(1)

```
#include "iostream"
using namespace std;


void input(int* arr, int size);
void shift(int * arr, int size, int numOfShifts);
void output(int* arr, int size);

void main()
{
    int size, numOfShifts;
    cout << "Enter array size: ";
    cin >> size;

    cout << "Enter number of shifts: ";
    cin >> numOfShifts;
    int* arr = new int[size];

    input(arr, size);
    shift(arr, size, numOfShifts);
    output(arr, size);

    delete[] arr;
}

void input(int* arr, int size)
{
    cout << "Enter array" << endl;
    for (int i = 0; i < size; i++)
        cin >> arr[i];
}

void shift(int * arr, int size, int numOfShifts)
{
    for (int i = 0; i < numOfShifts; i++)
    {
        int temp = arr[0];
        for (int j = 0; j < size - 1; j++)
        {
            arr[j] = arr[j + 1];
        }
        arr[size - 1] = temp;
    }
}

void output(int* arr, int size)
{
    cout << "The Output Array" << endl;
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";

    cout << endl;
}
```

(2)

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

void Remove(int arr[], int& size, int num)
{
	for (int i = 0; i < size;i++)
		if (arr[i] == num)
		{
			for (int j = i; j < size - 1; j++)
				arr[j] = arr[j + 1];
			size--;
			break;
		}
}
void RemoveAll(int arr[], int& size, int num)
{
	for (int i = 0; i < size; i++)
		if (arr[i] == num)
			Remove(arr, size, num);
}

void main()
{
	int size;
	int* arr;
	int num;
	char ch = 'y';
	int R;
	cout << "Enter Size\n";
	cin >> size;
	arr = new int[size];
	for (int i = 0; i < size; i++)
		cin >> arr[i];

	while (ch=='y')

	{
		cout << "Enter number\n";
		cin >> num;
		cout << "Enter 1 to Remove, 2 to RemoveALL\n";
		cin >> R;
		if (R == 1)
			Remove(arr, size, num);
		else
			RemoveAll(arr, size, num);

		cout << "Array of size " << size << ":\t";
		for (int i = 0; i < size; i++)
			cout<< arr[i]<<" ";
```

```
        cout << endl;
        cout << "continue 'y' or 'n' " << endl;
        cin >> ch;
    }


}
```

(3)

```
#include<iostream>
using namespace std;
struct Train
{
      int trainNum;
      float velocity;
};
void CalcAndDisplayRemainingDistance(Train *trains,int trainsCount, float 
totalDistance, float hoursPassed);

void main()
{
      int trainsCount;
      float totalDistance,hoursPassed;
      cout << "Enter trains count : ";
      cin >> trainsCount;

      Train *trains = new Train[trainsCount];

      for (int i = 0; i < trainsCount; i++)
      {
            cout << "Enter train " << i + 1 << " (number) : ";
            cin >> trains[i].trainNum;
            cout << "Enter train " << i + 1 << " (velocity) : ";
            cin >> trains[i].velocity;
      }

      cout << "Enter the total distance : ";
      cin >> totalDistance;

      cout << "Enter time passed : ";
      cin >> hoursPassed;

      CalcAndDisplayRemainingDistance(trains, trainsCount, totalDistance, hoursPassed);

      Delete [] trains;
}




void CalcAndDisplayRemainingDistance(Train *trains, int trainsCount ,float 
totalDistance, float hoursPassed)
{
      for (int i = 0; i < trainsCount; i++)
      {
            float passedDistance = trains[i].velocity*hoursPassed;
            int remainingDistance = totalDistance - passedDistance;
```

```
        cout << "The remaining distance ( train " ;
        cout << trains[i].trainNum << " ) : ";
        cout << remainingDistance << " Km " << endl;
    }
}
```

(4)

