(1)

```
struct Person
{
        int id;
        string name;
        int age;
};

void Sort(Person *Arr, int Size)
{
        for (int i=0;i<Size-1;i++)
        {
                int imin = i;
                int min = Arr[i].age;
                for (int j=i+1; j<Size; j++)
                {
                        if (Arr[j].age < min)
                        {
                                min = Arr[j].age;
                                imin=j;
                        }
                }
                Person tmp = Arr[imin];
                Arr[imin] = Arr[i];
                Arr[i] = tmp;
        }
}

void Input(Person *Arr, int Size)
{
        for(int i=0;i<Size;i++)
        {
                cout  << "Enter Person "<<i+1<<" ID ";
                cin >> Arr[i].id;

                cout  << "Enter Person "<<i+1<<" Name ";
                cin >> Arr[i].name;

                cout  << "Enter Person "<<i+1<<" age ";
                cin >> Arr[i].age;

        }
}

void main()
{

        int Size;
        cout << "Enter Number of Persons";
        cin >> Size;
        Person* Arr = new Person [Size];

        Input(Arr, Size);

        Sort(Arr, Size);

        for(int i=0;i<Size;i++)
```

```
       {
              cout  << "Person data "<<" ID "<<Arr[i].id<<" Name "<<Arr[i].name<<" Age 
"<<Arr[i].age<<endl;

       }
       system("Pause");
}
```

(2)

```
struct Date
{
       int Day;
       int Month;
       int Year;
};
struct Movie
{
       int ID;
       char Title[10];
       Date Release_Date;
};
void Search_Movies (Movie *AllMovies , int NumberOfMovies , Date SearchDate)
{
       cout << "Movies :" <<endl;
       for (int i = 0; i < NumberOfMovies; i++)
       {
              if(AllMovies[i].Release_Date.Year < SearchDate.Year)
              {
                     cout << AllMovies[i].Title << endl;
              }
              else if (AllMovies[i].Release_Date.Year == SearchDate.Year &&
AllMovies[i].Release_Date.Month < SearchDate.Month)
              {
                     cout << AllMovies[i].Title << endl;
              }
              else if (AllMovies[i].Release_Date.Year == SearchDate.Year &&
AllMovies[i].Release_Date.Month == SearchDate.Month && AllMovies[i].Release_Date.Day <=
SearchDate.Day)
              {
                  cout << AllMovies[i].Title << endl;
              }
       }
}
void main()
{
       cout << "Enter Release Date You want: ";
       cin>> SearchDate.Day >> SearchDate.Month >> SearchDate.Year;
       Search_Movies(AllMovies ,NumberOfMovies,SearchDate);
       delete[] AllMovies;
}
```

(3)

```
#include<iostream>
using namespace std;
struct point
{
        int x, y;
};
struct triangle
{
        point points[3];
};
bool IsIsosceles(triangle t);
float GetLen(point p1,point p2);
void main()
{
        int trianglesCount;
        cout << "Enter triangles count : ";
        cin >> trianglesCount;

        triangle * triangles = new triangle[trianglesCount];

        for (int i = 0; i < trianglesCount; i++)
        {
                for (int j = 0; j < 3; j++)
                {
                        cout << "Enter point " << j + 1 << " : ";
                        cin >> triangles[i].points[j].x;
                        cin >> triangles[i].points[j].y;
                }
        }
        for (int i = 0; i < trianglesCount; i++)
        {
                cout << "Triangle " << i + 1 << " type is ";
                if (IsIsosceles(triangles[i]))
                        cout << "isosceles " ;
                else
                        cout << "not isosceles " ;
                cout << endl;
        }
        delete[] triangles;
}
bool IsIsosceles(triangle t)
{
        float line1Len = GetLen(t.points[0], t.points[1]);
        float line2Len = GetLen(t.points[1], t.points[2]);
        float line3Len = GetLen(t.points[2], t.points[0]);
        if (line1Len == line2Len || line1Len == line3Len || line2Len == line3Len)
                return true;
        return false;
}
```

```
float GetLen(point p1,point p2)
{
	float part1 = (p2.x - p1.x)*(p2.x - p1.x);
	float part2 = (p2.y - p1.y)*(p2.y - p1.y);
	return sqrt((part1)+(part2));
}
```

(4)

```
#include<iostream>
using namespace std;
struct point
{
    int x, y;
};
struct polygon
{
    point points[4];
};
bool IsSquare(polygon p);
bool IsRectangle(polygon p);
float CalcLineLen(point p1,point p2);
void main()
{
    int polygonsCount;
    cout << "Enter polygons count : ";
    cin >> polygonsCount;

    polygon * polygons = new polygon[polygonsCount];
    for (int i = 0; i < polygonsCount; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            cout << "Enter point " << j + 1 << " : ";
            cin >> polygons[i].points[j].x;
            cin >> polygons[i].points[j].y;
        }
    }
    for (int i = 0; i < polygonsCount; i++)
    {
        if (IsSquare(polygons[i]))
        {
            cout << "Polygon " << i + 1 << " type is ";
            cout << "square" << endl;
        }
        else
        {
            cout << "Polygon " << i + 1 << " type is ";
            cout << "rectangle" << endl;
        }
    }
    delete[] polygons;
}
bool IsSquare(polygon p)
{
    float line1Len = CalcLineLen(p.points[0], p.points[1]);
    float line3Len = CalcLineLen(p.points[2], p.points[3]);
```

```
      if (line1Len != line3Len )
            return false;

      return true;
}
float CalcLineLen(point p1,point p2)
{
      float part1 = (p2.x - p1.x)*(p2.x - p1.x);
      float part2 = (p2.y - p1.y)*(p2.y - p1.y);
      return sqrt((part1)+(part2));
}
```

