# Basic reverse engineering

## Introduction

In this assignment, you will start basic reverse engineering. You are given 2 binaries which you have to analyze, determine what they do and reimplement the same functionality in a higher level language. Also, include a writeup(in a text file) describing how you reverse engineered the binary to find out what it does.

## Question 1: Reverse me 1

Analyze the binary 1-reverseme.out, understand what it does and implement the same in a higher level programming language(i.e. other than assembly programming language).

## How we will test your program

You have to provide two files along with your program:

1. **compile.sh**: A shell script that will compile your program to generate the executable. No arguments will be passed to this shell script when it is executed. This file is optional and required if you implement the program in a programming language that requires compilation such as C, C++, Java etc.
2. **run.sh**: A shell script that will execute your program with the arguments passed to the shell script. **This file is required: your solution will not be graded if this file is missing and a score of 0 will be awarded.**

We will invoke the above two shell scripts as follows:

```
$ sh ./compile.sh
$ sh ./run.sh <arguments>
Output for each argument: 1 per line
```

## Question 2: Reverse me 2

Analyze the binary 2-reverseme.out, understand what it does and implement the same in a higher level programming language(i.e. other than assembly programming language).

## How we will test your program

You have to provide two files along with your program:

1. **compile.sh**: A shell script that will compile your program to generate the executable. No arguments will be passed to this shell script when it is executed. This file is optional and required if you implement the program in a programming language that requires compilation such as C, C++, Java etc.
2. **run.sh**: A shell script that will execute your program with the arguments passed to the shell script. **This file is required: your solution will not be graded if this file is missing and a score of 0 will be awarded.**

We will invoke the above two shell scripts as follows:

```
$ sh ./compile.sh
$ sh ./run.sh <arguments>
Output for each argument: 1 per line
```