```
#include <iostream>
using namespace std;
struct hotel       [1.5 marks]
{
    char name[30];
    float rating ;
    float pricePerNight;
};
void input( hotel arr[] ,int size) [1.5 marks]{

    for(int i=0; i<size; i++) {
        cin>>arr[i].name;
        cin>>arr[i].rating;
        cin>>arr[i].pricePerNight;
    }
}
int displayhotels(hotel arr[], int size, int pay) [4 marks] {
    float maxrating =0;   //0.5
    int maxindex=0;
    for(int i=0; i<size; i++)//0.5
    {
        if(arr[i].pricePerNight<=pay) // 0.5
        {
            cout<<arr[i].name<<" "<<arr[i].pricePerNight<<endl; // 0.5
            if(maxrating<arr[i].rating) // 0.5
            {
                maxrating=arr[i].rating; //0.5
                maxindex=i; //0.5
            }
        }
    }
    return maxindex; //0.5
}

int main() [3 marks]
{
    int size; // 0.5 all declarations
    cin>>size; //0.5 both cin
    hotel arr [10];
    input(arr,size);  //0.5
    float pay;
    cin>>pay;
    int max = displayhotels(arr,size,pay); //0.5
    cout<<"the best hotel is "<< arr[max].name<<"with rating"<<arr[max].rating; // 1
}
```

(5)

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

struct actor
{
       string Name;
       int age;
       string movies[3];
};

void DisplayCommon(actor * All, int Size, string actorName);
/*
Enter the desired actor: Ali
Ali acted with Mona in movies m1
Ali acted with Rony in movies m6  m5

*/
void main()
{
       int size;
       actor *a;
       string selected;
       cout << " Enter the number of actors:\n";
       cin >> size;
       a = new actor[size];
       cout << "Enter the actors details:\n";
       for (int i = 0; i < size; i++)
       {
              cin >> a[i].Name;
              cin >> a[i].age;
              for (int j = 0; j < 3; j++)
                     cin >> a[i].movies[j];
       }
       cout << "Enter the desired actor:\n";
       cin >> selected;
       DisplayCommon(a, size, selected);
       delete[] a;
}

void DisplayCommon(actor * All, int n, string actorName)
{
       int index = -1;
       for (int i = 0; i < n; i++)
              if (All[i].Name == actorName)
              {
                     index = i;
                     break;
              }
```

```
    for (int i = 0; i < n; i++)
    {
        if (i == index)
            continue;
        string s = "";
        for (int j = 0; j < 3; j++)
            for (int k = 0; k < 3; k++)
                if (All[i].movies[j] == All[index].movies[k])
                    s += All[index].movies[k]+" ";
        if (s.size() != 0)
            cout << actorName << " acted with" << All[i].Name << " in movies " 
<< s << endl;

    }

}
```

(6)

```
#include <string>
#include <iostream>
using namespace std;
struct Student { // [1]
    int score;
    string TestAnswers, ID;
};

string CorrectAns = "TFFTFFTTTTFFTFTFTFTT"; // [1]

int calculatescore(string Ans) //[3]
{
    int testscore = 0; // 1/2
    for (int i = 0; i < 20; i++) // for+its body 2
    {
        if (CorrectAns[i] == Ans[i])
        {
            testscore += 2;
        }
        else if (Ans[i] != ' ')
            testscore--;
    }
    return testscore;// 1/2
}
void main() //[5]
{   int studentscount; // 1/2
    cout << "Enter students count: ";
    cin >> studentscount;
    Student* Studs = new Student[studentscount]; //1
    for (int i = 0; i < studentscount; i++) // 1
    {
        cout << "Enter Student " << i + 1 << " ID: ";
        cin >> Studs[i].ID;

        cout << "Enter Student " << i + 1 << " Answers: ";
        ws(cin);
        getline(cin, Studs[i].TestAnswers);
        Studs[i].score = calculatescore(Studs[i].TestAnswers);//1
    }

    cout << endl << endl;
    cout << "Students Info:" << endl;
```

```
        for (int i = 0; i < studentscount; i++)//1/2
        {
                cout << Studs[i].ID << endl << Studs[i].TestAnswers << endl << Studs[i].score << 
endl << endl;
delete [] Studs ; // 1
        }
}
```

