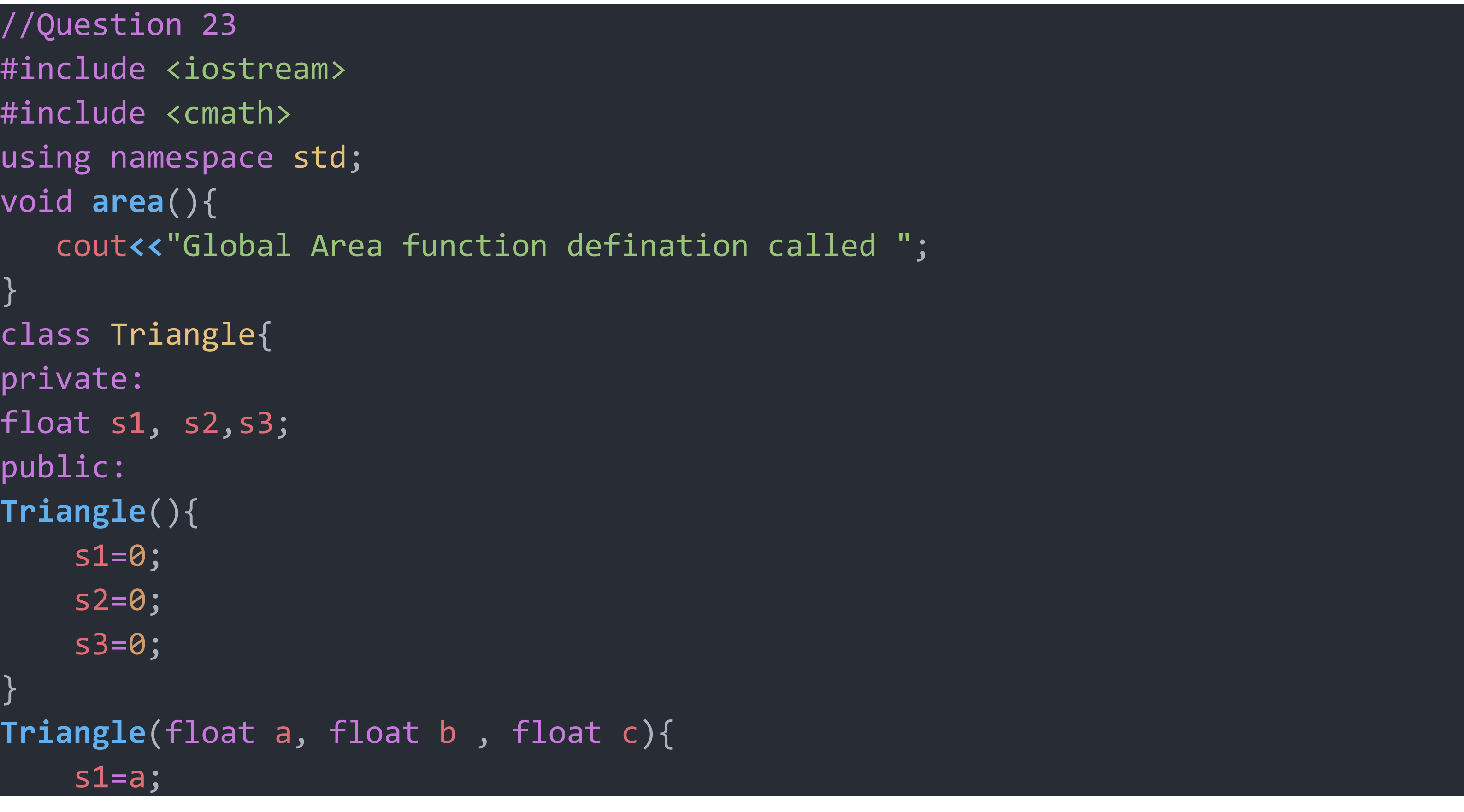
```cpp
    public:
    Person();
    Person(string Name, int Age,char Gender);
    ~Person(){cout<<"### Person destroyed ###"<<endl;}
    void display();

};

Person::Person(){

    name="Person_Name";
    age=10;
    gender='M';
    cout<<"### Person created by default ###"<<endl;
}

Person::Person(string Name, int Age,char Gender){
    name = Name;
    age=Age ;
    gender= Gender ;
     cout<<"### Person created by value ###"<<endl;
}
void Person::display(){
    cout<<"Name of the person : "<<name<<"\nAge of the person :
"<<age<<endl<<"Gender of person : "<<gender<<endl<<'\n';
}
class Teacher:public Person
{
    string subject;   // assuming subject to be unique trait of teacher
    public:
        Teacher();
        Teacher(string subj);
        ~Teacher(){cout<<"### Teacher destroyed ###"<<endl;}
        void display();

};
Teacher::Teacher(){
    subject="Default_Teacher";
    cout<<"### Teacher created by default ###"<<endl;
}

Teacher::Teacher(string subj){
    subject=subj;

    cout<<"### Teacher created by value ###"<<endl;
}
```



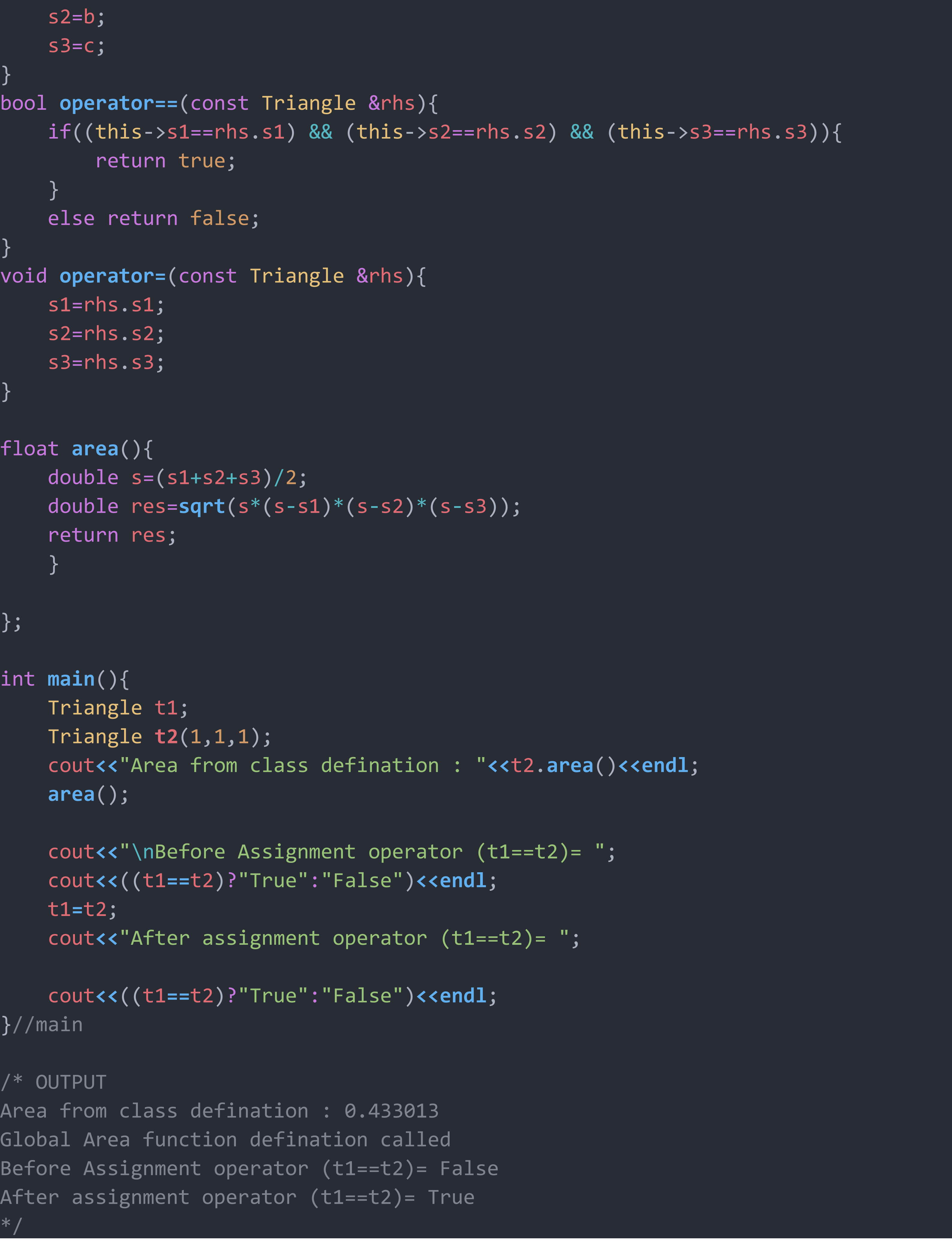
```
void Teacher::display(){
    //cout<<"\nName of the Teacher : "<<name<<"\nAge of the Teacher :
"<<age<<endl<<"Gender of Teacher: "<<gender<<endl;
    cout<<"Subject taught : "<<subject<<endl;
}
class Student:public Person
{
    int grade;   // assuming grades to be unique trait of student
    public:
        Student();
        Student(int standard);
        ~Student(){cout<<"### Student Destroyed ###"<<endl;}
        void display();
};

Student::Student(){
    grade=0; // default grade
    cout<<"### Student created by default ### "<<endl;

}

Student::Student(int standard){
    grade=standard;
      cout<<"### Student created by values ### "<<endl;
}



void Student::display(){
    //cout<<"\nName of the Student : "<<name<<"\nAge of the Student :
"<<age<<endl<<"Gender of Student : "<<gender<<endl;
    cout<<"Student studying in class : "<<grade<<endl;
}

int main(){
    Person p("Apple",11,'M');
    p.display();
    Person p1;

    p1.display();
    Teacher t("English");
    t.display();
    Student s(11);
    s.display();
```



