Bangladesh University of Business and Technology
Department of Computer Science and Engineering

CSE 232: Data Structures Lab

Solution to Lab 05 Tasks

Task 1

Sort a list of strings lexicographically using -

- Bubble Sort
- Selection Sort
- Insertion Sort

Sample Input:	**Sample Output:**
7	
Luna	Draco
Draco	Ginny
Ginny	Harry
Tom	Lily
Ron	Luna
Harry	Ron
Lily	Tom

Using Bubble Sort:

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

void BubbleSort(string s[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        bool swapped = false;
        for (int j = 0; j < n - i - 1; j++)
        {
            if (s[j] > s[j + 1])
            {
                string temp = s[j];
                s[j] = s[j + 1];
                s[j + 1] = temp;
                swapped = true;
            }
        }
        if (swapped == false)
            break;
    }
}

int main()
{
    int n;
    cin >> n;
    string str[n];
    for (int i = 0; i < n; i++)
        cin >> str[i];
    BubbleSort(str, n);
    for (int i = 0; i < n; i++)
        cout << str[i] << endl;
}
```

Using Selection Sort:

```cpp
#include <iostream>
using namespace std;
void SelectionSort(string s[], int n)
{
    int i;
    for (i = 0; i < n - 1; i++)
    {
        int min_idx = i;
        for (int j = i + 1; j < n; j++)
        {
            if (s[j] < s[min_idx])
                min_idx = j;
        }
        if (min_idx != i)
        {
            string temp = s[min_idx];
            s[min_idx] = s[i];
            s[i] = temp;
        }
    }
}
int main()
{
    int n;
    cin >> n;
    string str[n];
    for (int i = 0; i < n; i++)
        cin >> str[i];
    SelectionSort(str, n);
    for (int i = 0; i < n; i++)
        cout << str[i] << endl;
}
```

Using Insertion Sort:

```cpp
#include <iostream>
using namespace std;
void InsertionSort(string s[], int n)
{
    for (int i = 1; i < n; i++)
    {
        string key = s[i];
        int j = i - 1;
        while (j >= 0 && s[j] > key)
        {
            s[j + 1] = s[j];
            j--;
        }
        s[j + 1] = key;
    }
}
int main()
{
    int n;
    cin >> n;
    string str[n];
    for (int i = 0; i < n; i++)
        cin >> str[i];
    InsertionSort(str, n);
    for (int i = 0; i < n; i++)
        cout << str[i] << endl;
}
```

Task 2

Suppose there are **n** coins on a table. Find out the ***minimum*** number of coins you need to take from those **n** coins so that the amount of money you have is ***strictly greater*** than the amount left on the table. The input consists of **n**. Followed by the monetary value of each of those coins. You should print the number of coins you need to take.

Sample Input:	**Sample Output:**
2 3 3	2
Sample Input: 3 2 1 2	**Sample Output:** 2
Sample Input: 20 7 84 100 10 31 35 41 2 63 44 57 4 63 11 23 49 98 71 16 90	**Sample Output:** 6

Sample Solution:

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

void InsertionSort(int arr[], int n)
{
    for (int i = 1; i < n; i++)
    {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] < key)
        {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}

int main()
{
    int n, i, j, arr[100], cnt = 0, total = 0, me = 0;
    cin >> n;
    for (i = 0; i < n; i++)
    {
        cin >> arr[i];
        total = total + arr[i];
    }
    total = total / 2;
    InsertionSort(arr, n);
    for (i = 0; i < n; i++)
    {
        me = me + arr[i];
        cnt++;
        if (me > total)
            break;
    }
    cout << cnt << endl;
}
```