(7)

```
#include<iostream>
using namespace std;
struct point
{
      float X, Y;
};
struct StraightLine
{
      point start;
      point end;
      point mid;
};
void CalcMidPoints(StraightLine * lines, int size);
void main()
{
      int linesCount;
      cout << "Enter straight lines count : ";
      cin >> linesCount;

      StraightLine *lines = new StraightLine[linesCount];
      for (int i = 0; i < linesCount; i++)
      {
            cout << "Enter line " << i + 1 << " ( Start Point ): ";
            cin >> lines[i].start.X >> lines[i].start.Y;
            cout << "Enter line " << i + 1 << " ( End Point ): ";
            cin >> lines[i].end.X >> lines[i].end.Y;
      }
      CalcMidPoints(lines, linesCount);
      for (int i = 0; i < linesCount; i++)
      {
            cout <<"Line "<<i+1<<" mid point : ";
            cout << lines[i].mid.X << " "<< lines[i].mid.Y<<endl;
      }
      delete[] lines;
}
void CalcMidPoints(StraightLine * lines, int size)
{
      for (int i = 0; i < size; i++)
      {
            lines[i].mid.X = (lines[i].start.X +  lines[i].end.X) / 2.0;
            lines[i].mid.Y = (lines[i].start.Y + lines[i].end.Y)/2.0;
      }
}
```

(8)

```
#include <iostream>
using namespace std;
struct price { // 1 mark
    int pounds,piastres;
};
struct product { // 1 mark
    int ID;
    price p;
};
float calculateAverage(product arr[],int n) [3 marks]
{
    float avg,sum=0; // 0.5
    for (int i=0; i<n; i++)
        sum+=arr[i].p.pounds+arr[i].p.piastres/100.0; // 1.5
    avg=sum/n; // 0.5
    return avg; // 0.5
}

int main() [5 marks]
{
    int n; // declaration and input 0.5
    product arr[10]; // 0.5
    cin>>n;
    for(int i=0; i<n; i++)
        cin>>arr[i].ID>>arr[i].p.pounds>>arr[i].p.piastres;   //1
    float avg=calculateAverage(arr,n); // 1
    cout<<"The average is "<<avg<<endl; //0.5
    cout<<"IDs"<<endl;
    for(int i=0; i<n; i++)
        if(avg<=arr[i].p.pounds+arr[i].p.piastres/100.0) // 1
            cout<<arr[i].ID<<endl; // 0.5

}
```

(9)

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

void Concate_Alter(int *Arr1, int *Arr2, int *ResultArray, int Size)
{

	for (int i = 0; i<2 * Size; i++)
	{
		if (i % 2 == 0)
		{
			ResultArray[i] = Arr1[i / 2];
		}
		else
		{
			ResultArray[i] = Arr2[i / 2];
		}
	}



}

void display(int *Arr, int Size)
{
	for (int i = 0; i< Size; i++)
	{
		cout << Arr[i] << " ";
	}
	cout << endl;
}
void main()
{
	int Size;
	cout << "Enter Size of Arrays: ";
	cin >> Size;
	int* Arr1 = new int[Size];
	int* Arr2 = new int[Size];
	int * ResultArray = new int[2 * Size];
	cout << "Enter Values of Array 1 :";
	for (int i = 0; i<Size; i++)
	{
		cin >> Arr1[i];
	}
	cout << "Enter Values of Array 2 :";
	for (int i = 0; i<Size; i++)
	{
		cin >> Arr2[i];
	}
	Concate_Alter(Arr1, Arr2,ResultArray, Size);
```

```
        cout << "Result Array: ";
        display(ResultArray, 2 * Size);
        delete[] Arr1;
        delete[] Arr2;
        delete[] ResultArray;
}
```

(10)

```
#include <iostream>
using namespace std;
void input ( int ** arr , int size) // [1 mark]
{
    for(int i=0; i<size; i++)
        for(int j=0; j<size; j++)
            cin>>arr[i][j];
}
void display ( int ** arr , int size) //[2 mark , 1 for space and endline]
{
    for(int i=0; i<size; i++)
    {
        for(int j=0; j<size; j++)
        {
            cout<<arr[i][j]<<"  ";

        }
        cout<<endl;
    }
}

void setDiagonal (int **arr,int size) [ 2 marks or any other method]
{
    for(int i=0; i<size; i++)
        arr[i][i]=0;
}

int main() [ 5 marks]
{
    int s; // declaration and input 1
    cin>>s;
    int **arr = new int* [s]; // dynamic declaration 1.5
    for(int i=0; i<s; i++)
        arr[i] = new int [s];
    input(arr,s); // 0.5
    setDiagonal (arr,s); // 0.5
    display(arr,s); //0.5
    for(int i=0; i<s; i++) // 1 for deletion.
        delete[]arr[i];
    delete [] arr;
}
```