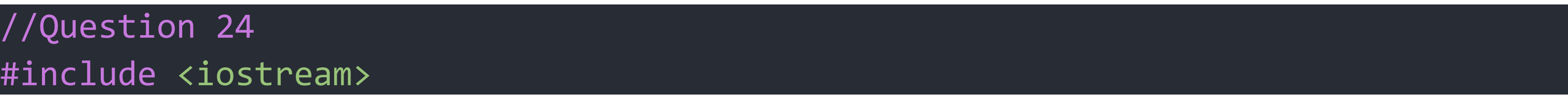
```
}

/* OUTPUT
### Person created by value ###
Name of the person : Apple
Age of the person : 11
Gender of person : M

### Person created by default ###
Name of the person : Person_Name
Age of the person : 10
Gender of person : M

### Person created by default ###
### Teacher created by value ###
Subject taught : English
### Person created by default ###
### Student created by values ###
Student studying in class : 11
### Student Destroyed ###
### Person destroyed ###
### Teacher destroyed ###
### Person destroyed ###
### Person destroyed ###
### Person destroyed ###
*/
```



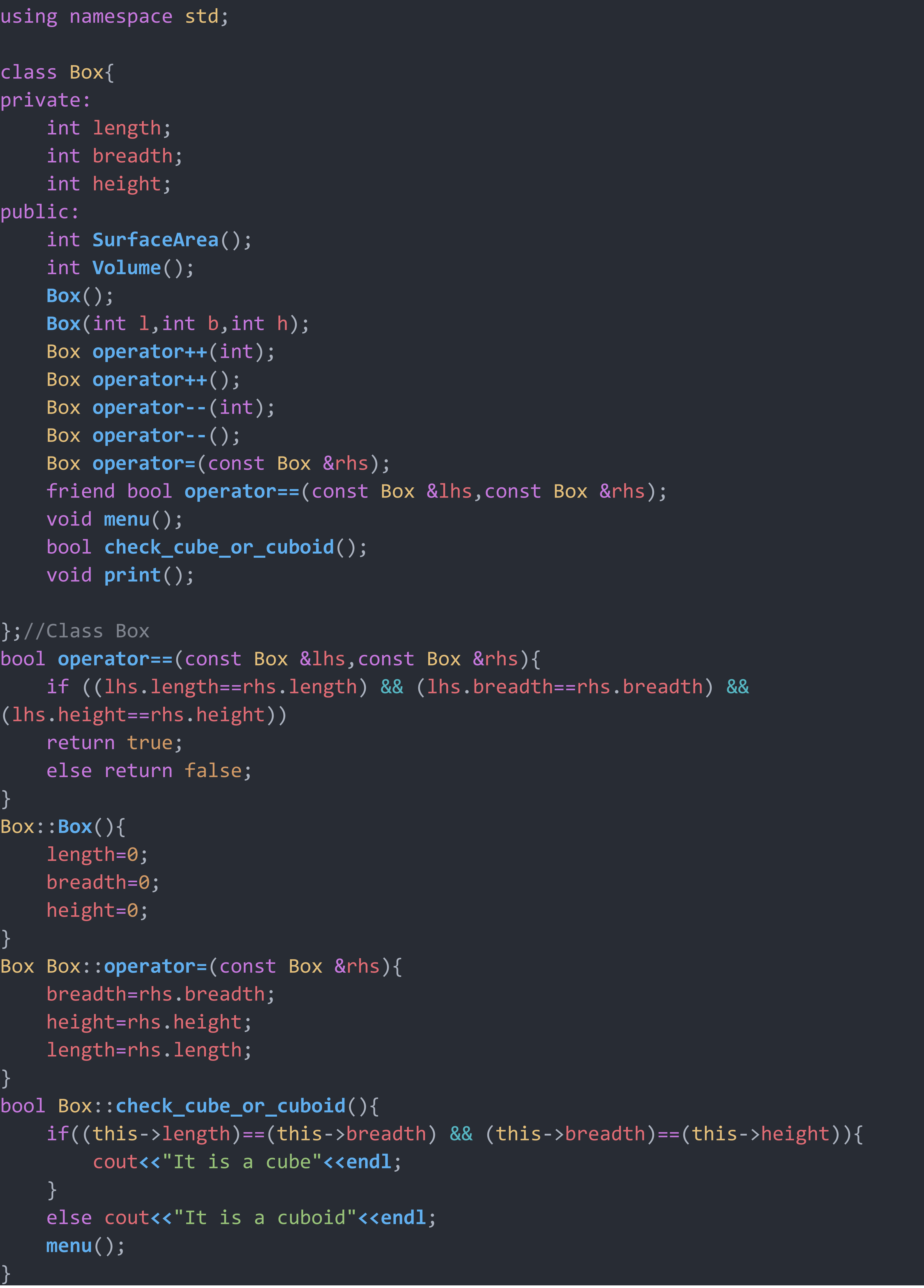
```
//Question 23
#include <iostream>
#include <cmath>
using namespace std;
void area(){
   cout<<"Global Area function defination called ";
}
class Triangle{
private:
float s1, s2,s3;
public:
Triangle(){
    s1=0;
    s2=0;
    s3=0;
}
Triangle(float a, float b , float c){
    s1=a;
```



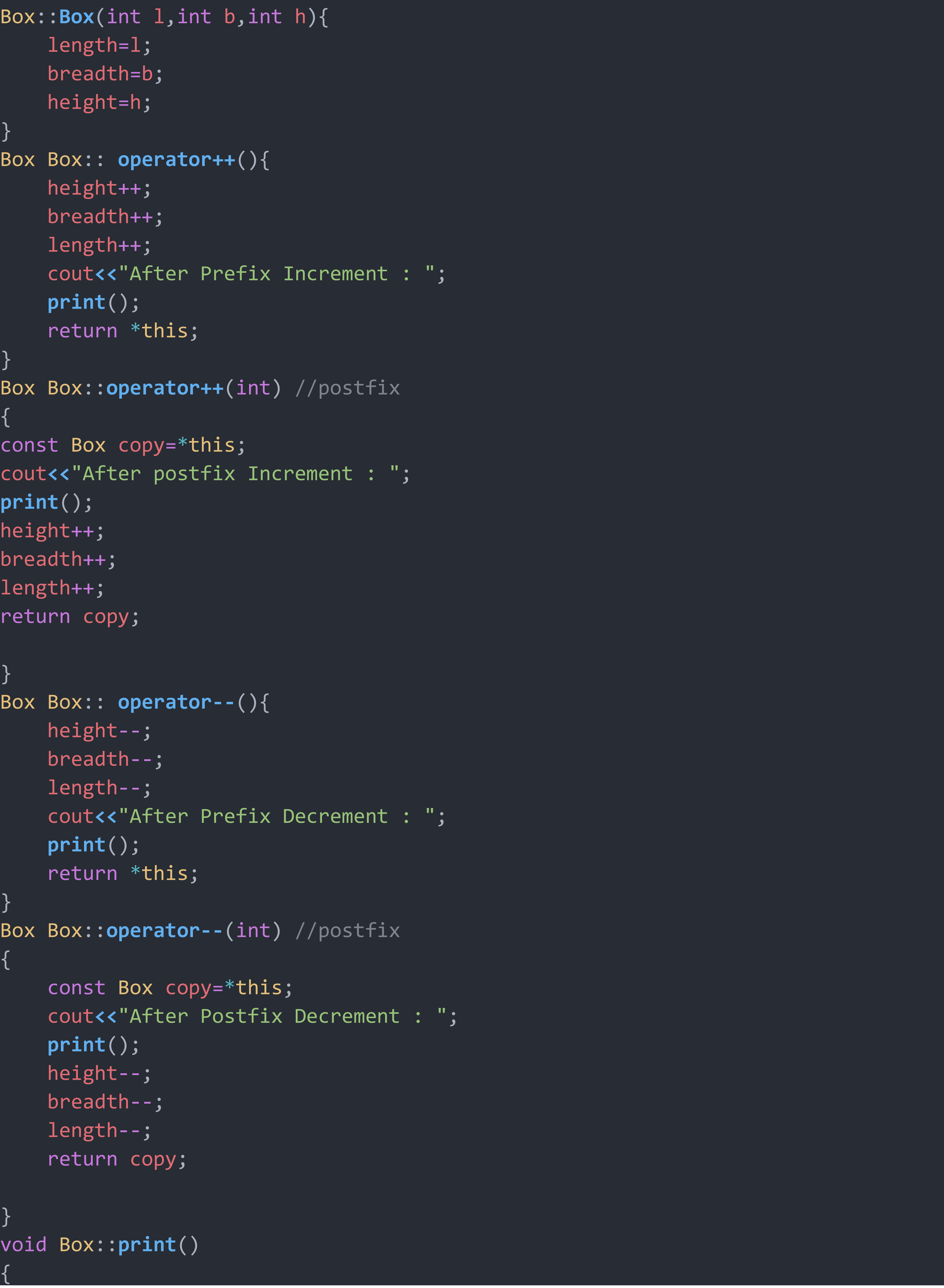
```
    s2=b;
    s3=c;
}
bool operator==(const Triangle &rhs){
    if((this->s1==rhs.s1) && (this->s2==rhs.s2) && (this->s3==rhs.s3)){
        return true;
    }
    else return false;
}
void operator=(const Triangle &rhs){
    s1=rhs.s1;
    s2=rhs.s2;
    s3=rhs.s3;
}

float area(){
    double s=(s1+s2+s3)/2;
    double res=sqrt(s*(s-s1)*(s-s2)*(s-s3));
    return res;
    }

};

int main(){
    Triangle t1;
    Triangle t2(1,1,1);
    cout<<"Area from class defination : "<<t2.area()<<endl;
    area();

    cout<<"\nBefore Assignment operator (t1==t2)= ";
    cout<<((t1==t2)?"True":"False")<<endl;
    t1=t2;
    cout<<"After assignment operator (t1==t2)= ";

    cout<<((t1==t2)?"True":"False")<<endl;
}//main

/* OUTPUT
Area from class defination : 0.433013
Global Area function defination called
Before Assignment operator (t1==t2)= False
After assignment operator (t1==t2)= True
*/
```



