1) Write a C++ program that finds the sum of the diagonal of the matrix

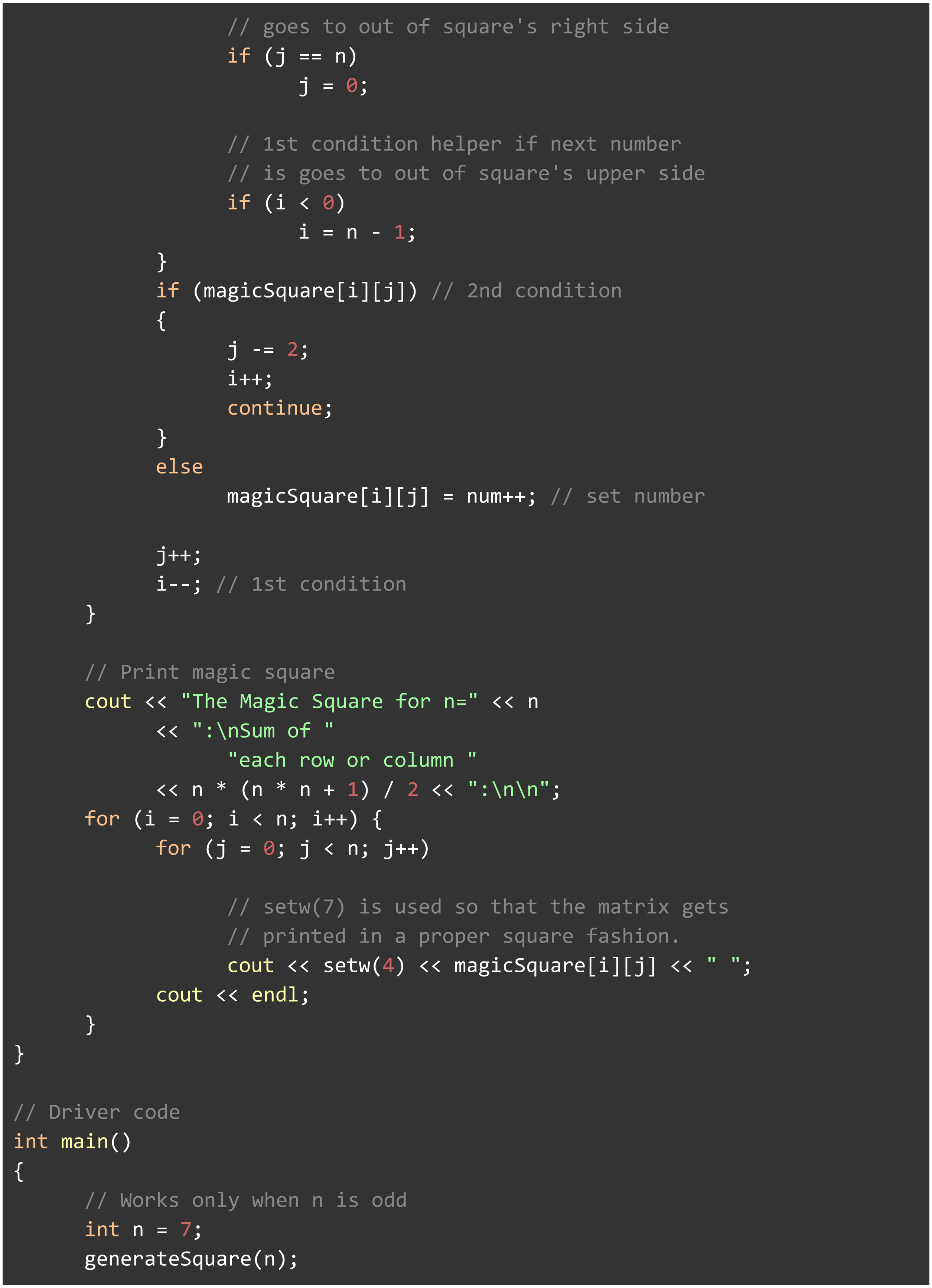
```cpp
#include <iostream>
#define SIZE 5
using namespace std;

int main()
{
  int matrix[SIZE][SIZE];
  int sum_left =0, sum_right = 0;
  cout << "Enter elements into the matrix \n";

  //Taking input into the Matrix and
  //Adding if they are diagonal elements
  for(int i=0; i<SIZE ; i++){
    for(int j=0; j<SIZE; j++){
      cin >> matrix[i][j];
      if(i==j)
        sum_left += matrix[i][j];
      if((i+j) == SIZE-1)
        sum_right += matrix[i][j];
    }
  }

  cout << "Sum of Left Diagonal: "<< sum_left << endl;
  cout << "Sum of Right Diagonal: "<< sum_right << endl;

  return 0;
}
```

2) Write a C++ program that prints the diagonal of a matrix.

```cpp
#include <iostream>
#define SIZE 5
using namespace std;

int main()
{
  int matrix[SIZE][SIZE];

  //Take input into Matrix
  cout << "Enter elements into the matrix: \n";
  for(int i=0; i<SIZE ; i++){
    for(int j=0; j<SIZE; j++){
      cin >> matrix[i][j];
    }
  }

  //Output the diagonal elements
  cout << "Diagonal Elements: \n";
  for(int i=0; i<SIZE; i++){
    for(int j=0; j<SIZE; j++){
      if(i==j || i+j==SIZE-1)
        cout << matrix[i][j] << "\t";
      else
        cout << " " << "\t";
    }
    cout << "\n";
  }

  return 0;
}
```

3) write a program to print a Magic square of odd order.

```
// C++ program to generate odd sized magic squares
#include <bits/stdc++.h>
using namespace std;

// A function to generate odd sized magic squares
void generateSquare(int n)
{
      int magicSquare[n][n];

      // set all slots as 0
      memset(magicSquare, 0, sizeof(magicSquare));

      // Initialize position for 1
      int i = n / 2;
      int j = n - 1;

      // One by one put all values in magic square
      for (int num = 1; num <= n * n;) {
            if (i == -1 && j == n) // 3rd condition
            {
                  j = n - 2;
                  i = 0;
            }
            else {
                  // 1st condition helper if next number
```



```
                // goes to out of square's right side
                if (j == n)
                    j = 0;

                // 1st condition helper if next number
                // is goes to out of square's upper side
                if (i < 0)
                    i = n - 1;
            }
            if (magicSquare[i][j]) // 2nd condition
            {
                j -= 2;
                i++;
                continue;
            }
            else
                magicSquare[i][j] = num++; // set number

            j++;
            i--; // 1st condition
        }

        // Print magic square
        cout << "The Magic Square for n=" << n
            << ":\nSum of "
                "each row or column "
            << n * (n * n + 1) / 2 << ":\n\n";
        for (i = 0; i < n; i++) {
            for (j = 0; j < n; j++)

                // setw(7) is used so that the matrix gets
                // printed in a proper square fashion.
                cout << setw(4) << magicSquare[i][j] << " ";
            cout << endl;
        }
}

// Driver code
int main()
{
        // Works only when n is odd
        int n = 7;
        generateSquare(n);
```

```
      return 0;
}
```

4) Write a program to print a Magic square of doubly even order

```
define an 2-D array of order n*n
    // fill array with their index-counting
    // starting from 1
    for ( i = 0; i<n; i++)
    {
        for ( j = 0; j<n; j++)
            // filling array with its count value
            // starting from 1;
            arr[i][j] = (n*i) + j + 1;
    }

    // change value of Array elements
    // at fix location as per rule
    // (n*n+1)-arr[i][j]
    // Top Left corner of Matrix
   // (order (n/4)*(n/4))
    for ( i = 0; i<n/4; i++)
    {
        for ( j = 0; j<n/4; j++)
            arr[i][j] = (n*n + 1) - arr[i][j];
    }

    // Top Right corner of Matrix
    // (order (n/4)*(n/4))
    for ( i = 0; i< n/4; i++)
    {
        for ( j = 3* (n/4); j<n; j++)
            arr[i][j] = (n*n + 1) - arr[i][j];
    }

    // Bottom Left corner of Matrix
    // (order (n/4)*(n/4))
    for ( i = 3* n/4; i<n; i++)
    {
        for ( j = 0; j<n/4; j++)
```

```
            arr[i][j] = (n*n + 1) - arr[i][j];
    }
    // Bottom Right corner of Matrix
   // (order (n/4)*(n/4))
    for ( i = 3* n/4; i<n; i++)
    {
        for ( j = 3* n/4; j<n; j++)
            arr[i][j] = (n*n + 1) - arr[i][j];
    }
    // Centre of Matrix (order (n/2)*(n/2))
    for ( i = n/4; i<3* n/4; i++)
    {
        for ( j = n/4; j<3* n/4; j++)
            arr[i][j] = (n*n + 1) - arr[i][j];
    }
```

