COP3275 – Spring 2019

Instructor: Roozbeh Ketabi

Programming Assignment 1

1- You have to upload your codes in both Canvas and the Judge system at https://cop3275.cise.ufl.edu
2- For guide on how to access the Judge, visit Judge Access Page under Pages section in Canvas.
3- The judge will run test cases against your code and assign it a grade. You can have multiple submission for the same problem but you have to choose which one is the final (we will consider your final submission for grading). Canvas is only used for record keeping.
4- When upload on canvas, zip your programs in a zipfile with your ufid as name (nothing more, just 8 digit ufid, for instance 12345678.zip). Name your files (inside the zipped archive) p1.c, p2.c, and so on.
5- For all the problems input is read from standard input (i.e. read using scanf) and must be written to standard output (i.e. printf).
6- **Don't print extra stuff. Since the assignments are automatically graded, if the output doesn't match the expected output, you will not get the points. For instance:**
If the problem is to read a number and return its square, the expected output is a number (i.e. "printf("%d",i*i)"). If you print using something like "printf("square is %d",i*i);" you will not receive any points.
7- The assignment is due on 11:59 pm Monday Feb 4th 2019. There is a 20% penalty for late submission of one day. No submission is accepted beyond that time.

Problem 1: Accumulator Machine

In this problem we try to mimic a simplified accumulator architecture. An accumulator machine is a type of 1-operand machine where it uses an "accumulator" to store the results. For example, the add operation would be something like "add R1" which means add the value of R1 to the accumulator, so if accumulator was 0 now it will be equal to R1. If another command such as "add R2" is issued, then R2 will be added to accumulator which has the value of R1 so that the new value of accumulator is R1 + R2.

You will be given the input command (operator) as a number between 0 and 9, followed by the number (the operand) in each line of stdin (read with scanf). Assume accumulator starts at 0. EXIT, INC, DEC, PRINT and RESET don't have an operand. If the command is Print then you print the value of the accumulator with a newline, if it is RESET, the value of accumulator is reset to 0. If it is EXIT you exit the program. INC would increment the accumulator by 1 whereas DEC would decrement it by one.

In cases of division and modulus, the remainder must be always positive. You must make sure both division and modulus results are consistent.

0	1	2	3	4	5	6	7	8	9
EXIT	ADD	INC	SUB	DEC	MULT	DIV	MOD	PRINT	RESET

Sample:

Input:

```
1 50
2
5 3
8
7 100
8
9
1 100
3 10
8
0
```

Output:

```
153
53
90
```

Problem 2: Fibonacci Number

A number in the Fibonacci sequence is the one that is the sum of the two preceding numbers in the sequence. For instance, if the first number is 5 and the second is 3 the third would be 8, fourth would be 11 and so on. For this problem you have to write a program where it takes in the first two numbers of the sequence and another number N in a newline where you have to compute and print the Nth element of the sequence.

Sample
Input1:
```
5 10
9
```
Output1:
```
275
```

Input2:
```
1 1
12
```
Output2:
```
144
```

Problem 3: Sacred number!

A number is sacred if it has an even number of zeroes and is divisible by 13 or it has odd number of zeroes and is divisible by 7. Write a program to test whether a given number is sacred or not. Print 1 if it is sacred and 0 otherwise.
If you just printf 1 or 0 for the output in hope of getting 50% (without implementing any logic) you will not receive any points for the problem.

Samples:
Input1:
3003
Output1:
1

Input2:
5810
Output2:
1

Input3:
6320
Output3:
0

Input4:
6323
Output4:
0

Input5:
9893
Output5:
1