COMPUTING LAB – ASSIGNMENT 2

MTech(CS) I year   2020–2021

Deadline: 15 December, 2021

Total: 60 marks

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `cs21xx-assign2-progy.c` (assuming `cs21xx` denotes your roll number and `progy` denotes the program number).
2. To submit the solution files (`.c` or `.h`), ensure that they not password protected and mail them together to `<assignisik@gmail.com>` with the subject line as follows: `MTech (CS) 2021-23 cs21xx Computing Lab - assignment2`.
3. You may consult or use slides / programs provided to you as course material, or programs that you have written yourself as part of classwork / homework for this course, but please **do not** consult or use material from other Internet sources, your classmates, or anyone else.
4. Please make sure that your programs adhere strictly to the specified input and output format. **You may lose marks if your program violates the input and output requirements.**
5. Submissions from different students having significant match will be **debarred from evaluation**.

NOTE: Unless otherwise specified, all programs should take the required inputs from stdin, and print the desired outputs to stdout.

Q1. Given two positive integers k and n, write a program that employs a recursive function to efficiently calculate the value of k^n. Note that, an efficient calculation will require $O(\log n)$ multiplications using the following recursion.

$$k^n = \begin{cases} (k^2)^{\frac{n}{2}} & \text{when } n \text{ is even} \\ k * (k^2)^{\lfloor \frac{n}{2} \rfloor} & \text{when } n \text{ is odd} \end{cases}$$

Your program must not be using any library function for power calculation.

[20 marks]

Input Format
The input (to be read from stdin) is the two integers k and n separated by spaces in the same line.

Output Format
The output (to be printed to stdout) is the value of k^n. For inappropriate inputs the output should be `INVALID INPUT`.

Sample Input 0
3 0

Sample Output 0
INVALID INPUT

Sample Input 1

3 10

Sample Output 1

59049

Sample Input 2

1.1 2

Sample Output 2

INVALID INPUT

Q2. Given an English alphabet matrix (comprising the characters a-z or A-Z as its elements), a pair of cells are defined to be *connected vowels* if both of them contains vowels and they are adjacent (either horizontally, vertically, or diagonally) to each other. Write a program to find out the number of cells in the maximum region of connected vowels in the given matrix.

[20 marks]

Input Format

The first line of the input (to be read from stdin) comprises the number of rows and columns in the English alphabet matrix. This will be followed by the elements of the input matrix provided in a row-major fashion and separated by spaces.

Output Format

The output (to be printed to stdout) will show the number of cells in the maximum region of connected vowels in the given matrix.

Sample Input 0

```
5 5
a a a b b
a a a b b
b b a b a
a b b b a
b a b a a
```

Sample Output 0

7

Sample Input 1

```
2 10
a z e b f a g h j o
s i o u r w d e t i
```

Sample Output 1

5

Sample Input 2

```
6 7
E B b C c D d
M a n x f G H
q y I Q P P P
U H J u z k K
s o s s A S S
n D i n n e L
```

Sample Output 2

```
6
```

Q3. A path in a tree defines the finite sequence of edges that joins a sequence of distinct vertices. Let us define the Z-turn in a path as the number of switches that appear as Z (`parent - right child - left child - right 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 a Z-turn 2-6-4-5, of length 3
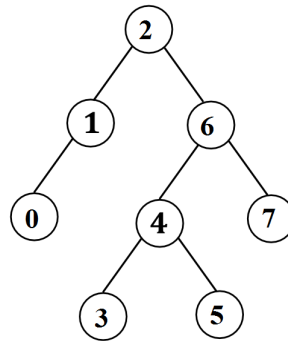
[20 marks]

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
```