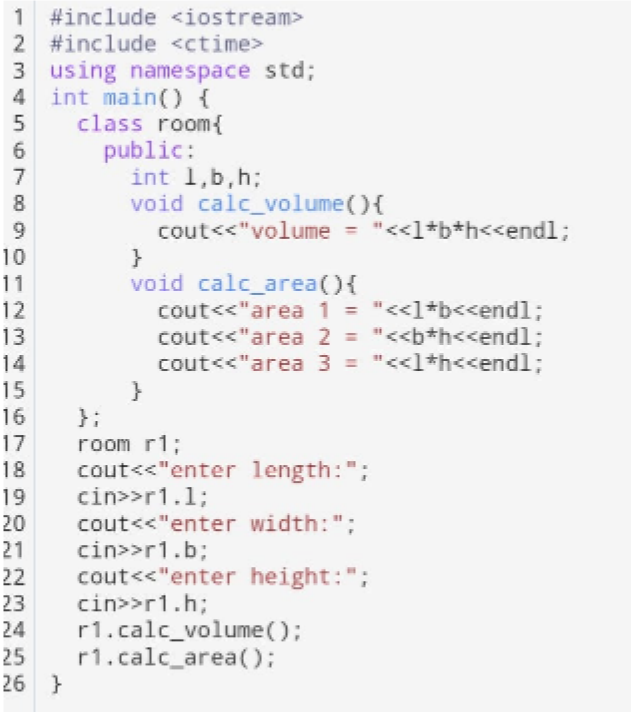
5) Write a program to check whether a given matrix is magic square or not?

```
// C++ program to check whether a given
// matrix is magic matrix or not
#include <bits/stdc++.h>

# define my_sizeof(type) ((char *)(&type+1)-(char*)(&type))
using namespace std;

// Returns true if mat[][] is magic
// square, else returns false.
bool isMagicSquare(int mat[][3])
{
      int n = my_sizeof(mat)/my_sizeof(mat[0]);
      // calculate the sum of
      // the prime diagonal
      int i=0,j=0;
      // sumd1 and sumd2 are the sum of the two diagonals
      int sumd1 = 0, sumd2=0;
      for (i = 0; i < n; i++)
      {
            // (i, i) is the diagonal from top-left -> bottom-right
            // (i, n - i - 1) is the diagonal from top-right -> bottom-left
            sumd1 += mat[i][i];
            sumd2 += mat[i][n-1-i];
      }
      // if the two diagonal sums are unequal then it is not a magic square
      if(sumd1!=sumd2)
            return false;

      // For sums of Rows
      for (i = 0; i < n; i++) {

            int rowSum = 0, colSum = 0;
            for (j = 0; j < n; j++)
            {
                  rowSum += mat[i][j];
                  colSum += mat[j][i];
            }
            if (rowSum != colSum || colSum != sumd1)
                  return false;
      }
return true;
}

// driver program to
// test above function
int main()
{
      int mat[3][3] = {{ 2, 7, 6 },
                        { 9, 5, 1 },
                        { 4, 3, 8 }};

      if (isMagicSquare(mat))
            cout << "Magic Square";
      else
            cout << "Not a magic Square";

      return 0;
}
```

6) Use the built in functions on string (append,strcat,push_back,pop_back,length)by writing a program

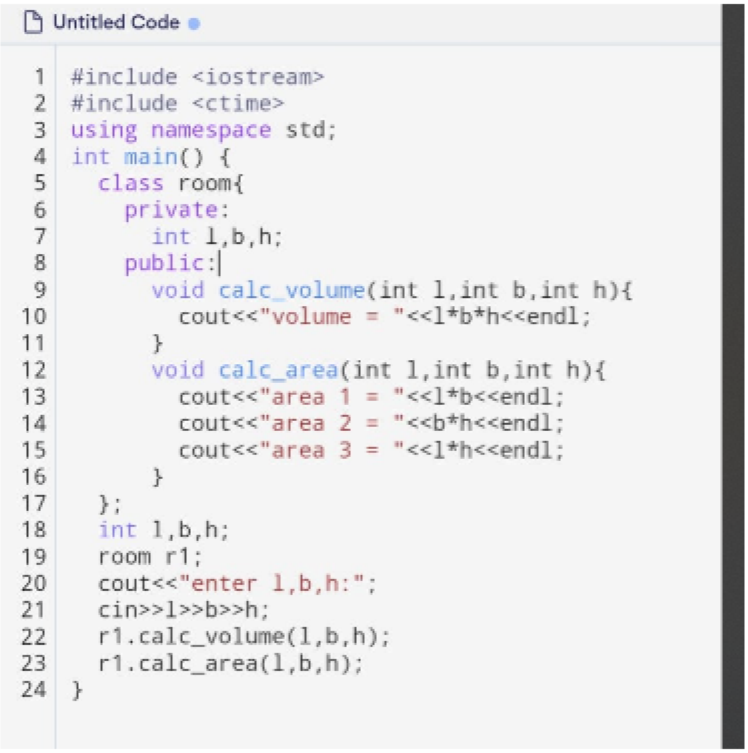
LAB 14

1. Write a Program to illustrate the working of objects and class with name ‘Room’ by specifying public member variables as ‘length’, ‘breadth’ and ‘height’ and public member functions as ‘calculate_area’ and’calculate_volume’. And display the output.



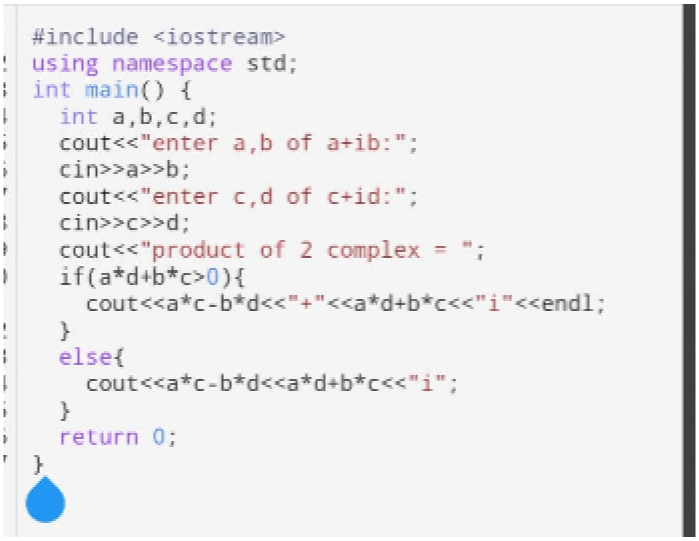
```
#include <iostream>
#include <ctime>
using namespace std;
int main() {
  class room{
    public:
      int l,b,h;
      void calc_volume(){
        cout<<"volume = "<<l*b*h<<endl;
      }
      void calc_area(){
        cout<<"area 1 = "<<l*b<<endl;
        cout<<"area 2 = "<<b*h<<endl;
        cout<<"area 3 = "<<l*h<<endl;
      }
  };
  room r1;
  cout<<"enter length:";
  cin>>r1.l;
  cout<<"enter width:";
  cin>>r1.b;
  cout<<"enter height:";
  cin>>r1.h;
  r1.calc_volume();
  r1.calc_area();
}
```

2. Write a Program to illustrate the working of objects and class with name ‘Room’ by specifying private member variables as ‘length’, ‘breadth’ and ‘height’ and public member functions as ‘calculate_area’ and’calculate_volume’. And display the output.



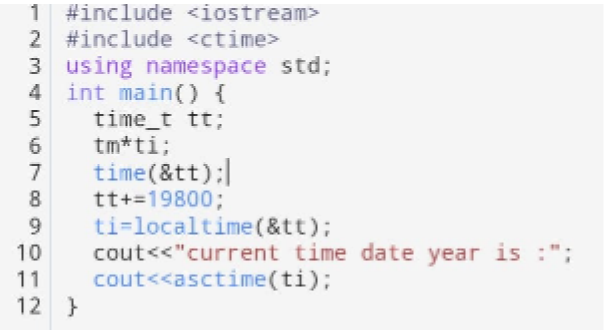
```
#include <iostream>
#include <ctime>
using namespace std;
int main() {
  class room{
    private:
      int l,b,h;
    public:
      void calc_volume(int l,int b,int h){
        cout<<"volume = "<<l*b*h<<endl;
      }
      void calc_area(int l,int b,int h){
        cout<<"area 1 = "<<l*b<<endl;
        cout<<"area 2 = "<<b*h<<endl;
        cout<<"area 3 = "<<l*h<<endl;
      }
  };
  int l,b,h;
  room r1;
  cout<<"enter l,b,h:";
  cin>>l>>b>>h;
  r1.calc_volume(l,b,h);
  r1.calc_area(l,b,h);
}
```

3. Write a program to multiply two complex numbers.



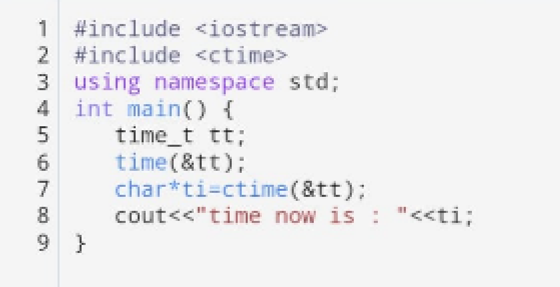
```cpp
#include <iostream>
using namespace std;
int main() {
  int a,b,c,d;
  cout<<"enter a,b of a+ib:";
  cin>>a>>b;
  cout<<"enter c,d of c+id:";
  cin>>c>>d;
  cout<<"product of 2 complex = ";
  if(a*d+b*c>0){
    cout<<a*c-b*d<<"+"<<a*d+b*c<<"i"<<endl;
  }
  else{
    cout<<a*c-b*d<<a*d+b*c<<"i";
  }
  return 0;
}
```

4. Write a program to display the current time by using localtime(), asctime().



```cpp
#include <iostream>
#include <ctime>
using namespace std;
int main() {
  time_t tt;
  tm*ti;
  time(&tt);
  tt+=19800;
  ti=localtime(&tt);
  cout<<"current time date year is :";
  cout<<asctime(ti);
}
```

5. Write a program to display the current time by using ctime().



```cpp
#include <iostream>
#include <ctime>
using namespace std;
int main() {
   time_t tt;
   time(&tt);
   char*ti=ctime(&tt);
   cout<<"time now is : "<<ti;
}
```