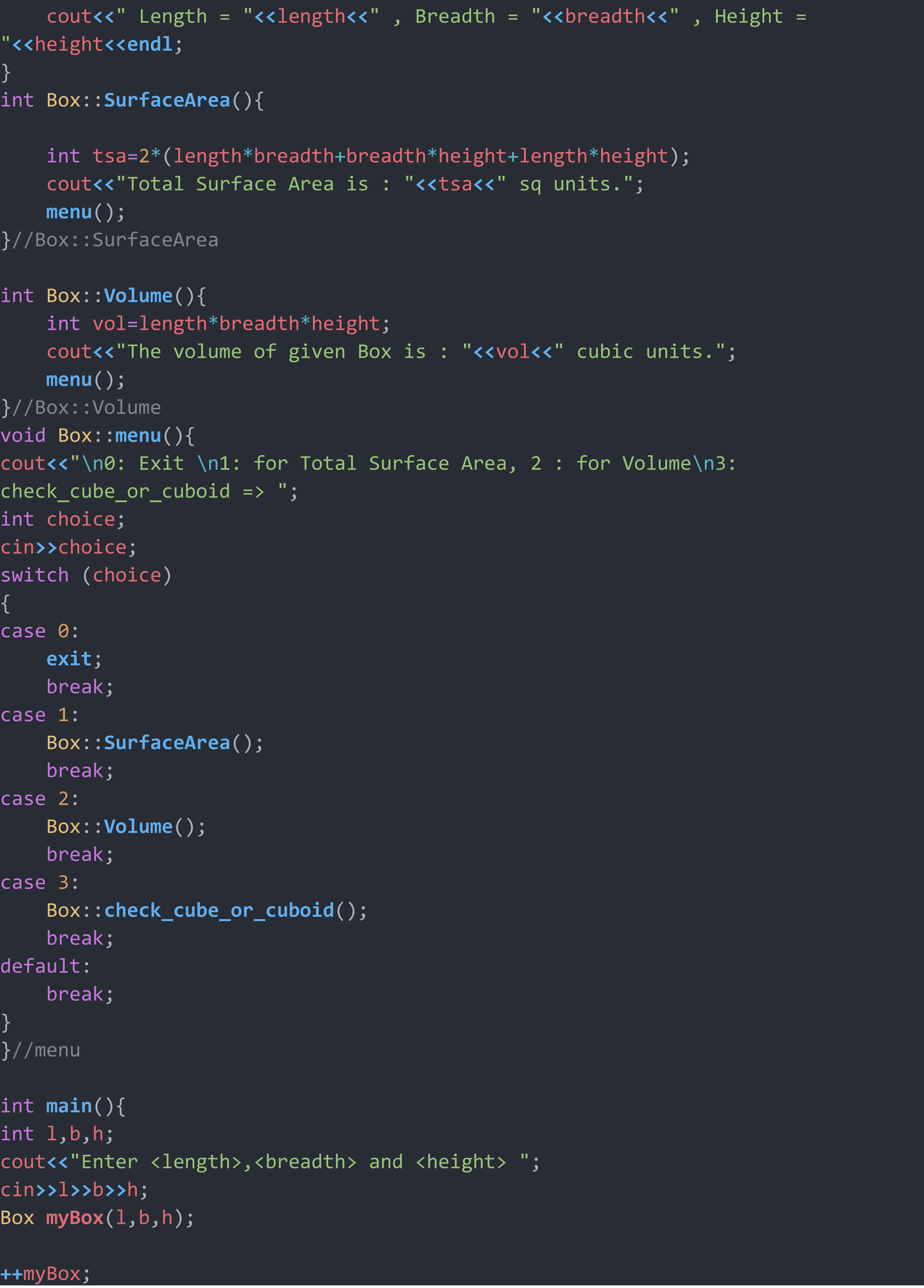
```
//Question 24
#include <iostream>
```



```
using namespace std;

class Box{
private:
    int length;
    int breadth;
    int height;
public:
    int SurfaceArea();
    int Volume();
    Box();
    Box(int l,int b,int h);
    Box operator++(int);
    Box operator++();
    Box operator--(int);
    Box operator--();
    Box operator=(const Box &rhs);
    friend bool operator==(const Box &lhs,const Box &rhs);
    void menu();
    bool check_cube_or_cuboid();
    void print();

};//Class Box
bool operator==(const Box &lhs,const Box &rhs){
    if ((lhs.length==rhs.length) && (lhs.breadth==rhs.breadth) &&
(lhs.height==rhs.height))
    return true;
    else return false;
}
Box::Box(){
    length=0;
    breadth=0;
    height=0;
}
Box Box::operator=(const Box &rhs){
    breadth=rhs.breadth;
    height=rhs.height;
    length=rhs.length;
}
bool Box::check_cube_or_cuboid(){
    if((this->length)==(this->breadth) && (this->breadth)==(this->height)){
        cout<<"It is a cube"<<endl;
    }
    else cout<<"It is a cuboid"<<endl;
    menu();
}
```

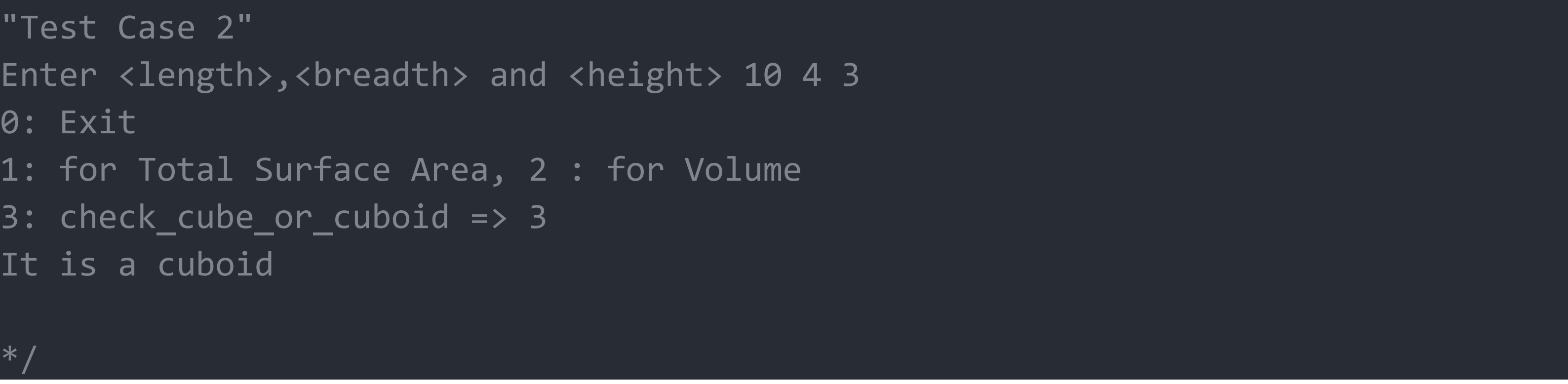


```
Box::Box(int l,int b,int h){
    length=l;
    breadth=b;
    height=h;
}
Box Box:: operator++(){
    height++;
    breadth++;
    length++;
    cout<<"After Prefix Increment : ";
    print();
    return *this;
}
Box Box::operator++(int) //postfix
{
const Box copy=*this;
cout<<"After postfix Increment : ";
print();
height++;
breadth++;
length++;
return copy;

}
Box Box:: operator--(){
    height--;
    breadth--;
    length--;
    cout<<"After Prefix Decrement : ";
    print();
    return *this;
}
Box Box::operator--(int) //postfix
{
    const Box copy=*this;
    cout<<"After Postfix Decrement : ";
    print();
    height--;
    breadth--;
    length--;
    return copy;

}
void Box::print()
{
```

