# CS6570 - Secure Systems Engineering: Assignment-3

## Submission guidelines

- **Deadline: 8th March 2025**
- We expect you to submit an arhive (tar or zip) that contains the following
    - The archive should be named as your RollNumber1_RollNumber2 (`<roll_no>.tar/zip`) when you upload it to teams. **(1 point)**
    - Files `solution_Q#` that contain the exploit strings you have crafted, and any scripts that were used to genarate the exploit strings.
    - The provided binary (unmodified)
    - A PDF report (preferably in LaTeX) that should contain the following things compulsarily:
        - Your Team-Name and Roll-Numbers.
        - The explanation of the ROPchains crafted for each question
        - The gadgets you have used
        - Pictures of your working exploits
        - Your individual contribution towards the submission.

## Files provided

- main
- This README

## Description

- The provided binary simply prints the following and waits for input

```
❯ ./rops
The Answer to Everything in Life is
========> 42
ARE U SATISFIED?

❯
```

- Upon normal execution, the binary answers question of life.
- Can you make it do something else by writing a ROPChain such that it computes the tasks that are provided to you.
- There are 2 tasks given to you based on this binary.

## Tasks

1. The binary computes the answer to life, can you exploit the program to compute 12th fibonacci number?

   a) Describe the vulnerability in the binary, how you exploited it and can it be fixed? (10 points)

   b) Upon execution of the payload that you created the program should PRINT `144`. (29 points)
2. Craft a ROPchain to calculate the `N`th number in the fibonacci sequence? The value of `N` should be from as STDIN using functions like `scanf()` (60 points)

**BONUS**: If your ROPChain can compute the `N`th Factorial where `N` is taken from as STDIN using functions like `scanf()`

## Testing

- Ensure that your exploit string is self-contained in `solution_Q#`, any additional steps or modifications are not allowed.
- Your input will be passed to the program in the following manner:

```
> cat solution_Q1 - | ./rops >&1
```

## General Guidelines

- You are expected to write ROPchains to compute the required expressions. Solutions such as simply printing 144 in Task-1 would not be accepted
- ROPgadget is a tool installed on the VM to identify gadgets that you can use in your ROPchain, you are free to use other tools
- Please ensure that your `solution_Q#` works on the provided course VM.
- You can write scripts to generate `payload` , you must include these in your archive.