6. Write a program to display the current time in terms of structure variables ‘tm’.

7. Write a program to display the difference in time between the given dates.

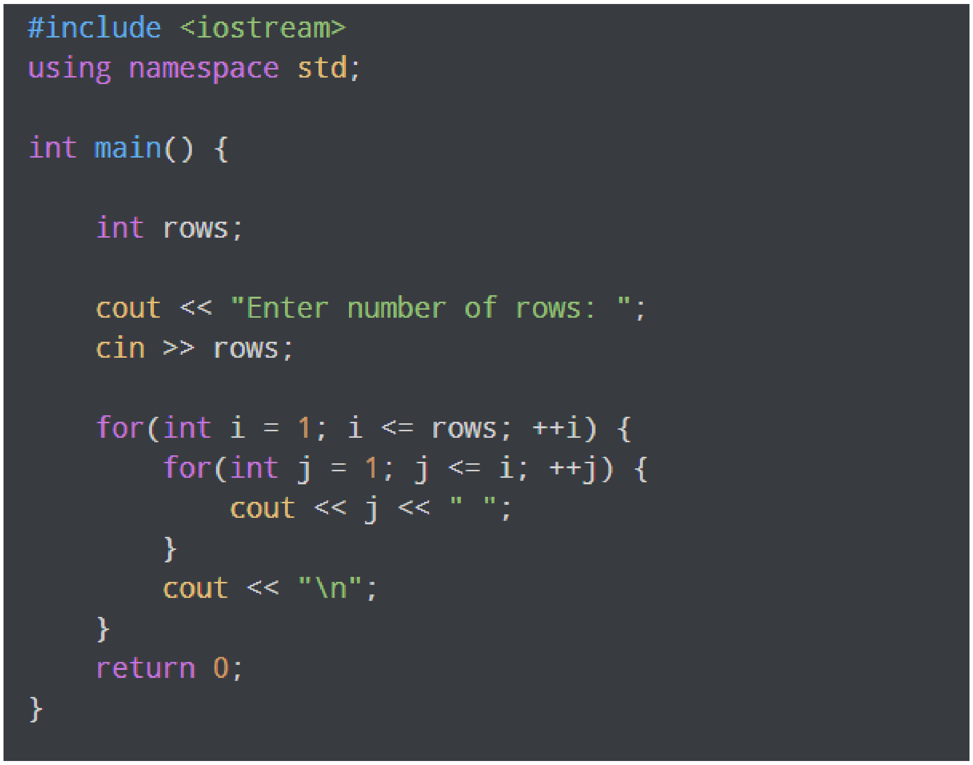
LAB

1. Write a Program To Accept Student Roll No, Marks in 3 Subjects and Calculate Total, Average and Print it.

```
 # include
#include
# include
using namespace std;
int main()
{
int r,b,c,d, tot, avg;
cout<<"ENTER STUDENT ROLL NO ; "<>r;
cout<<"ENTER FIRST SUBJECT MARKS ;"<>b;
cout<<"ENTER SECOND SUBJECT MARKS;"<>c;
cout<<"ENTER THIRD SUBJECT MARKS ;"<>d;
tot=b+c+d;
avg=tot/3;
cout<<"\n\n\t\t Lovely Professional University \n\n";

cout<<"\t STUDENT RNO :"<
```

2. Write a Program to print numeric pyramid

```
1
1 2
1 2 3
1 2 3 4 5
```

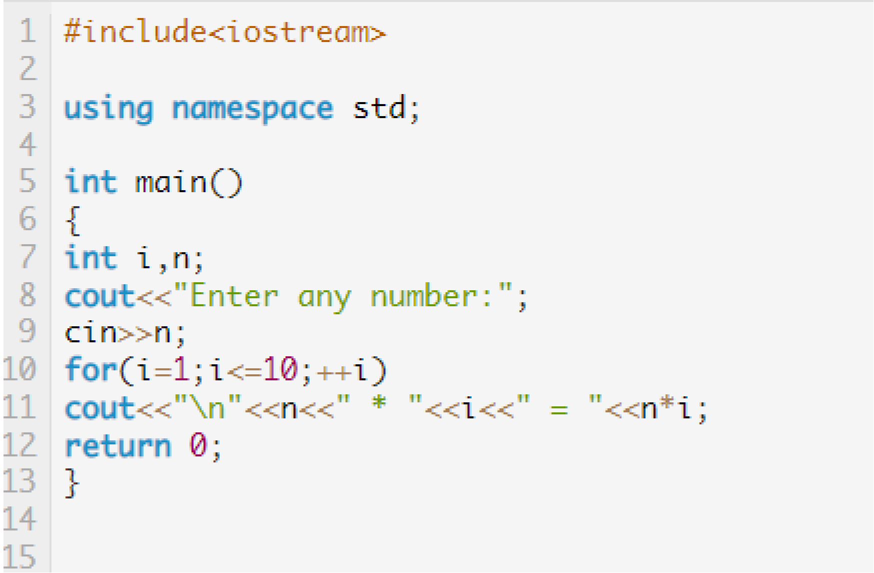


```
#include <iostream>
using namespace std;

int main() {

    int rows;

    cout << "Enter number of rows: ";
    cin >> rows;

    for(int i = 1; i <= rows; ++i) {
        for(int j = 1; j <= i; ++j) {
            cout << j << " ";
        }
        cout << "\n";
    }
    return 0;
}
```

3. Write a Program to print ODD numbers from 1 to 10.

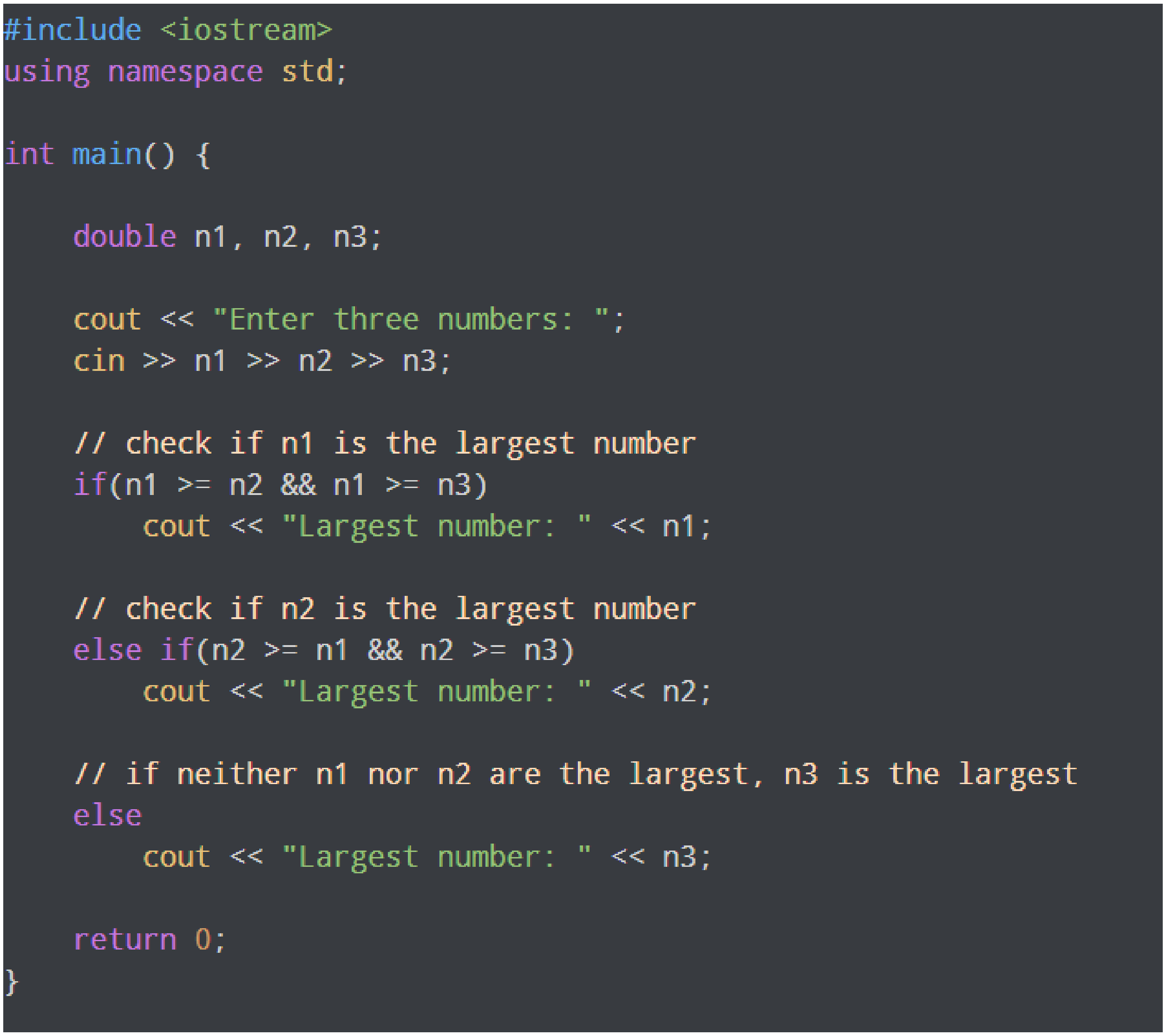
```
 #include<iostream>
using namespace std;

int main()
{
	int number;

	cout << "\nPlease Enter Maximum limit Value to print Odd Numbers =  ";
	cin >> number;

	cout << "\nList of Odd Numbers from 1 to " << number << " are\n";
	for(int i = 1; i <= number; i++)
	{
		if ( i % 2 != 0 )
		{
			cout << i <<" ";
		}
	}

	return 0;
}
```

4. Write a Program to print table of any number.



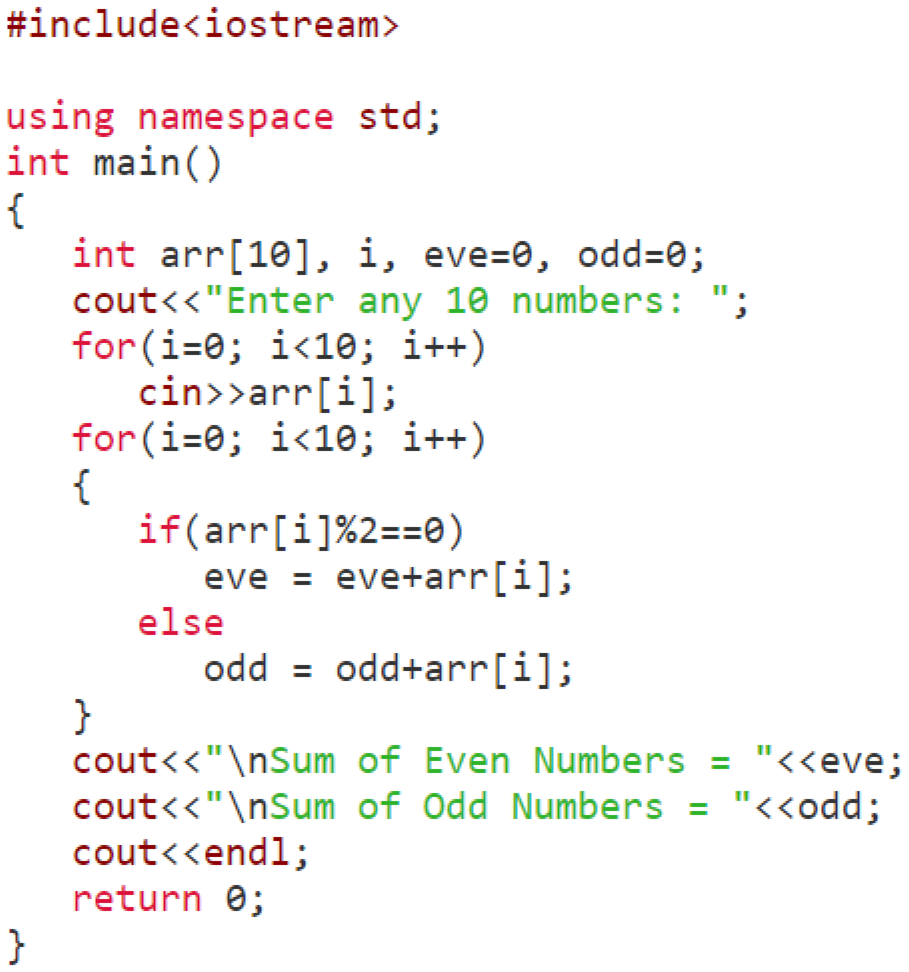
```
#include<iostream>

using namespace std;

int main()
{
int i,n;
cout<<"Enter any number:";
cin>>n;
for(i=1;i<=10;++i)
cout<<"\n"<<n<<" * "<<i<<" = "<<n*i;
return 0;
}


