Ron Paolo Q. Molejona - 202151010
CMSC 21 First Long Examination April 22, 2022

Part I. True or False

1. True, because long double is 10 bytes while double is 8 bytes. [1]
2. True, usually, 0 is used as a false value, and 1 as a true value.
3. False, == is used for comparison while = is used for assigning values
4. True, it is on the rules of creating variables
5. True, it is also called as pointer.
6. False, it can only be used on int and char types. [2]
7. False, (a==b) false; (b>a) && (d<a) = true; false || true = true
8. False, it is not required if the default case is in last.
9. False, both x>y or a<b needs to be true
10. True, they produce similar output

Part II. Find the errors in the following program. Indicate the possible correction.

1.
```
int x = 1;
while (x <= 10)
{
  x++;
}
```

2.
```
for (double y = 1; y == 1; y += 1)
{
  printf("%f\n", y);
}
```

3.
```
switch (n)
{
case 1: printf("The number is 1");
break;
case 2: printf ("The number is 2");
break;
default:
printf ("The number is not 1 or 2");
}
```

4. int n = 1;
   while (n <= 10)
   {
     printf("%d ", n++);
   }

Part III. Answer the following questions.

1. Accessing the value of an uninitialized variable will result in strange errors.
   int x;
   printf("%d", x); result: -494368328
   2nd result: 1019296776

2. Nothing will happen if you don't use the return function at the end of the main function. return 0; is only to indicate if the exit of a program is successful.

3. The only difference is that %d is for specifying decimals while %i is for the type integer.

4. The values are: a = 10, b = 5, c = 5.000000

5. The values are: a = 12.300000, b = 789, c = 45

6. A. ((a * b) - (c * d)) + e
   B. ((a / b) % c) / d
   C. (((- a - b) + c)-)+ d
   D. ((a * - b) / c) – d

7. for (int i = 0; i > 0; i++)
   {

   }

Part IV. Coding Applications

Github Link:
https://github.com/Ron-Paolo-Molejona/CMSC21/tree/main/First%20Long%20Exam

8. A. The issue with the code is the lack of brackets, usage of indention for readability.

   B. Code on Github (Part 4_Num 8_ a.c ; Part 4_Num 8_ b.c ; Part 4_Num 8_ c.c)

9. Code on Github (Part 4_Num 9.c)

   Sample Output:

```
Enter square size:5
*****
*   *
*   *
*   *
*****
Print another square? Enter y or n: k
Not a valid choice.
Print another square? Enter y/n: y
Enter square size:4
****
*  *
*  *
****
Print another square? Enter y or n: n
END
```

10. Code on Github (Part 4_Num 10.c)

    Sample Output:

```
Enter x: 3

The squareroot of 3 is: 1.732051
```

Sources:

[1] Floating Point Types
https://www.tutorialspoint.com/cprogramming/c_data_types.htm

[2] signed and unsigned qualifiers
https://overiq.com/c-programming-101/data-types-in-c/#:~:text=signed%20and%20unsigned%20qualifiers&text=The%20range%20of%20values%20of,with%20int%20and%20char%20types.