

Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth

ECE160: Foundations of Computer Engineering I (Spring 2023)
Instructor: Dr. Liudong Xing

LAB # 8

(Relevant Lecture: #16-17)

Monday, March 20 (L1) and Wednesday, March 22 (L2)

OBJECTIVES

- To understand the difference between two ways of parameter passing for functions (L#16)
- To learn the random number generation functions (L#17)
- To learn the recursive functions (L#17)

SUBMISSION REQUIREMENT

1. Please follow “[Submission Guidelines](#)” in the lab section of the course website to submit your solution (program files) to the class M: drive by **5pm, Wednesday, March 22**.
2. Suggested format for naming your solution files: [lab#-your last name-p#.cpp](#)
For example: [lab8-xing-p1.cpp](#) for problem 1; [lab8-xing-p2.cpp](#) for problem 2; ...

EXERCISES

1. The program below contains some semantics/logic error. Please identify the error, fix the error and run the program. Note that there is no syntax error in the program (i.e., the program can be built successfully). *Hint: the error is related to the parameter passing* (refer to [Lecture#16 Slides 5-10 if necessary](#)).

The program is supposed to code a number by extracting the least significant digit (LSD) of the number using the modulo % operation and then adding this digit to the number.

Example (Test Case): if you input the original number 1254, the coded number will be 1258, which is obtained by adding the LSD 4 to 1254. The expected correct output from the program should be:

The value of a before the function call is: 1254

The value of a after the coding is: **1258**

You may use the debugging functions in Visual Studio you have learned from [Lab#7](#) to help you identify the error if necessary.

```
#include <stdio.h>

void code(int m);

void main(void)
{
    int a;
    printf("Enter the number to be coded\n");
    scanf_s("%d", &a);
```

```

    printf("The value of a before the function call is: %d\n", a);
    code(a);
    printf("The value of a after the coding is: %d\n", a);
}

void code(int m)
{
    int lsd;

    lsd = m % 10;
    m = m + lsd;
}

```

2. The program below generates a random number and prints it out. However, the same random number will be obtained each time you run the program. Please run the program twice to confirm this. Modify the program so that the program generates a different random number each time you run the program.

```

#include "stdio.h"
#include "stdlib.h"

void main(void)
{
    int rand1;

    srand(997);

    rand1 = rand();
    printf("The number is %d\n", rand1);
}

```

3. Modify the program in the last exercise so that the program generates a different random number **in the range of 30 ~ 90** each time you run the program. Run your program at least twice to test your program.
4. Write a program that can do the following tasks:
- 1) input two integer numbers from the keyboard.
 - 2) compute the greatest common divisor (**gcd**) of these two integers using the Euclidean algorithm as described in the equation below. Note that remainder(x,y) can be evaluated using the modulo operator %.
- $$\text{gcd}(x, y) = \begin{cases} x & \text{if } y = 0 \\ \text{gcd}(y, \text{remainder}(x, y)) & \text{if } y > 0 \end{cases}$$
- 3) print the **gcd** out.

You are required to compute the greatest common divisor in a **user-defined recursive function** and call the function in the **main()**.

Test Cases:

```

gcd(27, 9)=9
gcd(111, 259)=37
gcd(8, 6)=2

```