CSDC101 Fundamentals of Computing
Laboratory Exercise 6

PROBLEMS

1) Count Upper

Count how many uppercase letters are in the input.

INPUT FORMAT

Input begins with an integer N indicating how many character sequences are in the input. Succeeding lines begin with an integer t indicating how many characters ch follow.

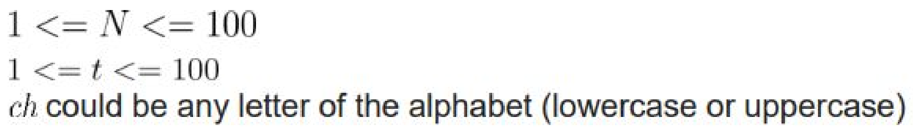
OUTPUT FORMAT

For each input, output how many uppercase letters are found.

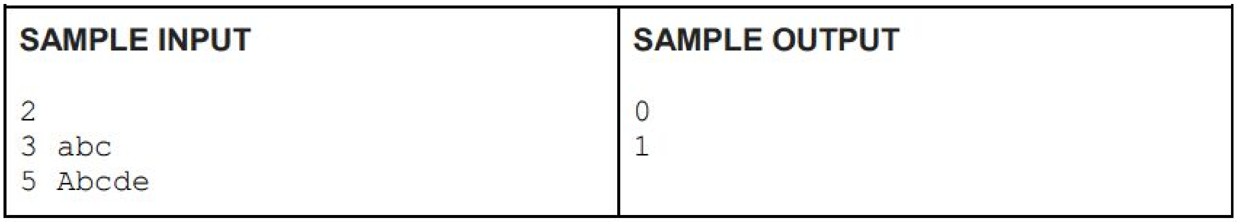
PROGRAM SPECIFICATION

Program must have a function called count_upper that takes two arguments: an integer denoting the length a character sequence and a string representing the character sequence. It counts the number of uppercase letters found in the character sequence. It returns an integer value indicating how many uppercase letters the sequence contains.

CONSTRAINTS



$1 <= N <= 100$
$1 <= t <= 100$
ch could be any letter of the alphabet (lowercase or uppercase)



SAMPLE INPUT	SAMPLE OUTPUT
2 3 abc 5 Abcde	0 1

2) My Power

Write a program that computes for x raised to y. Your program must have a function called **my_power** that takes two integer arguments – the first being the base and the second being the exponent. It returns an integer result which is the computed base raised to exponent value.

**Note: You are not allowed to use the pre-defined* pow *function.*

INPUT FORMAT

Input contains arbitrary pairs of integers x and y, where x is the base and y is the exponent.

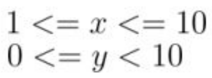
OUTPUT FORMAT

For each pair, compute for x raised to y. Output should be separated by a new line.

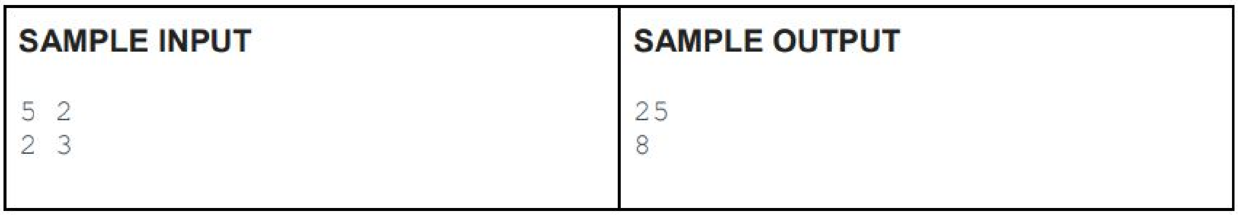
PROGRAM SPECIFICATION

Your program must have a function called **my_power** that takes two integer arguments – the first being the base and the second being the exponent. It returns an integer result which is the computed base raised to exponent value.

CONSTRAINTS



$$1 <= x <= 10$$
$$0 <= y < 10$$



SAMPLE INPUT	SAMPLE OUTPUT
5 2 2 3	25 8

3) Alpha Flip

Convert all uppercase letters to lowercase and vice-versa. If input is not a valid letter of the alphabet, replace it with “***bug here!***” without the quotation marks.