```

5. Write a program to find largest number of a list of numbers entered through keyboard.



```
#include <iostream>
using namespace std;

int main() {

    double n1, n2, n3;

    cout << "Enter three numbers: ";
    cin >> n1 >> n2 >> n3;

    // check if n1 is the largest number
    if(n1 >= n2 && n1 >= n3)
        cout << "Largest number: " << n1;

    // check if n2 is the largest number
    else if(n2 >= n1 && n2 >= n3)
        cout << "Largest number: " << n2;

    // if neither n1 nor n2 are the largest, n3 is the largest
    else
        cout << "Largest number: " << n3;

    return 0;
}
```

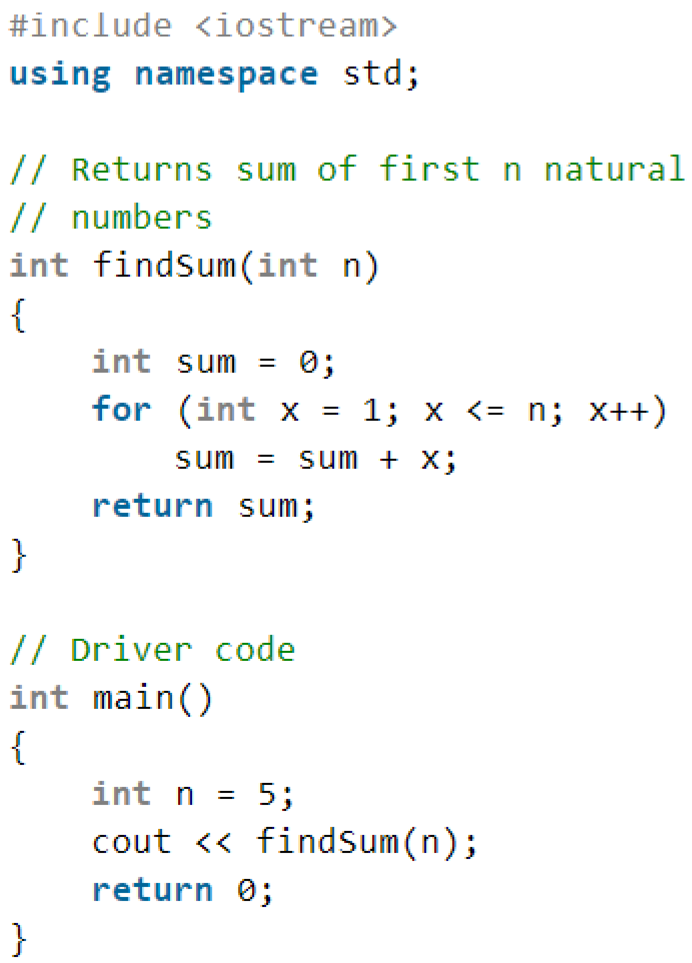
6. Write a Program to calculate and print the sum of even and odd integers of the first n natural numbers.



```
#include<iostream>

using namespace std;
int main()
{
   int arr[10], i, eve=0, odd=0;
   cout<<"Enter any 10 numbers: ";
   for(i=0; i<10; i++)
      cin>>arr[i];
   for(i=0; i<10; i++)
   {
      if(arr[i]%2==0)
         eve = eve+arr[i];
      else
         odd = odd+arr[i];
   }
   cout<<"\nSum of Even Numbers = "<<eve;
   cout<<"\nSum of Odd Numbers = "<<odd;
   cout<<endl;
   return 0;
}
```

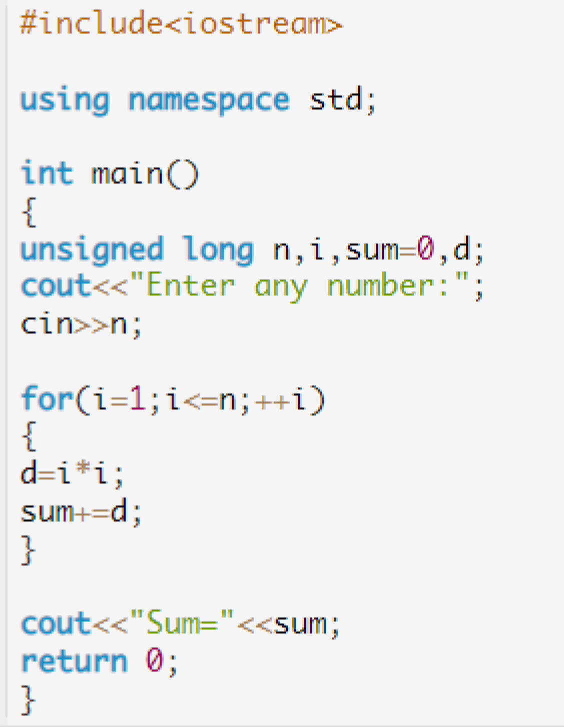
---

7. Write a program to find sum of first N natural number.



```
#include <iostream>
using namespace std;

// Returns sum of first n natural
// numbers
int findSum(int n)
{
    int sum = 0;
    for (int x = 1; x <= n; x++)
        sum = sum + x;
    return sum;
}

// Driver code
int main()
{
    int n = 5;
    cout << findSum(n);
    return 0;
}
```

8. Write a program to print sum of the squares of first N natural number.

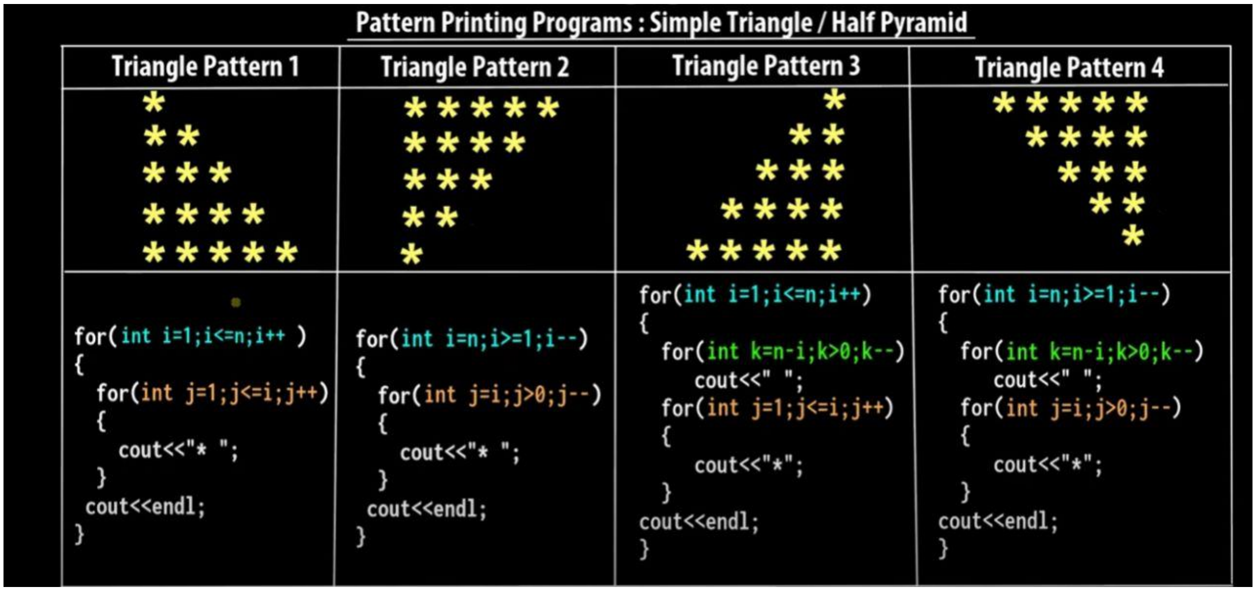


```cpp
#include<iostream>

using namespace std;

int main()
{
unsigned long n,i,sum=0,d;
cout<<"Enter any number:";
cin>>n;

for(i=1;i<=n;++i)
{
d=i*i;
sum+=d;
}

cout<<"Sum="<<sum;
return 0;
}
```

9. Write a program to print sum of the cubes of first N natural number.
Same like above just d=I*I*i

10 Write a Program to print the following pattern

```
****
***
**
*
```



Pattern Printing Programs : Simple Triangle / Half Pyramid

Triangle Pattern 1

```
*
* *
* * *
* * * *
* * * * *

for(int i=1;i<=n;i++ )
{
  for(int j=1;j<=i;j++)
  {
    cout<<"* ";
  }
 cout<<endl;
}
```

Triangle Pattern 2

```
* * * * *
* * * *
* * *
* *
*

for(int i=n;i>=1;i--)
{
  for(int j=i;j>0;j--)
  {
    cout<<"* ";
  }
 cout<<endl;
}
```

