COP 2200 Introduction to C
Homework 5

Feng-Hao Liu

In this assignment, you are going to turn in **a single** cpp file. Your goal is to compute modular exponentiation as we will explain below. Then you are going to practice how to design a function that needs to output "two" things.

Task 1: You need to design a function that takes in inputs three integers a, e, and n. The function returns an integer which is supposed to be $(a^e \mod n)$. Basically, a^e is a to the power of e, and "mod" means the modulo operation. Recall in the midterm exam, the operation is %, i.e. $a \mod b$ can be written as $a\%b$ in C/C++ code. The outcome is the remainder of a divided by b.

One nice property about modulo computation is the following:

$$(a+b) \mod n = (a \mod n) + (b \mod n) = ((a \mod n) + (b \mod n)) \mod n,$$

$$(a*b) \mod n = ((a \mod n) * (b \mod n)) \mod n.$$

This property allows you to compute $a^e \mod n$ precisely without worrying about overflow. You should note that if you first compute a^e directly as an integer, then it is very likely that the result will be larger than what an int can represent. This is typically called as the problem of overflow. On the other hand, if you always apply mod n after one multiplication, then you can bring the number back to the range of $[0, n]$. This avoids the problem of overflow.

Task 2: We want to design a function that takes inputs of 5 integers, a, x, b, y, n, and the function computes $a^x \mod n$ and $b^y \mod n$. However, as we discussed in the class, a function cannot return two integers. To achieve the same effect, we need to apply the technique of pointers we learned in the class.

So the task becomes: design a function a function that takes inputs of 5 integers, a, x, b, y, n, and two integer pointers $*out1$ and $*out2$. The function returns void, but needs to write a^x to the place that $out1$ points to and similarly b^y to the place that $out2$ points to. This has the same effect as returning two outputs.

Task 3: In the main function, the program asks the user to cin 5 numbers a, x, b, y, n. Then the program calls the function in Task2 to figure out the answer of a^x and b^y. Then it prints the answers. You can design whatever interface you like as long as it fits the spirit! A sample interface looks like the following.

- Give me 5 numbers for a, x, b, y, n.
- 2 2 3 3 5 (This line is input by the user.)
- The answers are 4 and 2.