Assignment On: Bubble sort & Selection Sort

Course Code : CSE 214/215

Course Title : Algorithms & Lab

Submitted To:

Mr. Saiful Islam

Lecturer

Department of Computer Science and Engineering

Faculty of FSIT.

Daffodil International University.

Submitted By:

Md. Abdullah Ibna Harun

ID: 193-15-13426

Batch: 54

Section: A(O-1)

Department Of CSE.

Faculty of FSIT.

Daffodil International University

Date of performance : 12-02-2021

Date of Submission : 14-02-2021

1. **You are given an array A of non-negative integers of size m. Your task is to sort the array in non-decreasing order and print out the original indices of the new sorted array using selection sort.**

   Solution: Language C++14
   https://ideone.com/BVd8l2

```
#include<iostream>
#include<cstdio>
#include<string.h>
#include<stdlib.h>

using namespace std;

#define MAX 10005

void swapvalue(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;
          }
    }
        swapvalue(&arr[min_idx], &arr[i]);
    }
}
```

```
int main()
{
    int i,arr[MAX],n;
    scanf("%d",&n);
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }

    selectionSort(arr, n);

    printf("%d", arr[0]);
    for (i=1; i < n; i++)
        printf(" %d", arr[i]);
    printf("\n");

    return 0;
}
```

Sample Input	Sample output
5 4 5 3 7 1	1 3 4 5 7

2) **Coders here is a simple task for you, you have given an array of size N and an integer M.**
Your task is to calculate the difference between the maximum sum and the minimum sum of N-M elements of the given array.

Solution: Language C++14
https://ideone.com/bBafpD

```
#include<iostream>
#include<cstdio>
#include<string.h>
#include<stdlib.h>

using namespace std;

#define MAX 10005
```

```
void bubbleSort(int arr[], int n)
{
    int i, j,k, range;

    range  = n - 1;
    for(i = 0 ; i < n - 1 ; i++)
    {
        for(j = 0 ; j < range - i - 1 ; j++)
        {
            if(arr[j] > arr[j+1]){
                k = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = k;
            }
        }
    }
}


int main()
{
    int tc,i,j;
    scanf("%d",&tc);
    while(tc--)
    {
        int arr[MAX],sz,n,m,maxsum=0,minsum=0;
        scanf("%d %d",&n,&m);
        sz=n;
        for(i=0; i<sz; i++){
            scanf("%d",&arr[i]);
        }

        bubbleSort(arr, sz);

        int dif = abs(n-m);
        for(int p=1,i=0,j=sz-1; p<=dif; i++,j--,p++)
        {
            maxsum+=arr[j];
            minsum+=arr[i];
        }
        printf("%d\n",abs(maxsum-minsum));
    }
    return 0;
}
```

Sample Input	sample output
1 5 1 1 2 3 4 5	4

3) Chandler suggests that Joey should give each restaurant some points, and then choose the restaurant having maximum points. If more than one restaurant has the same points, Joey can choose the one with the lexicographically smallest name.

Solution: Language C++14
https://ideone.com/Ui907g

```
#include<iostream>
#include<cstdio>
#include<string.h>
#include<stdlib.h>

using namespace std;

#define MAX 1005

typedef struct restaurant
{
    char name[25];
    int point;
} restaurants;
```

```
void selectionSort(restaurants 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].point < arr[min_idx].point)
            {
                min_idx = j;
            }
        }

        restaurants temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
}


int main()
{
    int tc,i,j;
    scanf("%d",&tc);
    restaurants rs[tc];

    for(i=0; i<tc; i++)
    {
        scanf("%s %d",&rs[i].name,&rs[i].point);
    }

    selectionSort(rs,tc);

    int samevalue=0;
    for(i=0; i<tc-1; i++)
    {
        if(rs[i].point==rs[i+1].point)samevalue++;
    }
```

```
        if(samevalue<=1)
        {
            printf("%s\n",rs[tc-1].name);
        }
        else
        {
            char smallname[25];
            strcpy(smallname,rs[0].name);
            for(i=0; i<tc; i++)
            {
                if(strcmp(smallname,rs[i].name)>0)
                {
                    strcpy(smallname,rs[i].name);
                }
            }
            printf("%s\n",smallname);
        }

        return 0;
    }
```

Sample Input	Sample Output
3 Pizzeria 108 Dominos 145 Pizzapizza 49	Dominos