Triangle Pattern 3

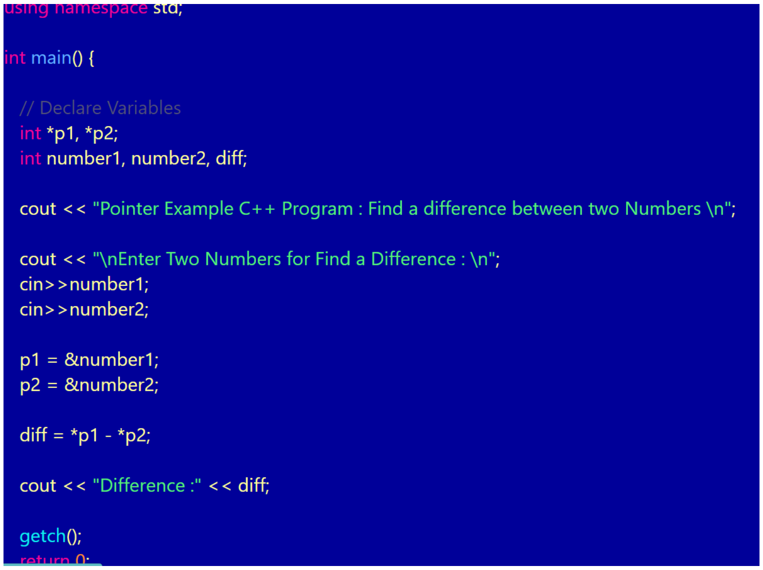
```
        *
      * *
    * * *
  * * * *
* * * * *

for(int i=1;i<=n;i++)
{
  for(int k=n-i;k>0;k--)
     cout<<" ";
  for(int j=1;j<=i;j++)
  {
     cout<<"*";
  }
cout<<endl;
}
```

Triangle Pattern 4

```
* * * * *
  * * * *
    * * *
      * *
        *

for(int i=n;i>=1;i--)
{
  for(int k=n-i;k>0;k--)
     cout<<" ";
  for(int j=i;j>0;j--)
  {
     cout<<"*";
  }
cout<<endl;
}
```

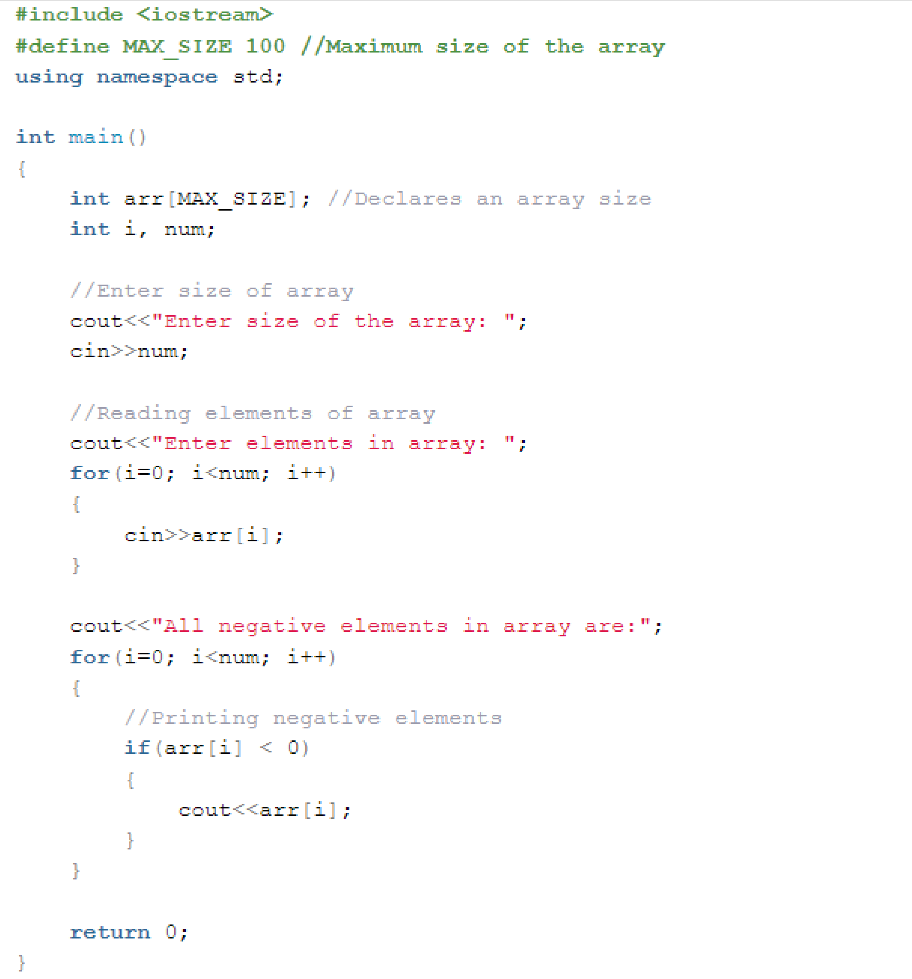
LAB

1. Write a Program to find a difference between two Numbers Using a Pointer in C++.



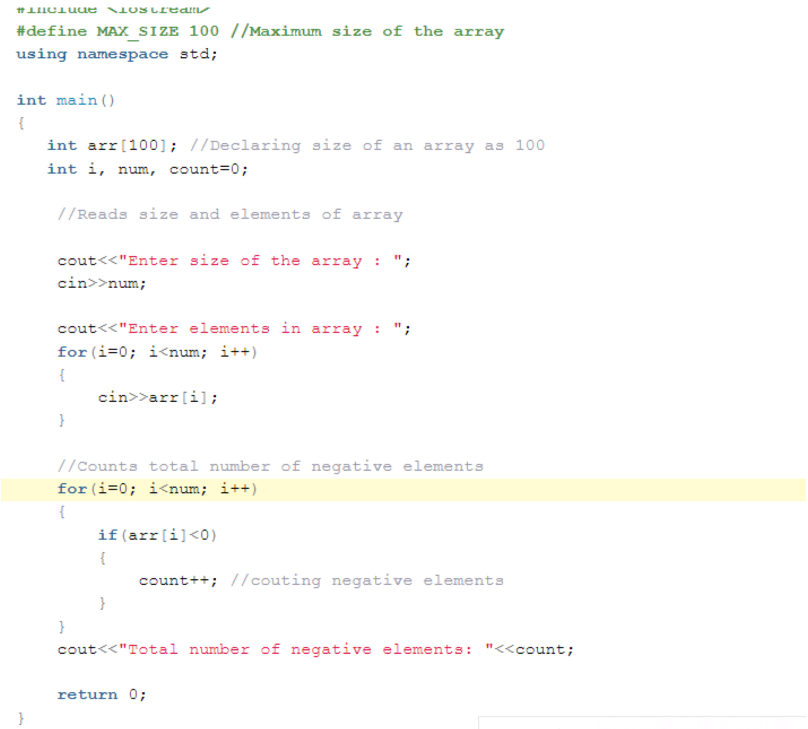
```
using namespace std;

int main() {

  // Declare Variables
  int *p1, *p2;
  int number1, number2, diff;

  cout << "Pointer Example C++ Program : Find a difference between two Numbers \n";

  cout << "\nEnter Two Numbers for Find a Difference : \n";
  cin>>number1;
  cin>>number2;

  p1 = &number1;
  p2 = &number2;

  diff = *p1 - *p2;

  cout << "Difference :" << diff;

  getch();
  return 0;
```

2. Write a program to print all negative elements in an array.



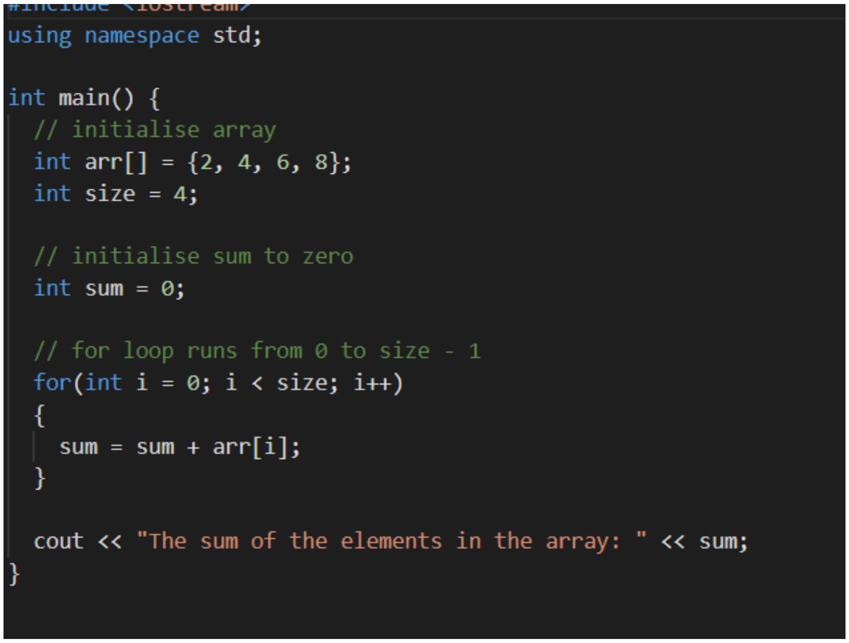
```
#include <iostream>
#define MAX_SIZE 100 //Maximum size of the array
using namespace std;

int main()
{
    int arr[MAX_SIZE]; //Declares an array size
    int i, num;

    //Enter size of array
    cout<<"Enter size of the array: ";
    cin>>num;

    //Reading elements of array
    cout<<"Enter elements in array: ";
    for(i=0; i<num; i++)
    {
        cin>>arr[i];
    }

    cout<<"All negative elements in array are:";
    for(i=0; i<num; i++)
    {
        //Printing negative elements
        if(arr[i] < 0)
        {
            cout<<arr[i];
        }
    }

    return 0;
}
```