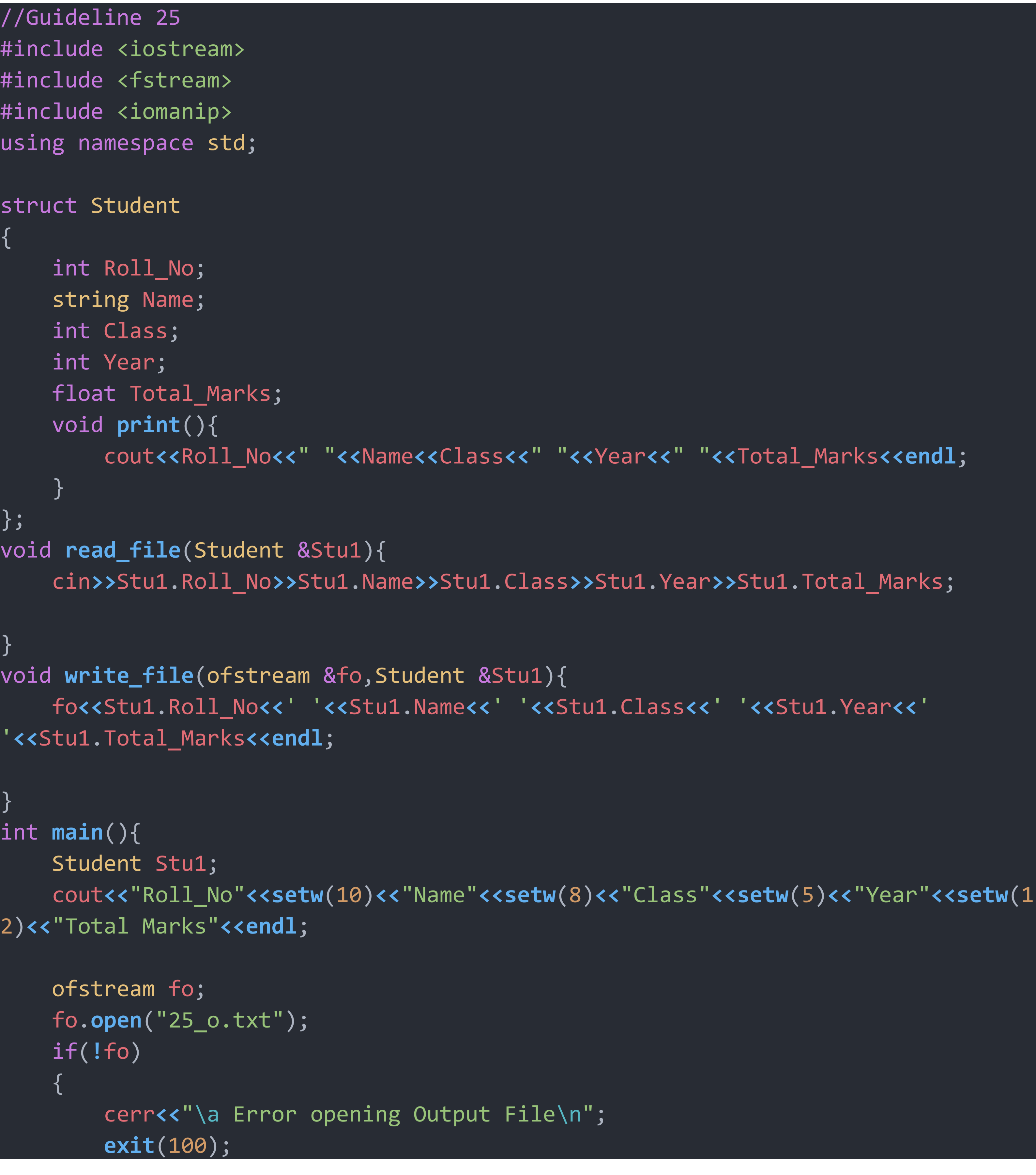


```
    cout<<" Length = "<<length<<" , Breadth = "<<breadth<<" , Height =
"<<height<<endl;
}
int Box::SurfaceArea(){

    int tsa=2*(length*breadth+breadth*height+length*height);
    cout<<"Total Surface Area is : "<<tsa<<" sq units.";
    menu();
}//Box::SurfaceArea

int Box::Volume(){
    int vol=length*breadth*height;
    cout<<"The volume of given Box is : "<<vol<<" cubic units.";
    menu();
}//Box::Volume
void Box::menu(){
cout<<"\n0: Exit \n1: for Total Surface Area, 2 : for Volume\n3:
check_cube_or_cuboid => ";
int choice;
cin>>choice;
switch (choice)
{
case 0:
    exit;
    break;
case 1:
    Box::SurfaceArea();
    break;
case 2:
    Box::Volume();
    break;
case 3:
    Box::check_cube_or_cuboid();
    break;
default:
    break;
}
}//menu

int main(){
int l,b,h;
cout<<"Enter <length>,<breadth> and <height> ";
cin>>l>>b>>h;
Box myBox(l,b,h);

++myBox;
```

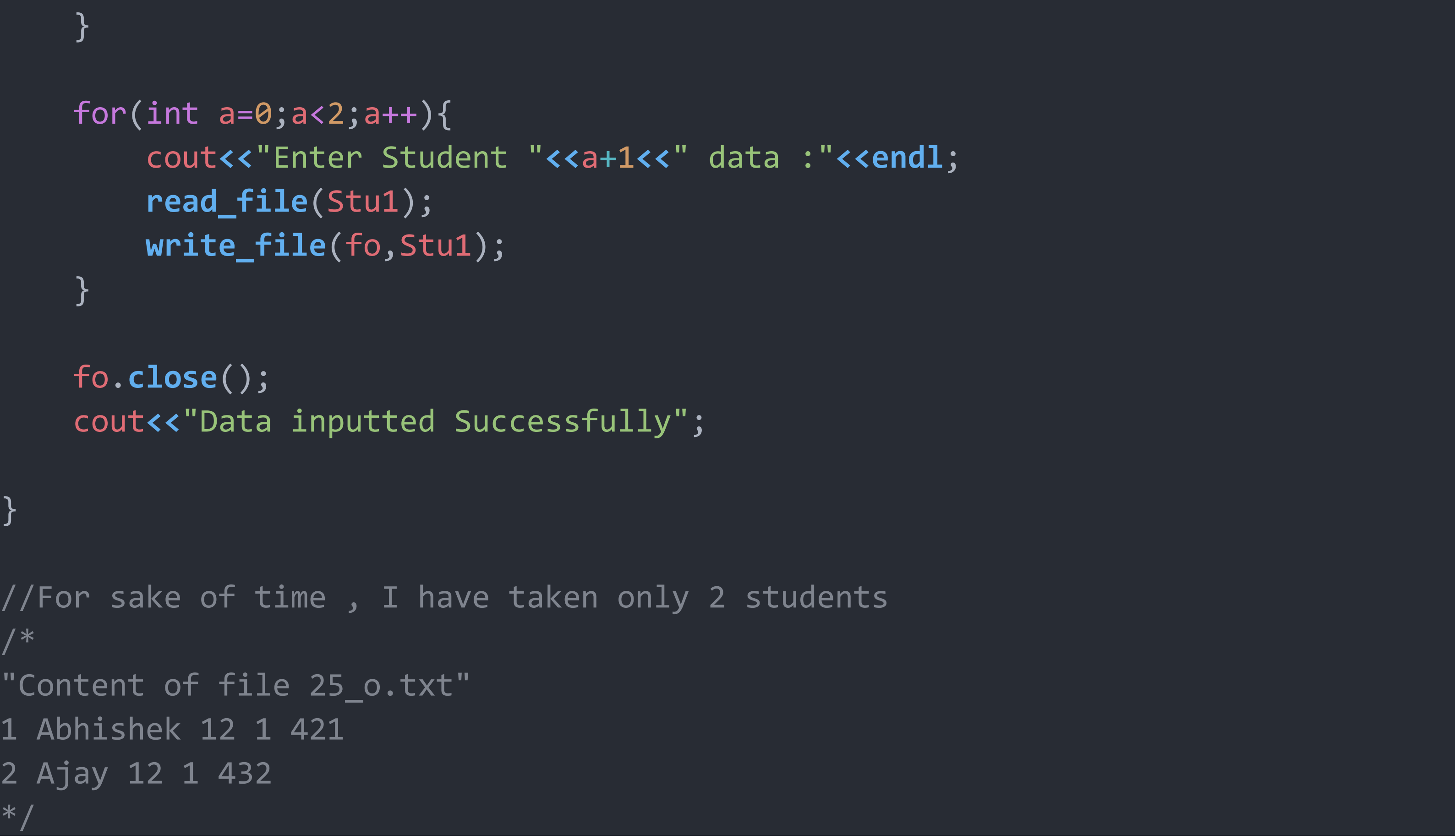
```
myBox++;
cout<<"\nCurrent Values : ";

myBox.print();
--myBox;
myBox--;
Box b2;

cout<<"\nCurrent Values : ";

myBox.print();
cout<<"b2==myBox : "<<((b2==myBox)?"True":"False")<<endl;
myBox.menu();
b2=myBox;
cout<<"After assignment , b2==myBox : "<<((b2==myBox)?"True":"False")<<endl;
cout<<"Second Box : ";
b2.print();

}//main

/* OUPTPUT
"Test Case 1"
Enter <length>,<breadth> and <height> 10 10 10
After Prefix Increment :  Length = 11 , Breadth = 11 , Height = 11
After postfix Increment :  Length = 11 , Breadth = 11 , Height = 11

Current Values :  Length = 12 , Breadth = 12 , Height = 12
After Prefix Decrement :  Length = 11 , Breadth = 11 , Height = 11
After Postfix Decrement :  Length = 11 , Breadth = 11 , Height = 11

Current Values :  Length = 10 , Breadth = 10 , Height = 10
b2==myBox : False

0: Exit
1: for Total Surface Area, 2 : for Volume
3: check_cube_or_cuboid => 1
Total Surface Area is : 600 sq units.
0: Exit
1: for Total Surface Area, 2 : for Volume
3: check_cube_or_cuboid => 2
The volume of given Box is : 1000 cubic units.
0: Exit
1: for Total Surface Area, 2 : for Volume
3: check_cube_or_cuboid => 3
It is a cube
------------------------------------------------------------------
```



