HW #5: Iterations

Due dates:

Part I: Monday Feb 15th, at the beginning of the class. Make sure to write your name and msunetid on your paper.
Part II: Sunday Feb 14th, 11:59 pm through Handin (https://secure.cse.msu.edu/handin)

Part I: Comprehension Questions

1. What is the output of the following program fragments? (3pts)

a. Program 1

```
int alpha = 500;
do {
    alpha /= 5;
    printf(“%d “, alpha);
} while (alpha > 2);
```

100 20 4 0

b. Program 2

```
int a = 0;
while (a < 11) {
   if (a % 3 == 0)
       continue;
   if (a % 5 == 0) {
       printf(“** %d **\n”, a);
    } else {
       printf(“%d\n”, a);
    }
    a++;
}
```

The program will be stuck in an infinite loop printing nothing.

Since a is 0, a < 11, so the program will enter the loop.

a%3 is 0, so the program will execute the first if statement, which tells the program to skip the remainder of the loop and go back to beginning of the loop (ie checking the condition again). Variable a does not change and the process is repeated indefinitely.

c. Program 3

```
int i = 5, j = 9;
for (i = 0; i < 5; i++) {
      for (j = 0; j < 3; j++)
            printf(“$”);
      printf(“\n”);
}
```

```
      $$$
      $$$
      $$$
      $$$
      $$$
```

The first loop is executed 5 times. Each time, the inner loop is executed followed by printing a newline. The inner loop prints 3 $ symbols. So the whole program will output five rows, each row consisting of $$$

2. Convert the following for loop into a **`for (;;) {   }`** loop form (2 pts)

```
for (idx = 5; idx<100; idx+=2) {
      printf(“Iteration #%d\n”, idx);
```

```
//Initialize idx before the start of the loop
idx = 5;
for (;;) {
      //Check the condition and exit if not valid
      if (idx >= 100)
            break;

      //execute the loop body
      printf(“Iteration #%d\n”, idx);

      //execute the increment statement last
      idx++;
}
```

3. What is the value of x and y in each of the following cases, after the execution of both statements? (2pts)

Program 1:

```
int x = 0, y = 0;
y = x = 11, 12, 13, 14;
```

x = 11

y = 11

The statement above is equivalent to: (y = (x = 11)), 12, 13, 14;

Program 2:

```
int x = 0, y = 0;
y = x = (11, 12, 13, 14);
```

x = 14

y = 14

The statement above is equivalent to: y = (x = (11, 12, 13, 14));

4. Write a program that outputs the multiplication table from 0 to 10 as follows: (5 pts)

```
0x0=0     0x1=0   0x2=0   …    …    0x10=0
1x0=0     1x1=1   1x2=2   …    …    1x10=10
2x0=0     2x1=2   2x2=4   …    …    2x10=20
…      …              …    …
9x0=0     9x1=9   9x2=18 …    …    9x10=90
10x0=0  10x1=10      10x2=20      … 10x10=100
```

```
#include<stdio.h>
int main(void) {
    int row, col;
    for (row = 0, row<=10; row++) {
        for(col =0; col<=10; col++) {
            printf("%dx%d=%d ", row, col, row*col);
        }
        printf("\n");

    return 0;
}
```

Part II: Lab Assignment

Getting started

Change into the cse220 directory
Create a new directory called lab05
Change into the new directory
Implement the program below in your lab05 directory

Euler's number

The number *e* is a mathematical constant approximated at 2.718.
The formula for computing it is as follows:

```
e = 1 + 1/1! + 1/2! + 1/3! + 1/4! + ...
```

Write a program Euler.c that approximates the value of e by using the following equation:

```
e = 1 + 1/1! + 1/2! + 1/3! + 1/4! + ... + 1/n!
```

where n is a number entered by the user.
Your program will first compute e as 1, then as 1 + 1/1!, then as 1 + 1/1! + 1/2! and so on. In each step, print the step number and the value of e, as follows (with 20 digits after the decimal point):

```
1:      2.00000000000000000000
2:      2.25000000000000000000
3:      ....
```

After how many steps the value does not change?
Change the type of e from float to double. Does your answer change?
Remove the intermediate print statements and only print the value once at the end of your program.
Compile your program into an executable called Euler.

Interest Rates

Write a program InterestRates.c that computes the interest amount on multiple loans. The program should ask the user to enter the loan amount and the interest rate. It should then output the interest amount with two digits after the decimal. The program should terminate when the loan amount entered is 0 as in the following example *(underlined numbers represent the user input)*:

```
Enter loan amount ($): 15000
Enter interest rate (%): 6.25
*** Interest amount ($): 937.50

Enter loan amount ($): 135
Enter interest rate (%): 21
*** Interest amount ($): 28.35

Enter loan amount ($): 0
```

Compile your program into an executable file called InterestRates.

Handin

Submit through the handin system your C code and the executables generated.

The “handin” system has options to allow you to review your files online and to download them. You Should always verify that you submitted the correct files and they were received by the handin system. You can submit files as many times as you like for a particular assignment. Handin will only keep the last version of each file. Remember to submit your files prior to the deadline as you won’t be able to use handin if the deadline has passed.