

CS2.201: Computer Systems Organization
Spring 2024
International Institute of Information Technology, Hyderabad

Assignment 2

Deadline: 23:59, 12/04/2024

Instructions: Read the below instructions carefully before you start working on the assignment.

- Writing complete code with successful execution guarantees full marks. Ensure that all edge cases are handled.
- Strict plagiarism checks will be performed on all submissions. Any and all forms of plagiarism will result in zero marks for this assignment.
- Hard coded solutions will get a straight zero.
- Comment your code properly explaining why you are doing , what you are doing.
- Total marks for assignment is 50(30 marks for working code submission, 20 marks for viva).
- C files should be used only to take input, call the function and give output. The complete logic of the function itself should be in Assembly file, failing which, you will be awarded zero for that particular question.
- For the questions which require you to return an array as output, you can create an array for output in C and pass its pointer as one of the arguments to your function.

Submission format: Strictly adhere to the following submission format. Failure to do so will result in penalty.

- The following directory structure is expected,

```
./<roll_number>
├── q1
│   ├── q1.s
│   └── q1.c
├── q2
│   ├── q2.s
│   └── q2.c
├── q3
│   ├── q3.s
│   └── q3.c
├── q4
│   ├── q4.s
│   └── q4.c
└── q5
    ├── q5.s
    └── q5.c
```

- Zip the ./<roll_number> folder and name the zipped folder as <roll_number>_assign2.zip

Assume all the integer variables to be long long int.
Time-limit for all the questions is : 1 second

Problem 1:

6 marks

Given an array of N integers, calculate the maximum subarray sum with the additional constraint that the length of the subarray should be between L and R. L and R will be given in the input.

Input/Output Format

- INPUT:
N L R
 a_1, a_2, \dots, a_N
- OUTPUT:
M, Integer denoting max sum subarray satisfying the constraint
- Constraints :
 $1 \leq N \leq 10^5$
 $1 \leq L \leq R \leq N$
 $-10^6 \leq a_i \leq 10^6$

Sample Test Case

Input:

4 3 4
-200 -9 100 200

Output:

291

Problem 2:

6 marks

Implement a recursive function $\binom{n}{r}$ that calculates the number of combinations of selecting r items from a set of n items in $O(r)$ time complexity. The function should take two arguments, n and r, and return the result.

Input/Output Format

- INPUT:
N R
- OUTPUT:
Integer denoting the number of combinations of selecting r items from a set of n items
- Constraints :
 $1 \leq n \leq 10$
 $0 \leq r \leq n$

Sample Test Case

Input:

4 2

Output:

6

Problem 3:

6 marks

Implement a function `evaluatePostfix` that takes a single string representing a postfix expression (also known as Reverse Polish Notation) and returns the result of the expression. The expression will contain single-digit integers and the operators `+`, `-`, `*`, and `/`. Assume division is integer division, which truncates towards zero.

Input/Output Format

- **INPUT:**
N , length of string
S, The string literal representing a postfix expression. The operands and operators are separated by at least one space.
- **OUTPUT:**
An integer denoting the result of evaluating the postfix expression
- **Constraints :**
 - $N \leq 10^5$
 - The expression will not lead to division by zero

Sample Test Cases

Input:

9

2 3 + 5 *

Output:

25

Problem 4:

6 marks

You are keeping the scores for a baseball game with strange rules. At the beginning of the game, you start with an empty record.

You are given a list of strings `operations`, where `operations[i]` is the i^{th} operation you must apply to the record and is one of the following:

An integer `x` : Record a new score of `x`.

'+' : Record a new score that is the sum of the previous two scores.

'D' : Record a new score that is the double of the previous score.

'C' : Invalidate the previous score, removing it from the record.

Return the sum of all the scores on the record after applying all the operations

Input/Output Format

- **INPUT:**
N
 $a_1, a_2..a_N$ (integers/characters representing `operations[i]`)
- **OUTPUT:**
Sum of all scores on record
- **Constraints :**
 - $1 \leq N \leq 1000$
 - `operations[i]` is "C", "D", "+", or a string representing an integer in the range [-100, 100].

Sample Test Cases

Input:

8

5 -2 4 C D 9 + +

Output:

27

Problem 5:

6 marks

Given a sorted list L of 32 signed Integers each of 16 bits and a key X (16 bit). Write a function to find the index of first occurrence of X within L , return -1 if X does not belong to L . The list L is stored in memory inside the data section. You have to use Binary Search to look for X within L . Report the number of iterations taken to successfully/unsuccessfully find X (Value may depend on implementation).

• INPUT:

 L_0, L_1, \dots, L_{31} X

• OUTPUT:

Index of first occurrence X in L if X is in L otherwise -1, and the number of iterations taken**Sample Test Case**

Input :

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 18 20 21 22 23 24 25 26 27 28 29 30 31 32
18

Output:

17 5

All the Best!