(5)

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

int getOccurence (int n , int *arr,int size) [3 marks] {
    int counter =0; // 0.5
    for(int i=0; i<size; i++) { // 0.5
        if(arr[i]==n) // 0.5
            counter++; // 1
    }
    return counter; // 0.5
}
int main () [7 marks] {
    int size; // all declarations 1
    cin>>size; //0.5
    int *arr = new int [size]; // 1
    int num;
    char choice;
    cout<<"Enter the”<<size<<” numbers : ";
    for(int i=0; i<size; i++) {
        cin>>arr[i]; // 0.5
    }
    do { // loop with condition 1.5
        cout<<"Enter the number ";
        cin>>num; // 0.5
        cout << "The Number "<< num<<" repeated " <<getOccurence(num,arr,size)<<" times"<<endl; // 1 for calling
        cout<<"Do you want to continue (Y/N)? ";
        cin>>choice; //0.5
    } while(choice =='y'||choice=='Y');
    delete []arr; // 0.5
}
```

(6)

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

int GetLastOccurrence(int* arr , int number , int size) [3 marks]
{
        int index=-1;

        for(int i=0;i<size;i++)
        {
                if(arr[i]==number)
                        index=i;
        }
        return index;
}
int main () [7 marks] {
   int size; // all declarations 1
   cin>>size; //0.5
   int *arr = new int [size]; // 1
   int num;
   char choice;
   cout<<"Enter the”<<size<<” numbers : ";
   for(int i=0; i<size; i++) {
      cin>>arr[i]; // 0.5
   }
   do { // loop with condition 1.5
      cout<<"Enter the number ";
      cin>>num; // 0.5
        int index = GetLastOccurrence(arr,num,size);  // 1 for calling
        if(index ==-1)
                cout<<"Number Doesn't exist"<<endl;
        else
        {
        cout<<"The number is in Index :"<<index<<endl;
        }

      cout<<"Do you want to continue (Y/N)? ";
      cin>>choice; //0.5
   } while(choice =='y'||choice=='Y');
   delete []arr; // 0.5
}
```

(7)

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

void input(int** arr, int rows, int cols);
void output(int** arr, int rows, int cols);

int main()
{
	int rows, cols;

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

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

	int* *arr = new int*[rows];
	for (int i = 0; i < rows; i++)
	{
		arr[i] = new int[cols];
	}

	input(arr, rows, cols);

	int rowIndex0, rowIndex1;
	cout << "Enter first row index: ";
	cin >> rowIndex0;

	cout << "Enter second row index: ";
	cin >> rowIndex1;

	for (int j = 0; j < cols; j++)
	{
		swap(arr[rowIndex0][j], arr[rowIndex1][j]);
	}

	output(arr, rows, cols);

	for (int i = 0; i < rows; i++)
	{
		delete[] arr[i];
	}
	delete[] arr;
	system("Pause");
	return 0;
}
void input(int** arr, int rows, int cols)
{
	cout << "Enter array: " << endl;
	for (int i = 0; i < rows; i++)
	{
		for (int j = 0; j < cols; j++)
		{
			cin >> arr[i][j];
		}
	}
}
```

```
void output(int** arr, int rows, int cols)
{
        cout << "Output array: " << endl;
        for (int i = 0; i < rows; i++)
        {
                for (int j = 0; j < cols; j++)
                {
                        cout<< arr[i][j] << "    ";
                }
                cout << endl;
        }
}
```

(8)

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

void input(int** arr, int rows, int cols);
void output(int** arr, int rows, int cols);

int main()
{
    int rows, cols;

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

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

    int* *arr = new int*[rows + 1];
    for (int i = 0; i < rows + 1; i++)
    {
        arr[i] = new int[cols];
    }

    input(arr, rows, cols);

    for (int j = 0; j < cols; j++)
    {
        arr[rows][j] = arr[0][j] + arr[rows - 1][j];
    }

    output(arr, rows + 1, cols);

    for (int i = 0; i < rows; i++)
    {
        delete[] arr[i];
    }
    delete[] arr;
    system("Pause");
    return 0;
}
void input(int** arr, int rows, int cols)
{
    cout << "Enter array: " << endl;
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            cin >> arr[i][j];
        }
    }
}

void output(int** arr, int rows, int cols)
{
    cout << "Output array: " << endl;
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            cout << arr[i][j] << "    ";
        }
        cout << endl;
    }
}
```