Periodical exam 1

There are 3 questions in this exam worth 5 marks each. All questions are mandatory. You are allowed to refer to classroom material but use of internet is not allowed and considered cheating. All other regular examination related rules are applicable here as well. The decision taken by the instructor in any matter related to the exam will be final and binding. The solution will be evaluated on Ubuntu 14.04. The total duration of exam is 2 hours.

Question 1

Write an assembly program that accepts N integers from the command line and prints out the maximum among them. You can assume that at least two integers will be provided and they will fit within 64 bits. Use of standard C library functions is permitted. You cannot invoke any additional code besides the assembly code you write and the standard C library functions.

Sample session

```
$ ./find-max.out 1 3 2
3
```

How we will test your program

We will compile the assembly code you submit using nasm and gcc. The exact commands that will be executed are:

```
$ nasm -f elf64 find-max.asm
$ gcc find-max.o -o find-max.out
```

We will pass arguments to the program as described in the question.

Question 2

Write an assembly program that greets the user(whose name is provided as a command line argument) by printing out “Hello <name>!”. If no name is provided, the program should greet the entire world as “Hello world!”. **Use of standard C library is not permitted**. You cannot invoke any additional code besides the assembly code you write and the kernel system calls.

Sample session

```
$ ./greet-user.out IntroSSOC
Hello IntroSSOC!
$ ./greet-user.out
```

Hello world!

How we will test your program

We will compile the assembly code you submit using nasm and gcc. The exact commands that will be executed are:

```
$ nasm -f elf greet-user.asm
$ gcc -nostdlib greet-user.o -o greet-user.out
```

We will pass arguments to the program as described in the question.

Question 3

Analyze the function “function1” in the given binary and reimplement the same functionality in higher level programming language. You are allowed to use any programming language in your solution except assembly programming language. Please include a run.sh file which will compile and/or execute your solution by passing the arguments appropriately.

Note: You need to reimplement the same functionality as function1 only. Your implementation need not work the same way as function1: you can choose any approach. The only requirement is that the return value printed to stdout should be correct.

Sample session

This sample session only demonstrates the input and expected output format. The exact output for specific inputs are not displayed.

```
$ ./run.sh ABCD EFGH IJKL
Return value of function(ABCD)
Return value of function(EFGH)
Return value of function(IJKL)
```

How we will test your program

We will invoke the run.sh file you upload using sh. We test your solution on Ubuntu 14.04, which symlinks sh to dash. If you use a different *nix OS, please ensure that your run.sh file is compatible with dash. For most simple commands, dash is mostly a drop-in replacement for sh but if you run some fancy commands or use shell specific tricks in run.sh, expect issues. We will pass 1 or more arguments and your solution should print the return value of function1 when applied to each command line argument on a separate line, as shown in the sample session.