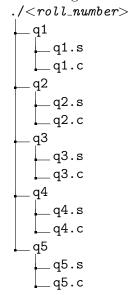# Assignment 1

Deadline: 23:55, 04/06/2022

Welcome to Assignment 1 of the Computer Systems Organisation Course. The aim of this assignment is to familiarize you with writing x86 code. On completion of this assignment you should be able to successfully write arithmetic, conditional, looping components, procedure calls, and conditional jumps in x86-64.

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

- There are 5 problems for this 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.

- The assignments will be manually evaluated for plagiarism. 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 25.

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.s
│   └── q4.c
└── q5
    ├── q5.s
    └── q5.c
```

- Zip the ./<roll_number> folder and name the zipped folder as <roll_number>_assign1.zip

Assume all the integer variables to be long long int. In case of error or invalid input return '-1'. Overflow cases to be handled (You can use remainder function).

Problem 1: 3 marks

Given two numbers M and N compute the quotient and remainder for M/N. Do not use inbuilt *div* or similar operations.

Input/Output Format

- INPUT: *M, N*

- OUTPUT: **q, r**; where q is the quotient and r is the remainder.

Sample Test Case
Input: 153 5
Output: 30 3

Problem 2: 4 marks

Given two numbers M and N calculate the greatest common divisor (GCD) of M and N.

Input/Output Format

- INPUT: N, M.

- OUTPUT: GCD(M, N)

Sample Test Case
Input: 24 39
Output: 3

Problem 3: 4 marks

Given a number N check if it is a prime number.

Input/Output Format

- INPUT : N

- OUTPUT: **TRUE**, if the given number is a prime number, **FALSE** otherwise.

Sample Test Case
Input : 13
Output: TRUE

Problem 4: 5 marks

Given a number N, find its largest prime factor.

Input/Output Format

- INPUT: N

- OUTPUT: *M*, Where M is the largest prime factor of N

Sample Test Case

Input: 24
Output: 3

Problem 5: 4 marks

Create a program to compute the square of the sum from 1 to n. Specifically, compute the square of integers from 1 to n and then sum the values.

Input/Output Format

- INPUT: A single integer number N

- OUTPUT: M, where M is the sum of squares of natural numbers till N

Sample Test Case

Input: 4
Output: 30