Dept. of CSE, Bennett University
ECSE379L – Programming Using C++

Lab Assignment – 6

In this lab, you will write a C++ program to solve the following problem.

(1) Being a great fan of the legend Asrani, a computer programmer plans to pay tribute to him by implementing his famous dialog from the movie Sholay :"Aadhe idhar jaao, aadhe udhar jaao" using recursive function. Names Of soldiers are stored dynamically in a list as soon as name='end' or 'End' is encountered, the list is done. The list is then splitted into two halves (Left List and Right List) till all the final list contains single element.

Test Case 1:
Input:
Anil
Mukesh
Raj
suresh
ramesh
end

Output:
Splitting['Anil', 'Mukesh', 'Raj', 'suresh', 'ramesh']
Left List=['Anil', 'Mukesh'] Right List=['Raj', 'suresh', 'ramesh']
Splitting['Anil', 'Mukesh']
Left List=['Anil'] Right List=['Mukesh']
Splitting['Anil']
Splitting['Mukesh']
Splitting['Raj', 'suresh', 'ramesh']
Left List=['Raj'] Right List=['suresh', 'ramesh']
Splitting['Raj']
Splitting['suresh', 'ramesh']
Left List=['suresh'] Right List=['ramesh']
Splitting['suresh']
Splitting['ramesh']

Test Case 2:
Input:
Eva
Naveen
Manish
Shivendra
End

Output:
Splitting ['Eva', 'Naveen', 'Manish', 'Shivendra']
Left List=['Eva', 'Naveen'] Right List=['Manish', 'Shivendra']
Splitting ['Eva', 'Naveen']

Left List=['Eva'] Right List=['Naveen']
Splitting ['Eva']
Splitting ['Naveen']
Splitting ['Manish', 'Shivendra']
Left List=['Manish'] Right List=['Shivendra']
Splitting ['Manish']
Splitting ['Shivendra']

(2) Some of the PhD holders are selected for interview at IIT, Roorkee for assistant professor post. There are some candidates that seem outstanding based on their API score. So, interviewer's panel wants to identify those candidates and assigns a separate interviewing panel for them. The objective for this to arrange the API scores in non-decreasing order. The interviews are done based on their API score in non-decreasing order. Given an array of N distinct elements, find the minimum number of swaps required to sort the array.

Testcase1

Input:

4

4 3 2 1

where:

- First line represents the number of elements in the array.
- Second line represents the elements in the array.

Output:

2

Explanation: Swap elements at index 0 with 3 and 1 with 2 to form the sorted array {1, 2, 3, 4}.

(3) In a class, 'N' students are asked to bring number of dishes as per their roll numbers, for instance roll number 1 will bring 1 item, roll number 2 will bring 2 items, roll number 3 will get 3 items and so on. Write the function to get the total number of items brought by "N" students.

Testcase 1
Input:
7

Output:
28

(4) MTV Roadies is a popular show across youths in India and provides fame to its participants. Most of the time it's all about participant's physical strength. So the makers of the show introduce a task named as 'Brain Vs Brawn' Challenge, where participants are divided in two groups 'Brain' and 'Brawn'. Brawn team performs a physical task while Brain team need to answer a mathematical riddle. One of Judge provides a number N1 to the Brain team and asks a smallest number N2 in return such that the multiplication of digits of N2 is equal to N1. For example, they provide a number 36 and expect 49 from any member of Brain team. Suppose, one of your friend is the member of Brain team then write a program in Python to help your friend using modules.

Test Case 1:
Input: 7
Output: 17

Test Case 2:
Input:36
Output: 49

Test Case 3:
Input: 13
Output: Not Possible

(Optional)

(5) Given a matrix **M** of **R** rows and **C** columns, find the maximum sum of an hour-glass in a matrix.

An hour-glass is made of 7 cells in the following form.

```
A B C
  D
E F G
```

It is evident from above that to find hour-glass the number of rows and number of columns in a matrix must be greater than or equal to 3.
In a matrix total number of **hour-glasses** is **(R-2)*(C-2)**.

Eg: For matrix **M[][]** =
```
2 3 0 0 0
0 1 0 0 0
1 1 1 0 0
0 0 2 4 4
0 0 0 2 0
```

Possible hour-glasses are:
```
2 3 0      3 0 0      0 0 0
  1          0          0
1 1 1      1 1 0      1 0 0

0 1 0      1 0 0      0 0 0
  1          1          0
```

```
0 0 2       0 2 4       2 4 4

1 1 1       1 1 0       1 0 0
  0           2           4
0 0 0       0 0 2       0 2 0
```

Input:

```
5
5
1 1 1 0 0
0 1 0 0 0
1 1 1 0 0
0 0 0 0 0
0 0 0 0 0
```

where:

- First line represents **R**, the number of rows.
- Second line represents **C,** the number of columns.
- Third to seventh line represents the matrix elements.

Output:

```
7
```

Explanation: Below is the hour-glass with maximum sum:

```
1 1 1
  1
1 1 1
```

and its sum is 7, hence the output 7.