3. Write a program to count the total number of negative elements in an array.



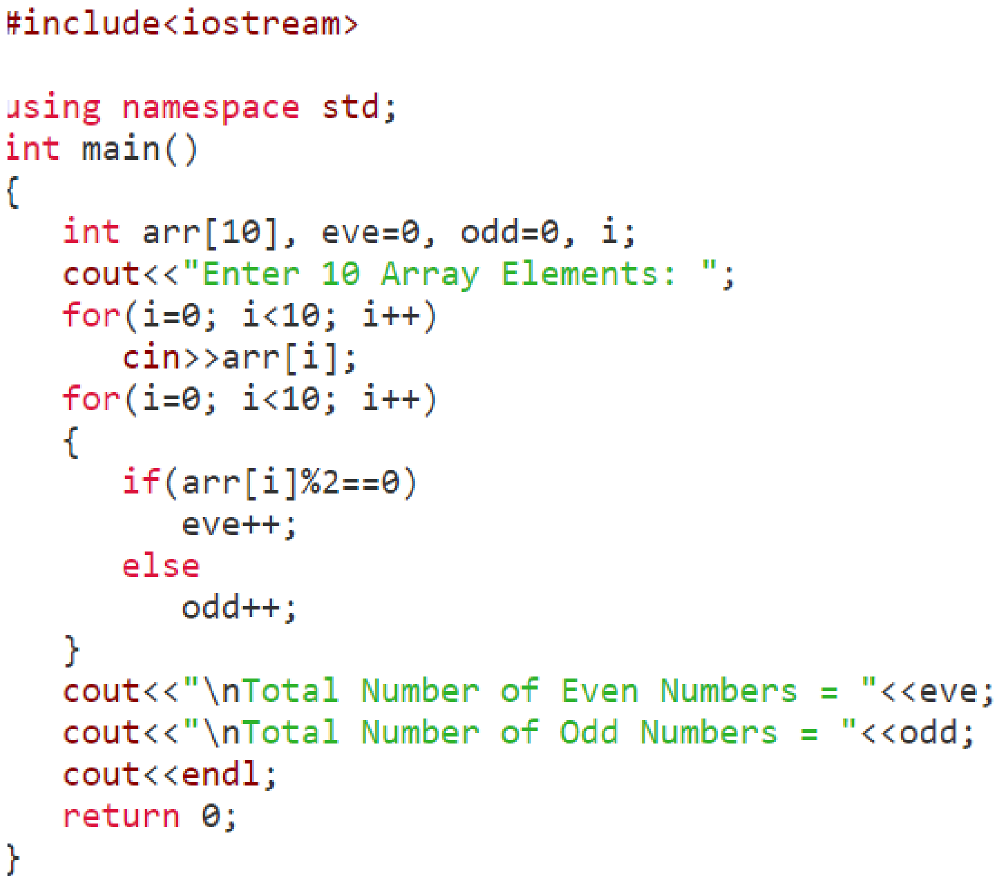
```
#include <iostream>
#define MAX_SIZE 100 //Maximum size of the array
using namespace std;

int main()
{
   int arr[100]; //Declaring size of an array as 100
   int i, num, count=0;

    //Reads size and elements of array

    cout<<"Enter size of the array : ";
    cin>>num;

    cout<<"Enter elements in array : ";
    for(i=0; i<num; i++)
    {
        cin>>arr[i];
    }

    //Counts total number of negative elements
    for(i=0; i<num; i++)
    {
        if(arr[i]<0)
        {
            count++; //couting negative elements
        }
    }
    cout<<"Total number of negative elements: "<<count;

    return 0;
}
```

4. Write a program to find the sum of all elements of an array.



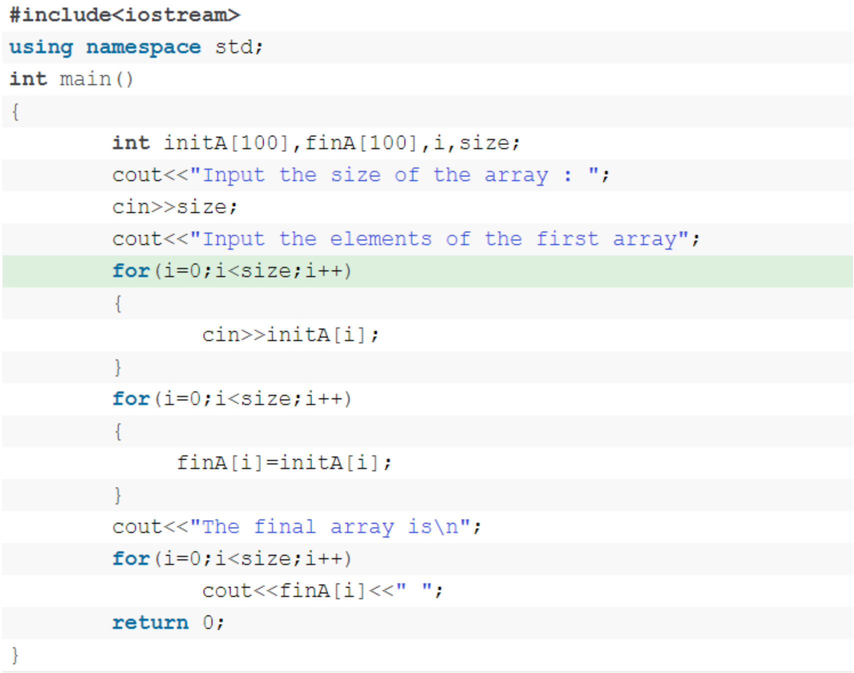
```
#include <iostream>
using namespace std;

int main() {
  // initialise array
  int arr[] = {2, 4, 6, 8};
  int size = 4;

  // initialise sum to zero
  int sum = 0;

  // for loop runs from 0 to size - 1
  for(int i = 0; i < size; i++)
  {
    sum = sum + arr[i];
  }

  cout << "The sum of the elements in the array: " << sum;
}
```

5. Write a program to count even and odd elements in an array.



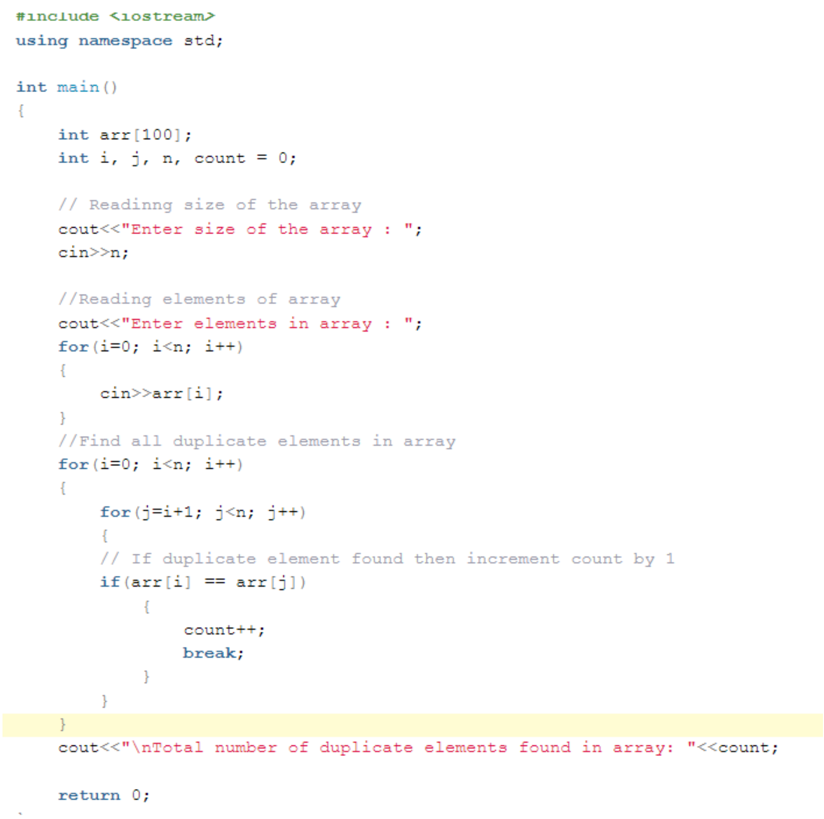
```
#include<iostream>

using namespace std;
int main()
{
   int arr[10], eve=0, odd=0, i;
   cout<<"Enter 10 Array Elements: ";
   for(i=0; i<10; i++)
      cin>>arr[i];
   for(i=0; i<10; i++)
   {
      if(arr[i]%2==0)
         eve++;
      else
         odd++;
   }
   cout<<"\nTotal Number of Even Numbers = "<<eve;
   cout<<"\nTotal Number of Odd Numbers = "<<odd;
   cout<<endl;
   return 0;
}
```

6. Write a program to copy all elements of one array to another.



```
#include<iostream>
using namespace std;
int main()
{
        int initA[100],finA[100],i,size;
        cout<<"Input the size of the array : ";
        cin>>size;
        cout<<"Input the elements of the first array";
        for(i=0;i<size;i++)
        {
               cin>>initA[i];
        }
        for(i=0;i<size;i++)
        {
             finA[i]=initA[i];
        }
        cout<<"The final array is\n";
        for(i=0;i<size;i++)
               cout<<finA[i]<<" ";
        return 0;
}
```