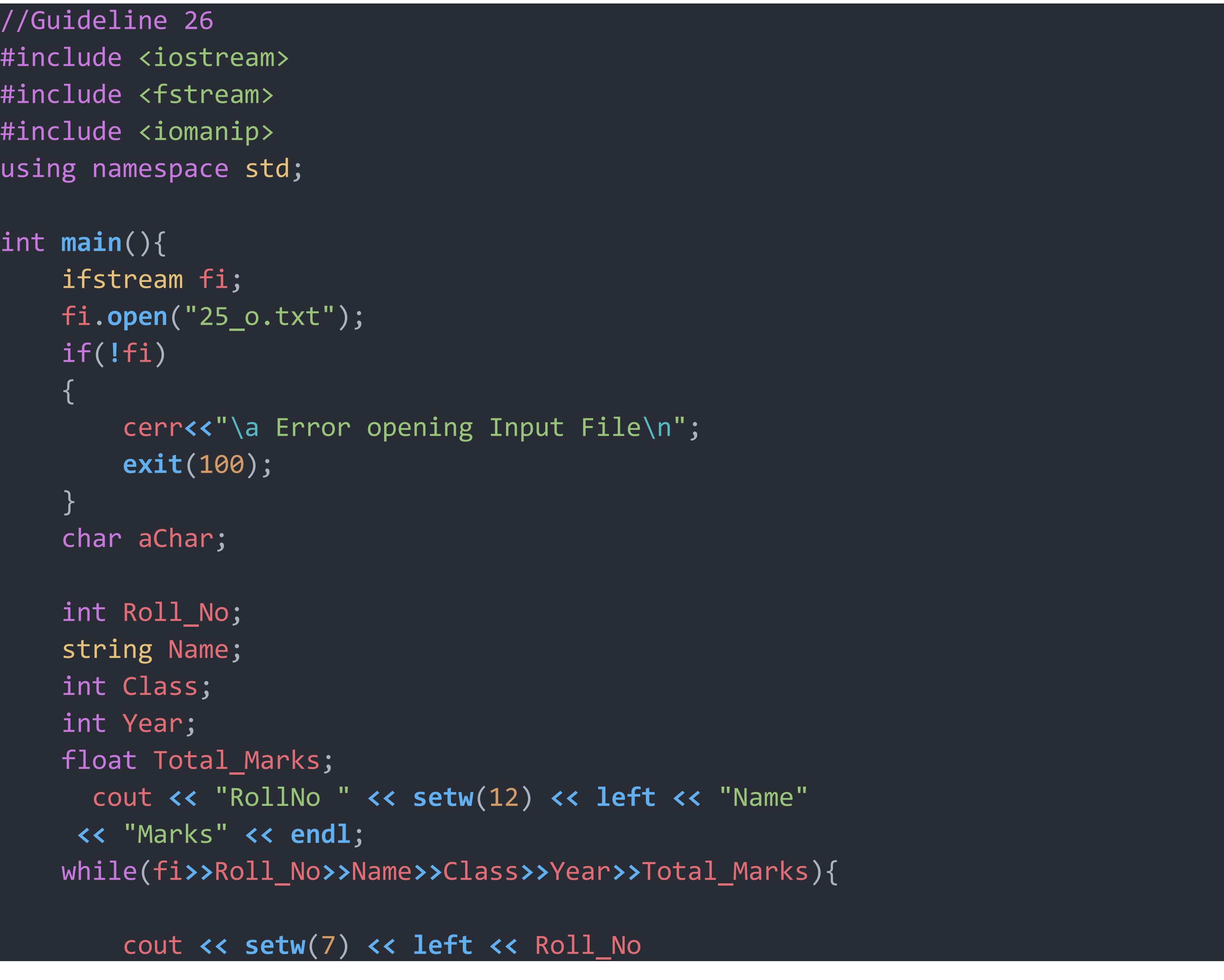
```
"Test Case 2"
Enter <length>,<breadth> and <height> 10 4 3
0: Exit
1: for Total Surface Area, 2 : for Volume
3: check_cube_or_cuboid => 3
It is a cuboid

*/
```



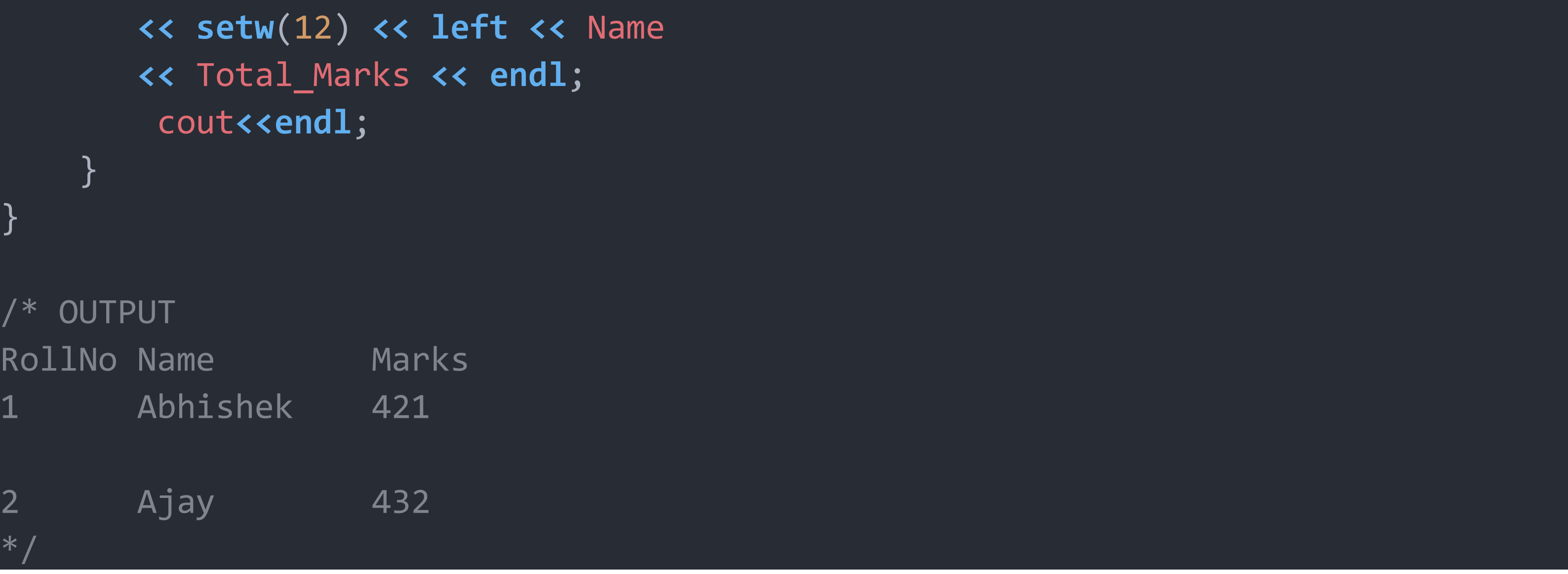
```
//Guideline 25
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;

struct Student
{
    int Roll_No;
    string Name;
    int Class;
    int Year;
    float Total_Marks;
    void print(){
        cout<<Roll_No<<" "<<Name<<Class<<" "<<Year<<" "<<Total_Marks<<endl;
    }
};
void read_file(Student &Stu1){
    cin>>Stu1.Roll_No>>Stu1.Name>>Stu1.Class>>Stu1.Year>>Stu1.Total_Marks;

}
void write_file(ofstream &fo,Student &Stu1){
    fo<<Stu1.Roll_No<<' '<<Stu1.Name<<' '<<Stu1.Class<<' '<<Stu1.Year<<'
'<<Stu1.Total_Marks<<endl;

}
int main(){
    Student Stu1;
    cout<<"Roll_No"<<setw(10)<<"Name"<<setw(8)<<"Class"<<setw(5)<<"Year"<<setw(1
2)<<"Total Marks"<<endl;

    ofstream fo;
    fo.open("25_o.txt");
    if(!fo)
    {
        cerr<<"\a Error opening Output File\n";
        exit(100);
```



