Indian Statistical Institute

Semester-I 2019-2020

M.Tech.(CS) - First Year

Lab Test 4 (8 November, 2019)

Subject: Data and File Structures Laboratory

Total: 60 marks Duration: 4 hrs.

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `cs19xx-test4-progy.c`
   IMPORTANT: Insert a single alpha-numeric string of your choice, 6-8 characters long, in the name given above as shown in the examples below. Think of this string as something like a security password, except that you are not required to remember the string. Examples: `cs1940-assign3-x19jdh4-prog1.c`, `ppo03wws-cs1940-assign3-prog2.c`, `cs1940-assign3-prog2-jsiwm7de.c`
2. When you have finished, copy all your files to `~dfslab/2019/labtest4/cs19XX/` on 192.168.64.35 using
   `scp <file names> cs19XX@192.168.64.35:~dfslab/2019/labtest4/cs19XX/`

1. **(20 marks)** Suppose a game is being played by a set S of people having distinct ages. Each participant knows all the other participants' ages. The game requires each participant p to choose a partner p' (with replacement) from the set. Note that multiple players are permitted to choose the same partner. Player p then gets a score proportional to the `Bitwise XOR` value of the ages of p and p'. The pair that receives the maximum score is declared as the winning pair.

   Write a program that takes a series of names and their ages and returns the names of the winning pair. Note that, finding out the `Bitwise XOR` of all the possible pairs of ages might not be the most efficient approach for large number of inputs.

   Input Format

   Input will be provided via standard input in the following format. If there are n players, the input will consist of n lines. Each line of input will consist of a string and an integer, corresponding to the name and the age of a participant, respectively.

   Output Format

   Output is to be printed on the standard output in the following format. The output will print the names of the winning pair. The name of the younger partner in the pair should be printed before the name of the older partner. The names must be separated by a space. If there is a tie, simply print `TIE` on the standard output.

   Sample Input 0

   ```
   Bulbul 3
   Fani 2
   Vayu 1
   ```

Sample Output 0

```
Vayu Fani
```

Sample Input 1

```
Third 3
Fourth 4
Eighth 8
Fifteenth 15
```

Sample Output 1

```
TIE
```

Sample Input 2

```
infant 3
boy 15
child 7
```

Sample Output 2

```
infant boy
```

2. **(20 marks)** Suppose a set of data items are extracted from a data repository. The data items are homogeneous and belong to one of the following types: integer, character, or real values. Our target is to return the data items that fall (inclusively) within a given range. The term range is defined as usual based on lexicographic ordering of the data items.

   Write a program that takes the name of an input file that contains the data items and a range (of values) from the user and returns the corresponding data items in ascending order.

   Input Format

   Inputs will be provided as command-line arguments in the following format. The first argument passed by user denotes the filename that contains the data items. This is followed by a pair of values (lower and then upper bound) representing the range. Finally, there is a flag character representing the type of data. The character is one of the following: 'i' (for integer), 'c' (for character), or 'r' (for real values).

   The input file will include the number of data items in the first line. This will be followed by the data items separated by spaces.

   Output Format

   Output is to be printed on the standard output. Output will print the data items that inclusively fall within the given range in ascending order.

Command-line Arguments

```
./prog2 <input_filename> <lower_bound> <upper_bound> <data_type>
```

Sample Input 0

```
Contents of input.txt:
7
1 5 2 4 7 3 9

Input from command-line:
./prog2 input.txt 4 7 i
```

Sample Output 0

```
4 5 7
```

Sample Input 1

```
Contents of input.txt:
5
1 1.1 1.01 1.001 1.0001

Input from command-line:
./prog2 input.txt 1.005 1.5 r
```

Sample Output 1

```
1.01 1.1
```

Sample Input 2

```
Contents of input.txt:
6
a c b g d e

Input from command-line:
./prog2 input.txt b e c
```

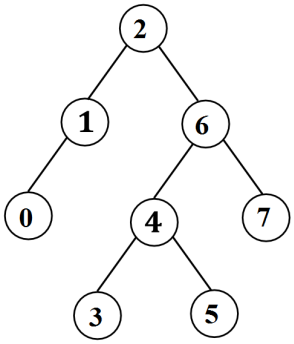
Sample Output 2

```
b c d e
```

3. **(20 marks)** A path in a tree defines the finite sequence of edges that joins a sequence of distinct vertices. Let us define the number of turns in a path as the number of switches (that appear either as `parent - left child - right child` or as `parent - right child - left child`) while traversing the path in a given tree.

   Given the pre-order traversal of a binary search tree (BST), write a program to find out the length of the path, which starts from the root and reaches to one of the leaf nodes, that has the maximum number of turns in the tree. Note that, a straight path has no turns, hence the number of turns is zero. If there is a tie among multiple paths having the same number of turns, return the shortest path. If still there is a tie, return all possible shortest paths having the maximum number of turns.

   As for example, the binary tree shown below has the path 2-6-4-5, of length 3, that has the maximum number of turns (with a pair of turns at 2-6-4 and 6-4-5).



Input Format

Input will be provided via standard input in the following format. The first line of input consists of an integer, namely the number of data items in the BST. It follows by another input line that consists of integers corresponding to the data items obtained from the pre-order traversal on the BST.

Output Format

Output is to be printed on the standard output in the following format. The output simply prints the path, by the data items it contains that are separated by hyphens, and its length in the next line. The path has the maximum number of turns, and in case there is a tie, has the minimum length. Return all such possible shortest paths.

Sample Input 0

```
8
2 1 0 6 4 3 5 7
```

Sample Output 0

```
2-6-4-5
3
```

Sample Input 1

```
6
5 3 1 4 8 9
```

Sample Output 1

```
5-3-4
2
```

Sample Input 2

```
7
6 2 1 3 8 7 9
```

Sample Output 2

```
6-2-3
2
6-8-7
2
```