7. Write a program to count total duplicate elements in an array.

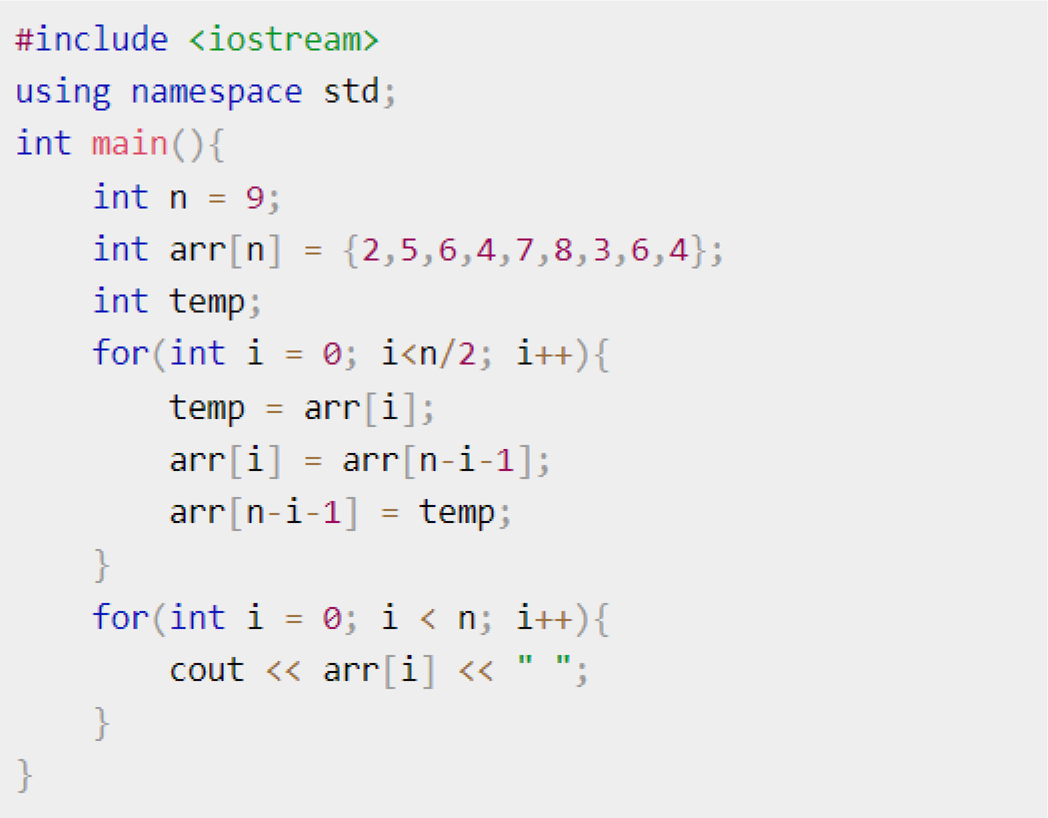


```
#include <iostream>
using namespace std;

int main()
{
    int arr[100];
    int i, j, n, count = 0;

    // Readinng size of the array
    cout<<"Enter size of the array : ";
    cin>>n;

    //Reading elements of array
    cout<<"Enter elements in array : ";
    for(i=0; i<n; i++)
    {
        cin>>arr[i];
    }
    //Find all duplicate elements in array
    for(i=0; i<n; i++)
    {
        for(j=i+1; j<n; j++)
        {
        // If duplicate element found then increment count by 1
        if(arr[i] == arr[j])
            {
                count++;
                break;
            }
        }
    }
    cout<<"\nTotal number of duplicate elements found in array: "<<count;

    return 0;
```

8. Write C++ program to put even and odd elements of an array in two separate arrays.

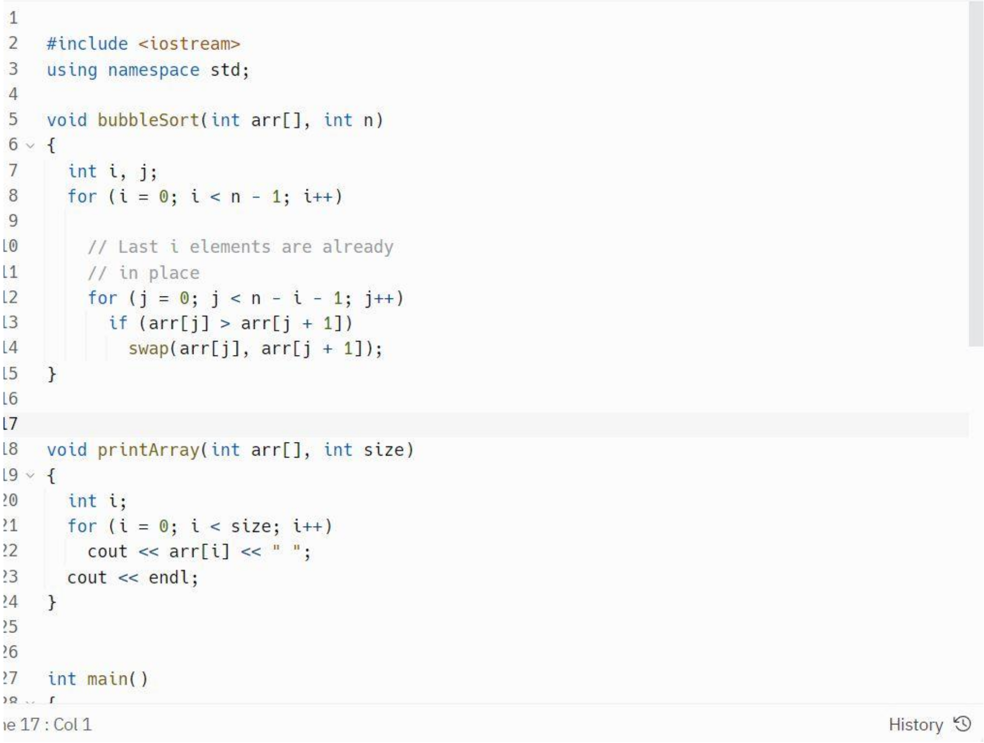
```
#include <iostream>
using namespace std;
int main()
{
    int arr[10],even[10],odd[10],evncnt=0,oddcnt=0,i;
    cout<<"Input numbers in the array";
    for(i=0;i<10;i++)
        cin>>arr[i];
    for(i=0;i<10;i++)
    {
            if(arr[i]%2==0)
                even[evncnt++]=arr[i];
            else
                odd[oddcnt++]=arr[i];
    }
    cout<<"The even numbers are: ";
    for(i=0;i<evncnt;i++)
        cout<<even[i]<<" ";
    cout<<"\nThe odd numbers are: ";
    for(i=0;i<oddcnt;i++)
        cout<<odd[i]<<" ";
}
```

9. Write C++ program to find the reverse of an array.



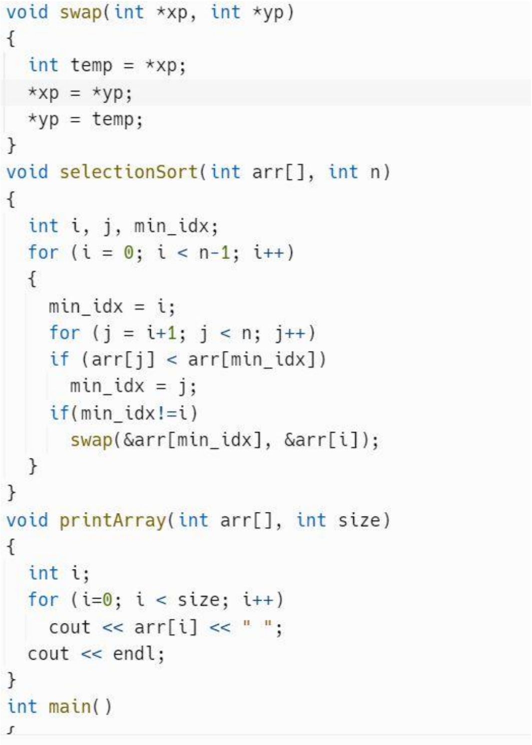
```
#include <iostream>
using namespace std;
int main(){
    int n = 9;
    int arr[n] = {2,5,6,4,7,8,3,6,4};
    int temp;
    for(int i = 0; i<n/2; i++){
        temp = arr[i];
        arr[i] = arr[n-i-1];
        arr[n-i-1] = temp;
    }
    for(int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
}
```

LAB 1. Write a program to sort elements of an array by using bubble sort.



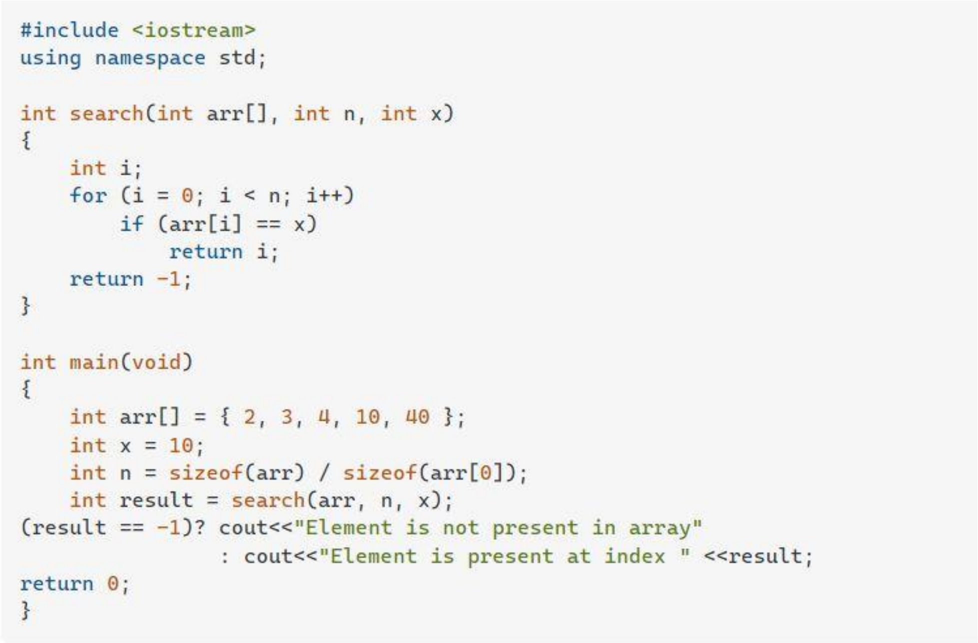
```

#include <iostream>
using namespace std;

void bubbleSort(int arr[], int n)
{
  int i, j;
  for (i = 0; i < n - 1; i++)

    // Last i elements are already
    // in place
    for (j = 0; j < n - i - 1; j++)
      if (arr[j] > arr[j + 1])
        swap(arr[j], arr[j + 1]);
}


void printArray(int arr[], int size)
{
  int i;
  for (i = 0; i < size; i++)
    cout << arr[i] << " ";
  cout << endl;
}


int main()
{
```