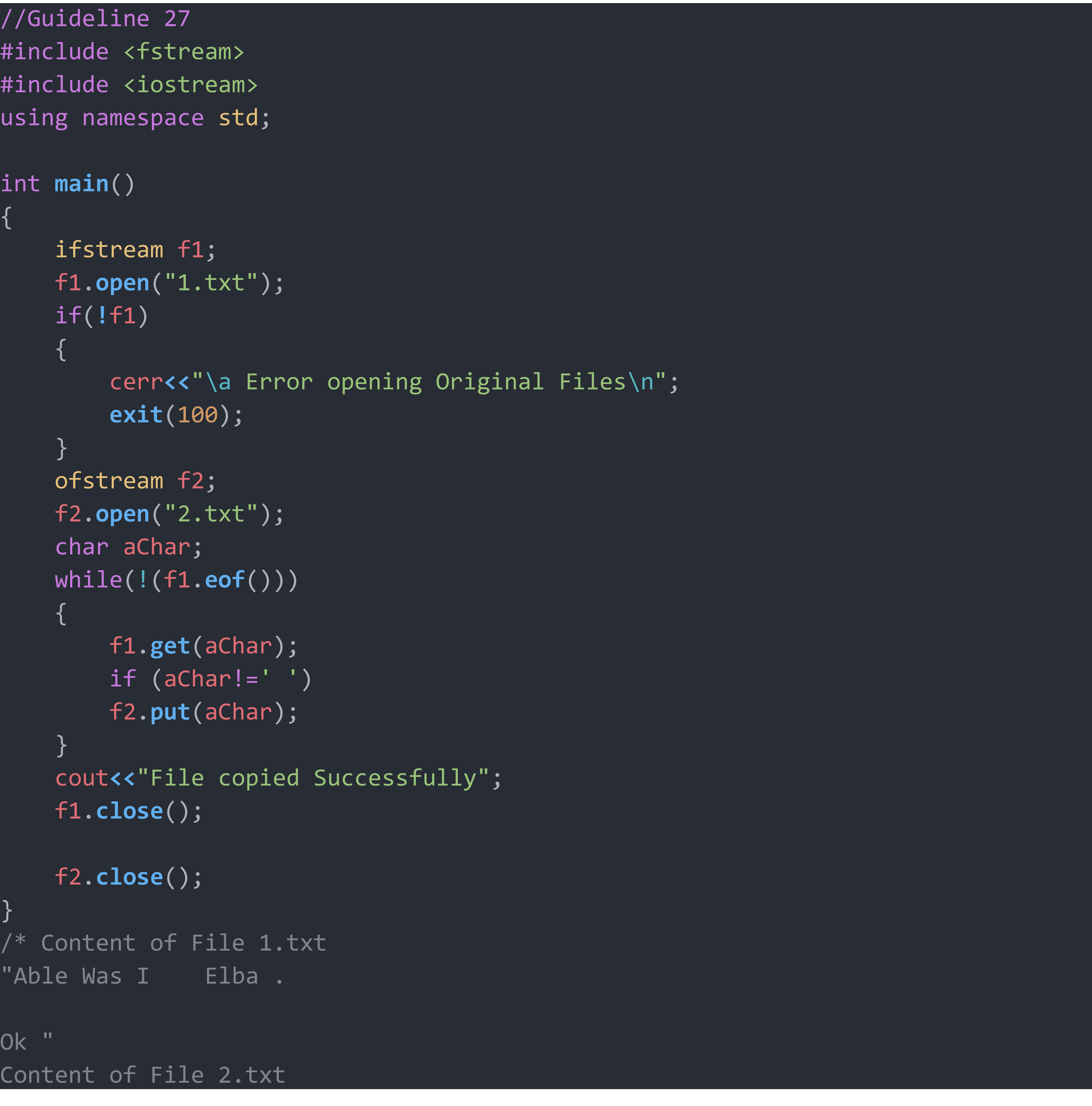
```
    }

    for(int a=0;a<2;a++){
        cout<<"Enter Student "<<a+1<<" data :"<<endl;
        read_file(Stu1);
        write_file(fo,Stu1);
    }

    fo.close();
    cout<<"Data inputted Successfully";

}

//For sake of time , I have taken only 2 students
/*
"Content of file 25_o.txt"
1 Abhishek 12 1 421
2 Ajay 12 1 432
*/
```



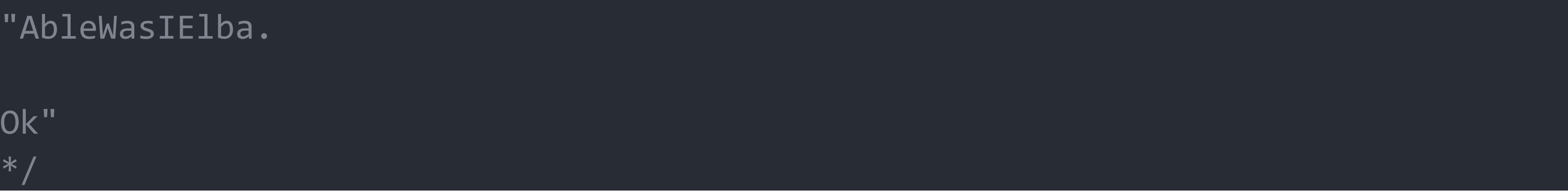
```
//Guideline 26
#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;

int main(){
    ifstream fi;
    fi.open("25_o.txt");
    if(!fi)
    {
        cerr<<"\a Error opening Input File\n";
        exit(100);
    }
    char aChar;

    int Roll_No;
    string Name;
    int Class;
    int Year;
    float Total_Marks;
      cout << "RollNo " << setw(12) << left << "Name"
     << "Marks" << endl;
    while(fi>>Roll_No>>Name>>Class>>Year>>Total_Marks){

        cout << setw(7) << left << Roll_No
```



