COMPUTING LAB – ASSIGNMENT 1

MTech(CS) I year   2020–2021

Deadline: 30 November, 2021

Total: 60 marks

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `cs21xx-assign1-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 - assignment1`.
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. Let us define the `DOMINATING FOLDING` of a string of length n as the folding of a string from the middle such that the bigger (by ASCII value) of the characters at the indices i and $(n-i)$ are retained. If the folding happens from the left or right, the bigger of ith and $(n-i)^{th}$ characters are retained at the end or beginning, respectively. Note that the `DOMINATING FOLDING` will turn the string into exactly half of its length except for the cases where the length of the string is odd. Write a program that takes a string and a series of folding instructions and returns the result.

[20 marks]

Input Format
The input (to be read from stdin) comprises the input string followed by a series of folding instructions represented either as 'L' or 'R', denoting `DOMINATING FOLDING` from the left or right, respectively.

Output Format
The output (to be printed to stdout) will show the intermediate and final strings after performing the `DOMINATING FOLDING`(s). The output strings are to be separated by space in the same line at which they are printed.

Sample Input 0
Computing LL

Sample Output 0
utmog mtu

Sample Input 1
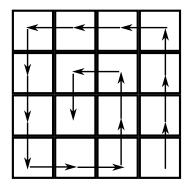
123456 LRL

Sample Output 1
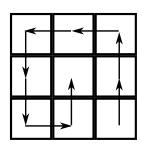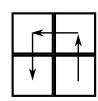
456 65 6

Sample Input 2

+-*/ LLLLL

Sample Output 2

-/ / / / / /

Q2. Consider the traversal of elements in a two-dimensional integer matrix as shown in the following figure. Note that the traversal starts from the bottom-right corner and pursue the directions up-left-down-right recursively. Write a program that takes a two-dimensional matrix as user input and returns the elements traversed in the said way. Your should write a recursive function for the traversal.

[20 marks]

Input Format

The input (to be read from stdin) is a two-dimensional integer matrix. The first line of the input comprises the row and column numbers of the matrix. This will be followed by the entries of the matrix in row-major fashion.

Output Format

The output (to be printed to stdout) is the list of traversed elements.

Sample Input 0

2 2
1 2
3 4

Sample Output 0

4 2 1 3

Sample Input 1

3 3
1 2 3
4 5 6
7 8 9

Sample Output 1
9 6 3 2 1 4 7 8 5

Sample Input 2
4 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

Sample Output 2
16 12 8 4 3 2 1 5 9 13 14 15 11 7 6 10

Q3. Recall that an ASCII string is a string that comprises ASCII characters only. Write a program that takes an ASCII string as a command-line argument and expands the non-numeric characters that are preceded by a numeric character within it. An expansion of a character increases the occurrence of it by one. The preceding numeric character will denote the number of times it has to be expanded. Hence, not all the characters will necessarily get expanded. The numeric characters used for expanding the succeeding non-numeric characters disappear in the resulting string, while the rest of the numeric characters remain. [20 marks]

Input Format
The input (to be read as command-line argument) is an ASCII string.

Output Format
The output (to be printed to stdout) is the expanded string.

Sample Input 0
d2ata

Sample Output 0
daaata

Sample Input 1
245

Sample Output 1
245

Sample Input 2
tricky3.50c

Sample Output 2
tricky....5c