2. Write a program to sort elements of an array by using selection sort.



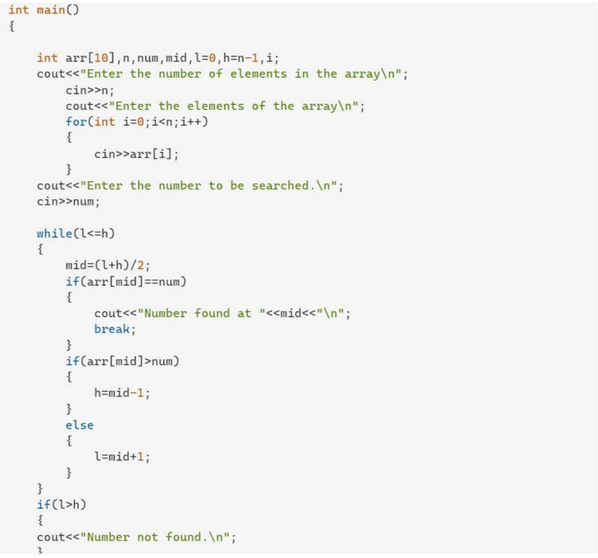
```
void swap(int *xp, int *yp)
{
  int temp = *xp;
  *xp = *yp;
  *yp = temp;
}
void selectionSort(int arr[], int n)
{
  int i, j, min_idx;
  for (i = 0; i < n-1; i++)
  {
    min_idx = i;
    for (j = i+1; j < n; j++)
    if (arr[j] < arr[min_idx])
      min_idx = j;
    if(min_idx!=i)
      swap(&arr[min_idx], &arr[i]);
  }
}
void printArray(int arr[], int size)
{
  int i;
  for (i=0; i < size; i++)
    cout << arr[i] << " ";
  cout << endl;
}
int main()
{
```

3. Write a program to search an element from the array of elements by using linear search.



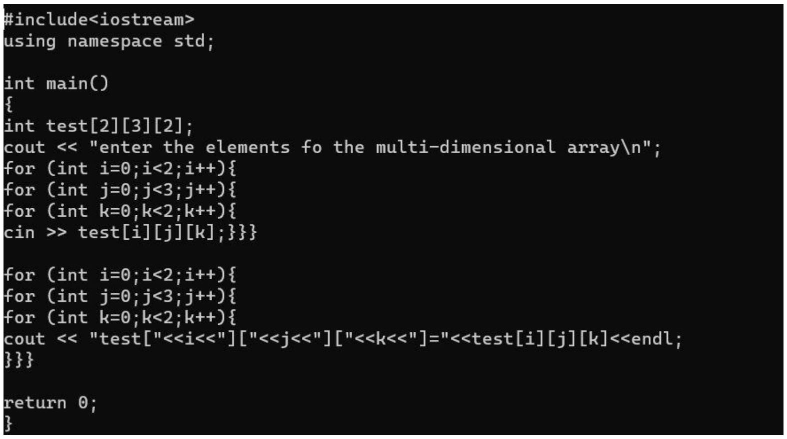
```
#include <iostream>
using namespace std;

int search(int arr[], int n, int x)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == x)
            return i;
    return -1;
}

int main(void)
{
    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10;
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = search(arr, n, x);
(result == -1)? cout<<"Element is not present in array"
                : cout<<"Element is present at index " <<result;
return 0;
}
```

4. Write a program to search an element from the array of elements by using binary search.



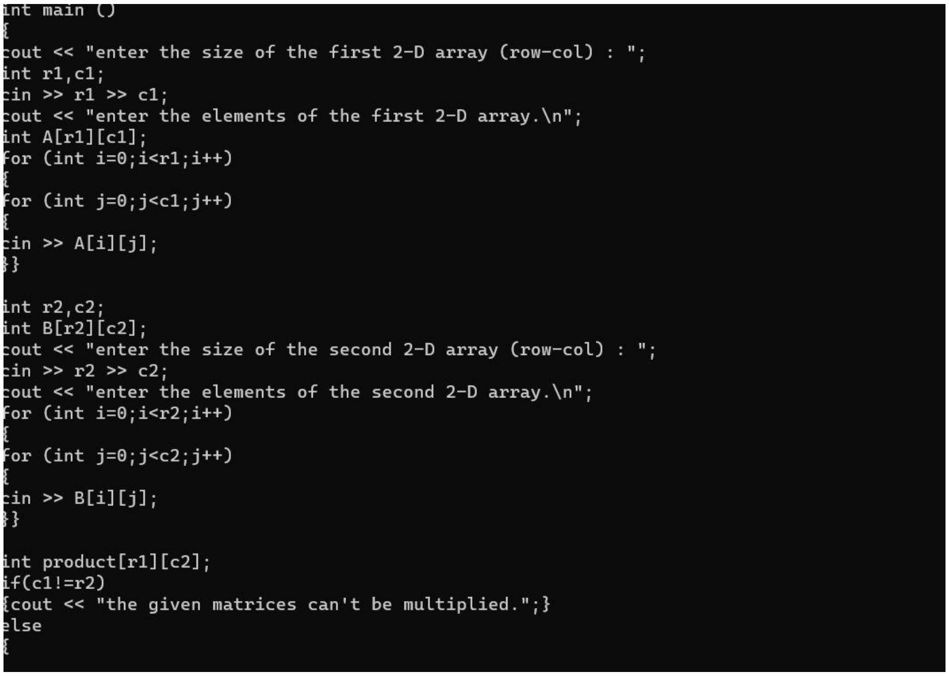
```
int main()
{

    int arr[10],n,num,mid,l=0,h=n-1,i;
    cout<<"Enter the number of elements in the array\n";
        cin>>n;
        cout<<"Enter the elements of the array\n";
        for(int i=0;i<n;i++)
        {
            cin>>arr[i];
        }
    cout<<"Enter the number to be searched.\n";
    cin>>num;

    while(l<=h)
    {
        mid=(l+h)/2;
        if(arr[mid]==num)
        {
            cout<<"Number found at "<<mid<<"\n";
            break;
        }
        if(arr[mid]>num)
        {
            h=mid-1;
        }
        else
        {
            l=mid+1;
        }
    }
    if(l>h)
    {
    cout<<"Number not found.\n";
    }
```

5. Write a program to declare the multi-dimensional array test [2][3][2] of integer type and take input from the user. Then, display them with a proper index.



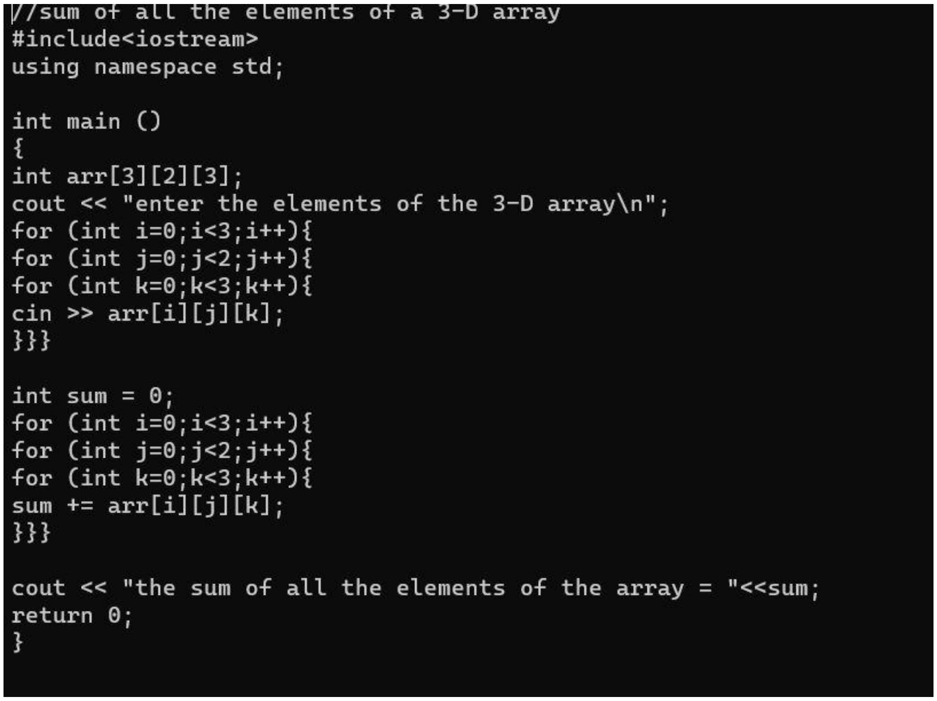
```
#include<iostream>
using namespace std;

int main()
{
int test[2][3][2];
cout << "enter the elements fo the multi-dimensional array\n";
for (int i=0;i<2;i++){
for (int j=0;j<3;j++){
for (int k=0;k<2;k++){
cin >> test[i][j][k];}}}

for (int i=0;i<2;i++){
for (int j=0;j<3;j++){
for (int k=0;k<2;k++){
cout << "test["<<i<<"]["<<j<<"]["<<k<<"]="<<test[i][j][k]<<endl;
}}}

return 0;
}
```

6. Write a program to declare two two-dimensional arrays and multiply them (in component wise). (size: var_name [3][3])



```
int main ()
{
cout << "enter the size of the first 2-D array (row-col) : ";
int r1,c1;
cin >> r1 >> c1;
cout << "enter the elements of the first 2-D array.\n";
int A[r1][c1];
for (int i=0;i<r1;i++)
{
for (int j=0;j<c1;j++)
{
cin >> A[i][j];
}}

int r2,c2;
int B[r2][c2];
cout << "enter the size of the second 2-D array (row-col) : ";
cin >> r2 >> c2;
cout << "enter the elements of the second 2-D array.\n";
for (int i=0;i<r2;i++)
{
for (int j=0;j<c2;j++)
{
cin >> B[i][j];
}}

int product[r1][c2];
if(c1!=r2)
{cout << "the given matrices can't be multiplied.";}
else
{
```

7. Write a program to find the sum of all elements of a three-dimensional array. (size: var_name [3][2][3])



```
//sum of all the elements of a 3-D array
#include<iostream>
using namespace std;

int main ()
{
int arr[3][2][3];
cout << "enter the elements of the 3-D array\n";
for (int i=0;i<3;i++){
for (int j=0;j<2;j++){
for (int k=0;k<3;k++){
cin >> arr[i][j][k];
}}}

int sum = 0;
for (int i=0;i<3;i++){
for (int j=0;j<2;j++){
for (int k=0;k<3;k++){
sum += arr[i][j][k];
}}}

cout << "the sum of all the elements of the array = "<<sum;
return 0;
}
```