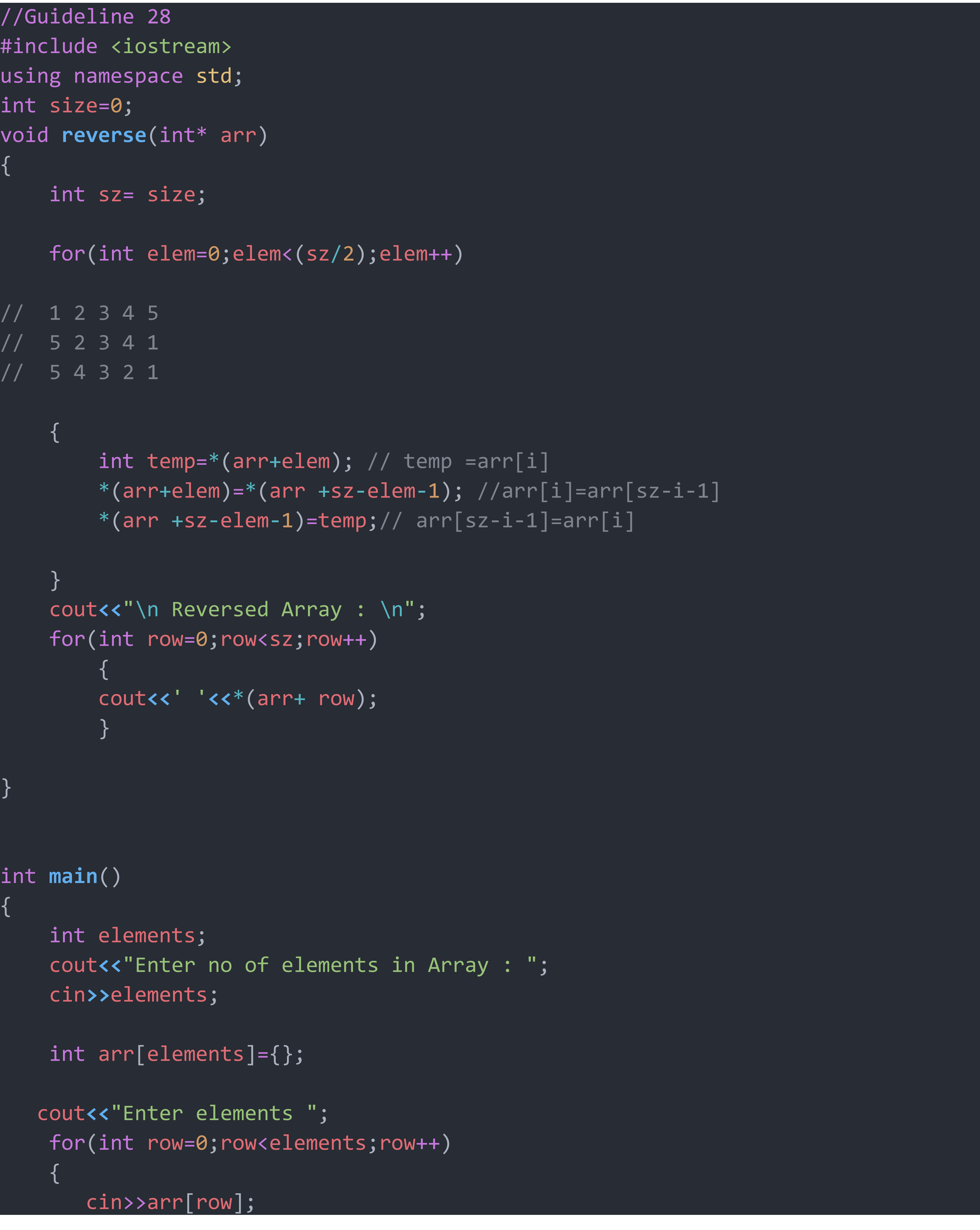
```
       << setw(12) << left << Name
       << Total_Marks << endl;
        cout<<endl;
    }
}

/* OUTPUT
RollNo Name        Marks
1      Abhishek    421

2      Ajay        432
*/
```



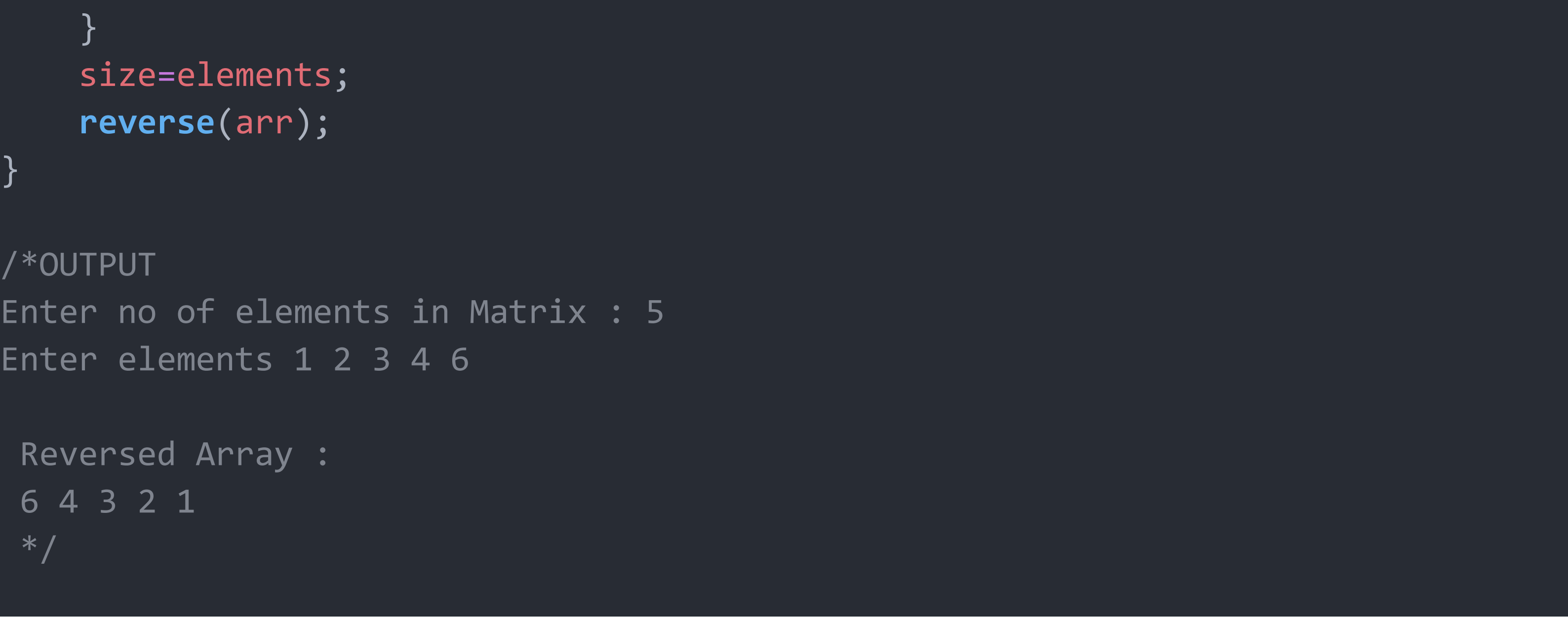
```
//Guideline 27
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
    ifstream f1;
    f1.open("1.txt");
    if(!f1)
    {
        cerr<<"\a Error opening Original Files\n";
        exit(100);
    }
    ofstream f2;
    f2.open("2.txt");
    char aChar;
    while(!(f1.eof()))
    {
        f1.get(aChar);
        if (aChar!=' ')
        f2.put(aChar);
    }
    cout<<"File copied Successfully";
    f1.close();

    f2.close();
}
/* Content of File 1.txt
"Able Was I    Elba .

Ok "
Content of File 2.txt
```