INPUT FORMAT

Input begins with an integer N indicating how many character sequences are in the input. Succeeding lines begin with an integer t indicating how many characters ch follow.

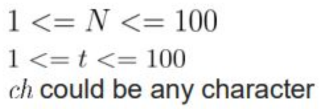
OUTPUT FORMAT

For each input, output the character sequences with their cases flipped.

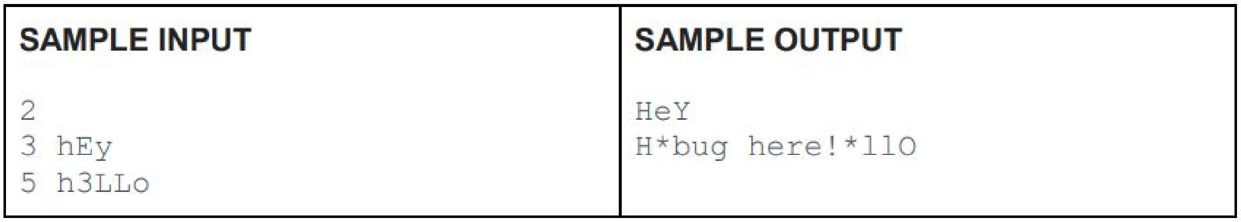
PROGRAM SPECIFICATION

Your program must have a function called **alpha_flip** that takes a string argument and that also returns a string. The function performs the conversion from lowercase to uppercase and vice versa. It returns the converted string.

CONSTRAINTS



$1 <= N <= 100$
$1 <= t <= 100$
ch could be any character



SAMPLE INPUT	SAMPLE OUTPUT
2 3 hEy 5 h3LLo	HeY H*bug here!*llO

4) Yassi’s List

Yassi says *"I have a list of integers, write me a program to perform simple arithmetic operations on certain elements of my list. Valid operations are addition, subtraction, and multiplication."*

INPUT FORMAT

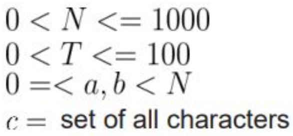
Input begins with an integer N indicating the number of integers in Yassi's list. Followed by an integer T, the number of test cases. Next line is a sequence of N integers that comprise the list. Subsequent are T test cases, each with 2 integers a and b and a character c. a and b are the positions of the list elements that will be used as operands in operation c.

Positions given by a and b begin from 1 (e.g. 1 pertains to the first value in the list).

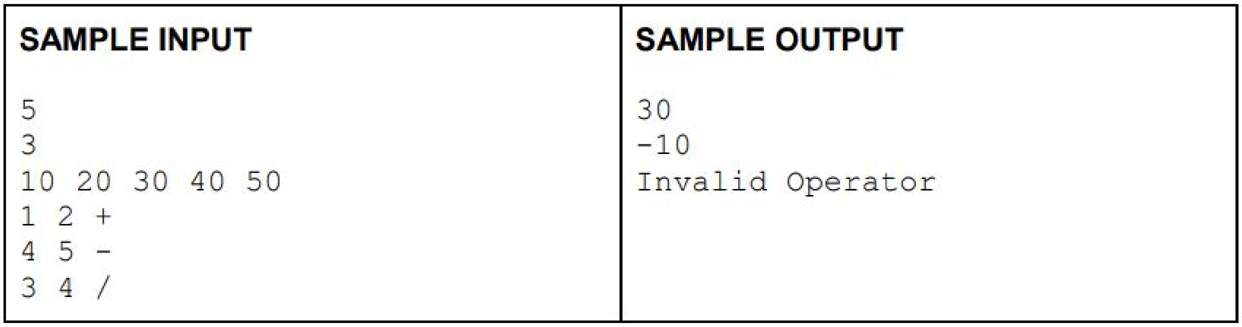
OUTPUT FORMAT

For each test case, output the result of performing operation c on a and b . If a and/or b are invalid positions, display "**Out of Bounds**". If c is not a valid operation, display "**Invalid Operator**". If both errors exist, display "**Out of Bounds and Invalid Operator**".

CONSTRAINTS



$0 < N <= 1000$
$0 < T <= 100$
$0 =< a, b < N$
$c =$ set of all characters



SAMPLE INPUT	SAMPLE OUTPUT
5 3 10 20 30 40 50 1 2 + 4 5 - 3 4 /	30 -10 Invalid Operator