Q.1 Write a program to display the minimum, maximum, sum, search and average of elements of an array.

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

// 1.Write a program to display the minimum, maximum, sum, search and average of elements of an array.

class Array
{
    int size;
    int arr[];

public:
    void setData()
    {
        cout << "Enter size of array: ";
        cin >> size;
        cout << "Enter " << size << " elements of array: ";
        for (int i = 0; i < size; i++)
        {
            cin >> arr[i];
        }
    }

    int minElement()
    {
        int min = arr[0];
        for (int i = 0; i < size; i++)
        {
            if (arr[i] < min)
            {
                min = arr[i];
            }
        }

        return min;
    }

    int maxElement()
    {
        int max = arr[0];
        for (int i = 0; i < size; i++)
        {
            if (arr[i] > max)
            {
                max = arr[i];
            }
        }

        return max;
    }

    int sumOfElements()
    {
        int sum = 0;
        for (int i = 0; i < size; i++)
        {
            sum += arr[i];
        }

        return sum;
    }

    float avgOfElements()
    {
        return (sumOfElements() / float(size));
    }

    void search();
};

```

```
70. void Array::search()
71. {
72.     int n;
73.     cout << "Enter the element (n) to search: ";
74.     cin >> n;
75.     for (int i = 0; i < size; i++)
76.     {
77.         if (arr[i] == n)
78.         {
79.             cout << arr[i] << " is found at index: " << i << endl;
80.         }
81.     }
82. }
83.
84. int main()
85. {
86.     Array arr;
87.     arr.setData();
88.     cout << "Average of all elements is: " << arr.avgOfElements() << endl;
89.     cout << "Sum of all elements is: " << arr.sumOfElements() << endl;
90.     cout << "Max element of array is: " << arr.maxElement() << endl;
91.     cout << "Min element of array is: " << arr.minElement() << endl;
92.     arr.search();
93.
94.     return 0;
95. }
96.
97.
```

Q.2 Define a class student with the following specification

Private members of class student

admno	integer
sname	20 character
eng. math, science	float
total	float

Public member function of class student

ctotal()	a function to calculate eng + math + science with float return type.
Takedata()	Function to accept values for admno, sname, eng, science
Showdata()	Function to display all the data members on the screen

#include <iostream>

using *namespace* std;

```
1. class Student
2. {
3.     int admno;
4.     char sname[20];
5.     float eng, math, science, total;
6.
7. public:
8.     void takeData()
9.     {
10.        cout << "Enter admission no: ";
11.        cin >> admno;
12.        cout << "Enter surname: ";
13.        cin >> sname;
14.        cout << "Enter Marks for Englis, Maths and Science: ";
15.        cin >> eng >> math >> science;
16.    }
17.
18.    float ctotal()
19.    {
20.        total = (eng + math + science);
21.
22.        return total;
23.    }
24.
25.    void showData()
26.    {
27.        cout << "\nAdmission no: " << admno << endl;
28.        cout << "Surname: " << sname << endl;
29.        cout << "\nMarks:- \n\t"
30.             << "English = " << eng << "\n\t"
31.             << "Math = " << math << "\n\t"
32.             << "Science = " << science << "\n";
33.        cout << "Total marks: " << ctotal() << endl;
34.    }
35. };
36.
37. int main()
38. {
39.     Student s1;
40.     s1.takeData();
41.     s1.ctotal();
42.     s1.showData();
43.     return 0;
44. }
45.
```

Q.3 Define a class in C++ with following description:

Private Members

A data member Flight number of type integer

A data member Destination of type string

A data member Distance of type float

A data member Fuel of type float

A member function CALFUEL() to calculate the value of Fuel as per the following criteria

Distance	Fuel
<=1000	500
more than 1000 and <=2000	1100
more than 2000	2200

Public Members

A function FEEDINFO() to allow user to enter values for Flight Number, Destination, Distance & call function CALFUEL() to calculate the quantity of Fuel.

A function SHOWINFO() to allow user to view the content of all the data members.

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

class Flight
{
    int flight_no;
    string destination;
    float distance, fuel;

    float calFuel()
    {
        if (distance <= 1000)
        {
            fuel = 500.0f;
        }
        else if (distance > 1000 && distance <= 2000)
        {
            fuel = 1100;
        }
        else
        {
            fuel = 2200;
        }
    }

public:
    void feedInfo(int flight_no, string destination, float distance)
    {
        this->flight_no = flight_no;
        this->destination = destination;
        this->distance = distance;

        calFuel();
    }

    void showInfo();
};

void Flight::showInfo()
{
    cout << "Flight number is: " << flight_no << endl;
    cout << "Your destination is: " << destination << endl;
    cout << "Total distance from airport to " << destination << " is: " << distance << "km" << endl;
    cout << "Total fuel required is: " << fuel << "ltr.s" << endl;
}

int main()
{
    Flight indigo_i1;
    indigo_i1.feedInfo(54, "USA", 3000);
    indigo_i1.showInfo();
    return 0;
}
```

