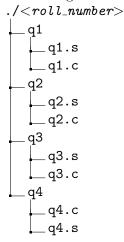# Assignment 2
Deadline: 23:55, 31/05/2023

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

- Writing complete code with successful execution guarantees full marks. Failure on test cases will result in penalisation. Therefore ensure 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.

- Write well-organised code using procedures for repeated operations.

- Hard coded solutions will get a straight zero.

- Comment every line of your code and justify why you write that statement. Total of 5 marks for commented code.

- Total marks for assignment is 50 (30 marks for working code submission, 20 marks for viva).

- C files are given for each question. Write the assembly code for the required function declared in these files. **DO NOT** modify the given C files.

Submission format: Strictly adhere to the following submission format. Failure to do so may result in an erroneous evaluation of your assignment.

- 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.c
    └── q4.s
```

- 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.

Problem 1: 10 marks

Write an assembly language program to determine whether a given string of characters, consisting of parentheses (including '(','{', and '['), is balanced or not. The program should use stack operations to solve the problem. The program should take an input string and its length as arguments and return 1 if the parentheses in the string are balanced, and 0 otherwise.

A balanced string is defined as one where each opening parenthesis has a corresponding closing parenthesis in the correct order. For example, the string "([]){}" is balanced, while the string "{[()}]" is not balanced. You can assume input string only contains parentheses.

Input/Output Format

- INPUT: Contains two lines. First line contains a single value N, size of the string. ($1 \leq N \leq 50$) The second line contains an N-length input string.

- RETURN VALUE: 1 for balanced string or 0 for not balanced

- OUTPUT: Balanced or Not balanced.

Sample Test Case
Input:
4
{()}
Output:
Balanced

Problem 2: 12 marks

You are given a sorted array of integers and a search value. Your task is to write an assembly language program that implements the binary search algorithm using recursion to search for the given value in the array.

The program should implement a recursive binary search function to find the index of the search value in the array. If the value is found, the program should return the index. If the value is not found, the program should return -1. (0 based indexing should be followed for solution).

Input/Output Format

- INPUT: Contains three lines. First line contains a single value N, size of the array. ($1 \leq N \leq 50$) The second line contains N distinct values in sorted order $a_1, a_2 \ldots a_n$, elements of the array. ($-10^9 \leq a_i < 10^9$). The third line contains the search value

- OUTPUT: M, where M is the index of search element

Sample Test Case
Input:
5
1 3 5 7 12
7
Output:
3

Problem 3: 6 marks

There are N students standing in a line. You are given the height for each student. For each student find the height of the nearest taller person to the right of the student. In case no such student exists, the answer is -1. Write a function in x86 assembly code for the problem.

Input/Output Format

- INPUT: Contains two lines. First line contains a single value N, size of the array. ($1 \leq N \leq 1000$). Second line contains N integers, the heights of N students.

- OUTPUT: Sequence of N integers, $b_1, b_2, \ldots, b_N$, where b_i ($1 \leq i \leq N$) is the height of the nearest taller person to the right of the ith student.

Sample Test Case

6
4 4 5 2 10 8
5 5 10 10 -1 -1

Explanation- For first element(i.e. height of first person in the line- 4) the next greater height is of 5. Similarly for third element it is 10 and for the fifth element there is no greater height in the remaining line, hence -1 is the output.

Problem 4: 2 marks

You are given 8 numbers as input. $x_1, x_2, x_3, x_4, x_5, x_6, x_7, y$. Your function in assembly must return

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 - y$$

8 numbers are passed as separate arguments to the function instead of passing one single array. Obtain 8 arguments correctly in your assembly program and return the desired result.

Input/Output Format

- INPUT: You will be given 8 integers in a single line. **-100 $\leq$ x, y $\leq$ 100**

- OUTPUT: A single integer which is the result of the above mentioned operation.

Sample Test Case

4 4 5 2 10 8 12 16
29

Explanation- Summation of first 7 numbers is 45. Subtracting 16 from 45 will give you 29

All the Best!