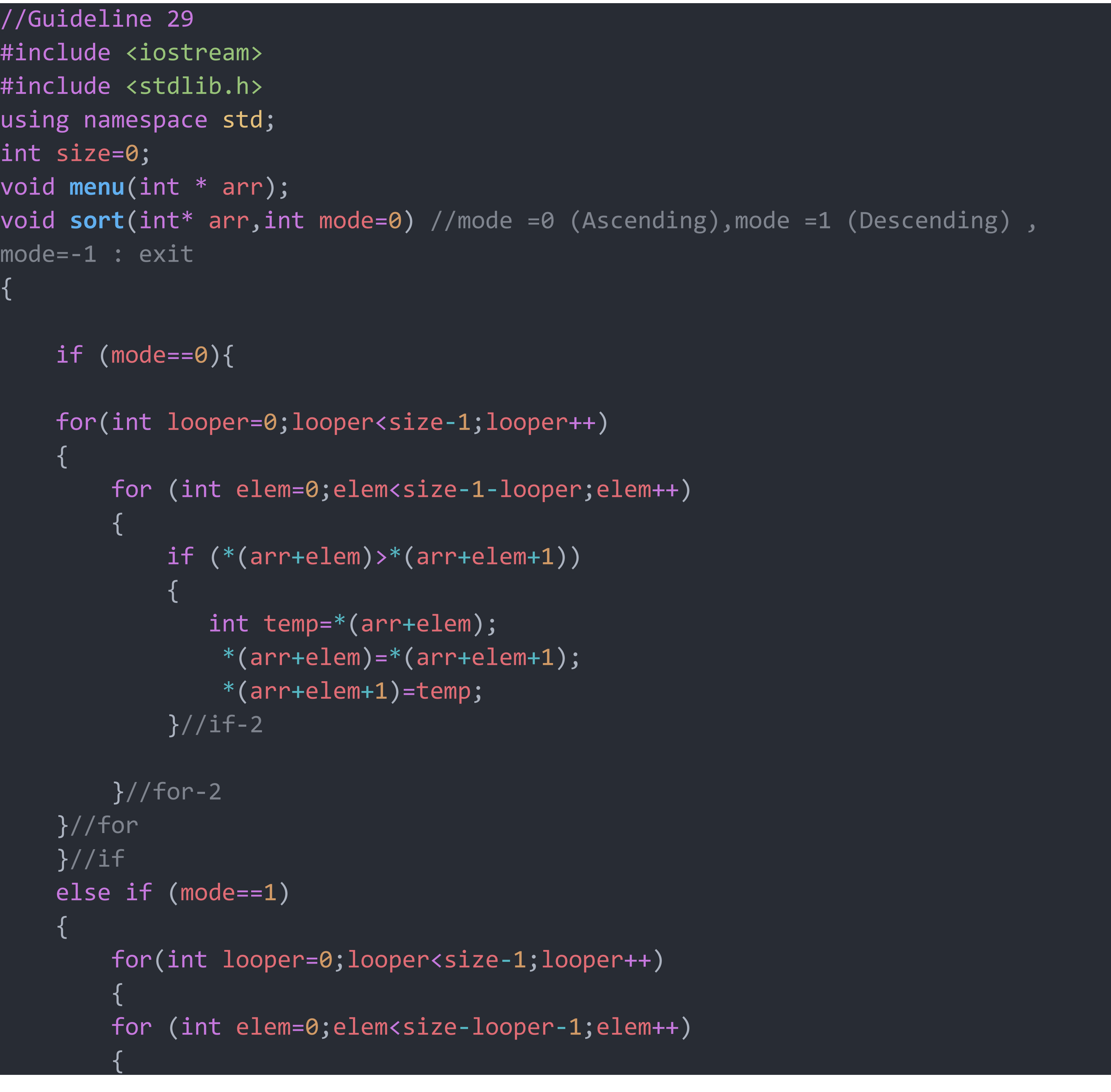
```
"AbleWasIElba.

Ok"
*/
```



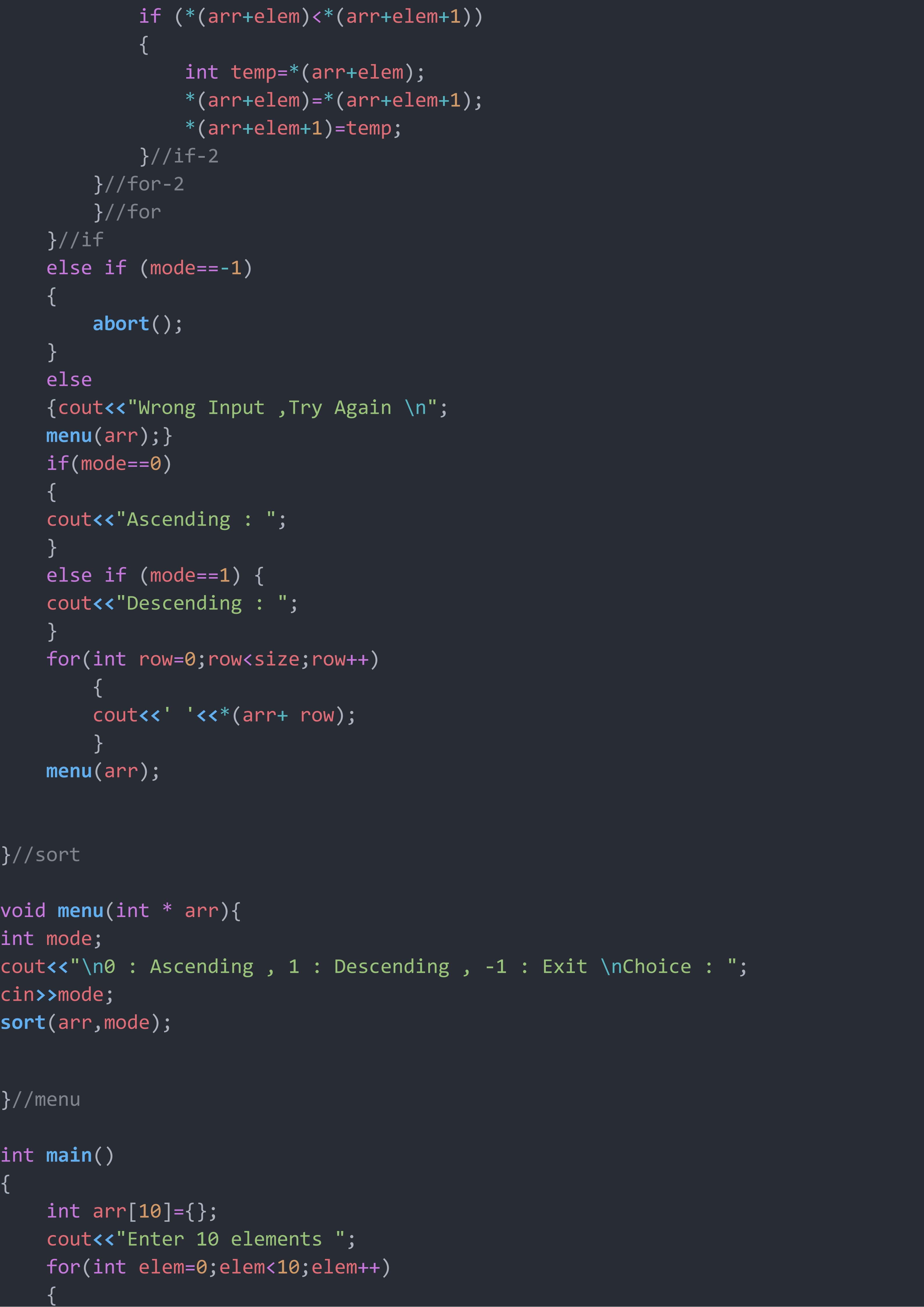
```
//Guideline 28
#include <iostream>
using namespace std;
int size=0;
void reverse(int* arr)
{
    int sz= size;

    for(int elem=0;elem<(sz/2);elem++)

//  1 2 3 4 5
//  5 2 3 4 1
//  5 4 3 2 1

    {
        int temp=*(arr+elem); // temp =arr[i]
        *(arr+elem)=*(arr +sz-elem-1); //arr[i]=arr[sz-i-1]
        *(arr +sz-elem-1)=temp;// arr[sz-i-1]=arr[i]

    }
    cout<<"\n Reversed Array : \n";
    for(int row=0;row<sz;row++)
        {
        cout<<' '<<*(arr+ row);
        }

}


int main()
{
    int elements;
    cout<<"Enter no of elements in Array : ";
    cin>>elements;

    int arr[elements]={};

   cout<<"Enter elements ";
    for(int row=0;row<elements;row++)
    {
       cin>>arr[row];
```



```
    }
    size=elements;
    reverse(arr);
}

/*OUTPUT
Enter no of elements in Matrix : 5
Enter elements 1 2 3 4 6

 Reversed Array :
 6 4 3 2 1
 */
```



```
//Guideline 29
#include <iostream>
#include <stdlib.h>
using namespace std;
int size=0;
void menu(int * arr);
void sort(int* arr,int mode=0) //mode =0 (Ascending),mode =1 (Descending) ,
mode=-1 : exit
{

    if (mode==0){

    for(int looper=0;looper<size-1;looper++)
    {
        for (int elem=0;elem<size-1-looper;elem++)
        {
            if (*(arr+elem)>*(arr+elem+1))
            {
               int temp=*(arr+elem);
                *(arr+elem)=*(arr+elem+1);
                *(arr+elem+1)=temp;
            }//if-2

        }//for-2
    }//for
    }//if
    else if (mode==1)
    {
        for(int looper=0;looper<size-1;looper++)
        {
        for (int elem=0;elem<size-looper-1;elem++)
        {
```



```
            if (*(arr+elem)<*(arr+elem+1))
            {
                int temp=*(arr+elem);
                *(arr+elem)=*(arr+elem+1);
                *(arr+elem+1)=temp;
            }//if-2
        }//for-2
        }//for
    }//if
    else if (mode==-1)
    {
        abort();
    }
    else
    {cout<<"Wrong Input ,Try Again \n";
    menu(arr);}
    if(mode==0)
    {
    cout<<"Ascending : ";
    }
    else if (mode==1) {
    cout<<"Descending : ";
    }
    for(int row=0;row<size;row++)
        {
        cout<<' '<<*(arr+ row);
        }
    menu(arr);


}//sort

void menu(int * arr){
int mode;
cout<<"\n0 : Ascending , 1 : Descending , -1 : Exit \nChoice : ";
cin>>mode;
sort(arr,mode);


}//menu

int main()
{
    int arr[10]={};
    cout<<"Enter 10 elements ";
    for(int elem=0;elem<10;elem++)
    {
        cin>>arr[elem];
    }
    size=sizeof(arr)/sizeof(arr[0]);
    menu(arr);
}//main


/* OUTPUT
Enter 10 elements 3 2 1 0 6 5 4 9 8 7

0 : Ascending , 1 : Descending , -1 : Exit
Choice : 0
Ascending :  0 1 2 3 4 5 6 7 8 9
0 : Ascending , 1 : Descending , -1 : Exit
Choice : 1
Descending :  9 8 7 6 5 4 3 2 1 0
0 : Ascending , 1 : Descending , -1 : Exit
Choice : -1
*/
```