Lab - 01 Solution For BCS-3F FALL-2022

Task 01: Create an Animal class (having appropriate attributes and functions) and do the following steps. Dynamically allocate memory to 5 objects of a class. Order data in allocated memories by Animal names in ascending order.

```
#include <iostream>

/* run this program using the console pauser or add your own getch,
system("pause"), or input loop */
using namespace std;

class students {
 public:
      string name;
      int id;
      int operator >(students &obj){
            if(id > obj.id)
            {
                  return 1;
             }
            else
            return 0;
       }

      void sort(students *ptr){
         for (int i = 0; i < 4; i++)
        {
             for (int j = 0; j < 4 - i; j++)
            {
                  if (*(ptr + j) > *(ptr + j + 1))
                 {
                      students temp;
                      temp = ptr[j];
                      ptr[j] = ptr[j + 1];
                      ptr[j + 1] = temp;
                 }
            }
        }
    }

};

int main(int argc, char** argv) {

students arr[5];
students temp, *ptr;
     int i = 0;
     ptr = &arr[5];
     for (int i = 0; i < 5; i++)
     {
         cout << "Enter the name: ";
         cin >> arr[i].name;
         cout << "Enter the id: ";
         cin >> arr[i].id;
     }
     //applying sorting here
     ptr->sort(&arr[0]);
     cout<<"\nSorted Students objects "<<endl;
     for (i = 0; i < 5; i++)
     {
         cout << "Name is: " << arr[i].name << endl;
         cout << arr[i].id << endl;
     }
      return 0;

}
```

Task 02: Write a C++ program to read elements in a matrix and check whether the matrix is an Identity matrix or not.

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

int main()
{
      int matrix[3][3]={{1,0,0},{0,1,0},{0,0,1}};
      bool flag=true;
      for(int i=0;i<3;i++)
      {
            for(int j=0;j<3;j++)
            {
                  if(i==j)
                  {
                        if(matrix[i][j]!=1)
                              flag=false;
                  }
                  else if(matrix[i][j]!=0)
                        flag=false;
                  cout<<matrix[i][j]<<" ";
            }
            cout<<endl;
      }
      if(flag==true)
      {
            cout<<"The matrix is an identity matrix.";
      }
      else
            cout<<"The matrix is not an identity matrix.";
      return 0;
}
```

Task 03: Write a program that will read 10 integers from the keyboard and place them in an array. The program then will sort the array into ascending and descending order and print the sorted list.

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

int main()
{
      int *array = new int[10];
      int *temp = new int;
      cout<<"Enter the values you would like to store below: " <<endl;
      for(int i=0;i<10;i++)
      {
            cin>>*(array+i);
      }
      for(int i=0;i<9;i++)
      {
            for(int j=0;j<9-i;j++)
            {
                  if(*(array+j)>*(array+j+1))
                  {
                        *temp=*(array+j);
                        *(array+j)=*(array+j+1);
                        *(array+j+1)=*temp;
                  }
            }
      }
      cout<<"Array in ascending order: "<<endl;
      for(int i=0;i<10;i++)
      {
            cout<<*(array+i)<<" ";
      }
      cout<<endl<<"Array in descending order: "<<endl;
      for(int i=9;i>=0;i--)
      {
            cout<<*(array+i)<<" ";
      }
      delete temp;
      delete[] array;
      return 0;
}
```

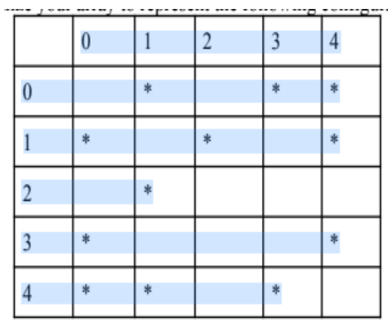
Task 04: Write a C++ program to rearrange a given sorted array of positive integers.
Note: In the final array, the first element should be the maximum value, second minimum value, third-second maximum value, fourth-second minimum value, fifth-third maximum, and so on.