Q.4 Write a menu driven program to perform following:

a) Input a matrix

b) Display matrix

c) Add two matrix

d) Multiply two matrixes

e) Transpose a matrix

```
#include <iostream>
using namespace std;
class Matrix
{
    int mat1[3][3];
    int mat2[3][3];

    void add()
    {
        cout << "Addition of two matrices: \n";
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                cout << (mat1[i][j] + mat2[i][j]) << " ";
            }
            cout << endl;
        }
    }

    void multiply()
    {
        cout << "Multiplication of the matrices: \n";
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                int mul_element = 0;
                for (int k = 0; k < 3; k++)
                {
                    mul_element += (mat1[i][k] * mat2[k][j]);
                }
                cout << mul_element << " ";
            }
            cout << endl;
        }
    }

    void transpose()
    {
        cout << "Transpose of mat1: \n";

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                cout << mat1[j][i] << " ";
            }
            cout << endl;
        }

        cout << "Transpose of mat2: \n";

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                cout << mat2[j][i] << " ";
            }
            cout << endl;
```

```
61.         }
62.     }
63.
64. public:
65.     void inputMatrices()
66.     {
67.         cout << "Enter elements for (3X3) matrix mat1: ";
68.         for (int i = 0; i < 3; i++)
69.         {
70.             for (int j = 0; j < 3; j++)
71.             {
72.                 cin >> mat1[i][j];
73.             }
74.         }
75.
76.         cout << "Enter elements for (3X3) matrix mat2: ";
77.         for (int i = 0; i < 3; i++)
78.         {
79.             for (int j = 0; j < 3; j++)
80.             {
81.                 cin >> mat2[i][j];
82.             }
83.         }
84.     }
85.
86.     void displayMatrices()
87.     {
88.         cout << "mat1: \n";
89.         for (int i = 0; i < 3; i++)
90.         {
91.             for (int j = 0; j < 3; j++)
92.             {
93.                 cout << mat1[i][j] << " ";
94.             }
95.             cout << endl;
96.         }
97.
98.         cout << "mat2: \n";
99.         for (int i = 0; i < 3; i++)
100.         {
101.             for (int j = 0; j < 3; j++)
102.             {
103.                 cout << mat2[i][j] << " ";
104.             }
105.             cout << endl;
106.         }
107.     }
108.
109.     void operation()
110.     {
111.         int choice;
112.         cout << "Enter 1 to add the matrices, \n"
113.              << "Enter 2 to multiply the matrices, \n"
114.              << "Enter 3 to transpose the matrices, \n"
115.              << "Enter 4 exit: ";
116.
117.         cin >> choice;
118.         switch (choice)
119.         {
120.         case 1:
121.             add();
122.             operation();
123.             break;
124.         case 2:
125.             multiply();
126.             operation();
127.             break;
128.         case 3:
129.             transpose();
130.             operation();
131.             break;
132.         case 4:
133.             return;
134.
135.         default:
```

```
136.             cout << "Invalid Input!!\n";
137.             operation();
138.             break;
139.         }
140.     }
141. };
142.
143. int main()
144. {
145.     Matrix M;
146.     M.inputMatrices();
147.     M.displayMatrices();
148.     M.operation();
149.
150.     return 0;
151. }
152.
```