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

int main()
{
      int *array = new int[10];
      int *temp = new int;
      cout<<"Enter the values you would like to store below: "<<endl;
      for(int i=0;i<10;i++)
      {
            cin>>*(array+i);
      }
      for(int i=0;i<9;i++)
```

```
    {
        for(int j=0;j<9-i;j++)
        {
            if(*(array+j)>*(array+j+1))
            {
                *temp=*(array+j);
                *(array+j)=*(array+j+1);
                *(array+j+1)=*temp;
            }
        }
    }
    cout<<"Array in ascending order: "<<endl;
    for(int i=0;i<10;i++)
    {
        cout<<*(array+i)<<" ";
    }
    cout<<endl<<"Array in descending order: "<<endl;
    for(int i=9;i>=0;i--)
    {
        cout<<*(array+i)<<" ";
    }
    delete temp;
    delete[] array;
    return 0;
}
```

Task 05: Write a program that creates a 2D array of 5x5 values of type boolean. Suppose indices represent students and that the value at row i, column j of a 2D array is true just in case i and j are studying the same course and false otherwise. Use the initializer list to instantiate and initialize your array to represent the following configuration: (* means "course-mates")



	0	1	2	3	4
0		*		*	*
1	*		*		*
2		*			
3	*				*
4	*	*		*	

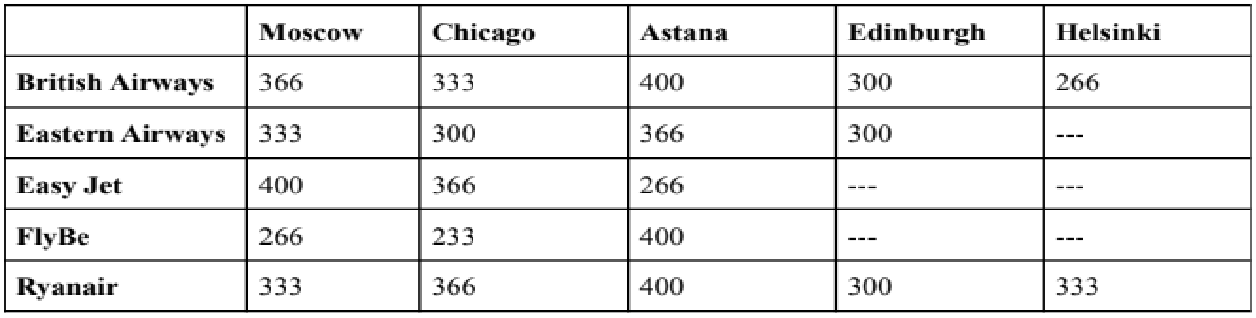
Write a method to check whether two people have a common course-mates. For example, in the example above, 0 and 4 are both course-mates with 3 (so they have a common course-mate), whereas 1 and 2 have no common course-mates.

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

int main()
{
    bool **arr= new bool*[5];
    int m,n;
    bool flag=false;
    for(int i=0;i<5;i++)
    {
        arr[i]= new bool[5];
    }
//likewise set all other indices where course is not taken ie. false.
    arr[0][0]=arr[0][2]=false;
//likewise set all other indices that are true.
    arr[0][1]=arr[0]arr[4][1]=arr[4][3]=true;
    for(int i=0;i<5;i++)
    {
        for(int j=0;j<5;j++)
        {
            if(*(*(arr+j)+i)==true)
                cout<<"* ";
            else
                cout<<"  ";
        }
        cout<<endl;
    }
    do
    {
        cout<<"Enter two numbers here: ";
        cin>>m>>n;
    }
    while(m<0 || m>4 ||n<0||n>4 || m==n);
    for(int i=0;i<5;i++)
    {

        if(*(*(arr+m)+i)==true && *(*(arr+n)+i)==true)
        {
            flag=true;
            cout<<i<<" is a common course mates of "<<m<<" and
"<<n<<endl;
        }
    }
    if (flag==false)
        cout<<"There are no common course-mates"<<endl;
    delete[] arr;
    return 0;
}
```

Task 06:
Write a program to calculate the cumulative distance in km covered by each airline. Assuming that each city may be at a max distance of 400 km from its next destination.



	Moscow	**Chicago**	**Astana**	**Edinburgh**	**Helsinki**
British Airways	366	333	400	300	266
Eastern Airways	333	300	366	300	---
Easy Jet	400	366	266	---	---
FlyBe	266	233	400	---	---
Ryanair	333	366	400	300	333

//sharing Avinash's solution, since I found it easier and short.

```
#include<iostream>
using namespace std;
int main(){
      string airline[5];
      int d, sum[5],c[5];
      int **arr = new int*[5];
int Size[5];
int i,j,k;
for(i=0;i<5;i++){
cout<<endl<<"Enter the name of Airlines: ";
cin>>airline[i];
cout<<"How many cities does it goes to? ";
cin>>c[i];
sum[i]=0;
for(j=0;j<c[i];j++){
      cout<<"Enter the distance to city "<<j+1<<" : ";
      cin>>d;
      sum[i]+=d;
}
}
for(i=0;i<5;i++){
      cout<<"The total distance covered by "<<airline[i]<<" is :
"<<sum[i]<<endl;
}
}
```