MARK VINCENT LIMSON **202150328**
CMSC 21 – LECTURE **FIRST LONG EXAM**

I. TRUE OR FALSE. JUSTIFY THE ANSWERS.

1. **True** – double data type has limit on the range it can hold, thus can be substituted by long double in holding higher ranges.

2. **True** – in Boolean algebra, a zero (0) would indicate a false return while one (1) would indicate a true return. Basing it from the branching concepts of C programming, a zero (0) is treated the same way as it is in Boolean, but any non-zero number is treated as true.

3. **False** – single equal (=) sign is for assignment, while paired equal (==) sign is used for comparison.

4. **False** – there is not much restriction on variable names and can be considered good for as long as it does not start with a number or contain any special characters aside from underscore (_).

5. **True** - & operator is utilized in c programming for storing data value on a specific variable, this is usually seen in user inputs after declaring the data type for a specific variable.

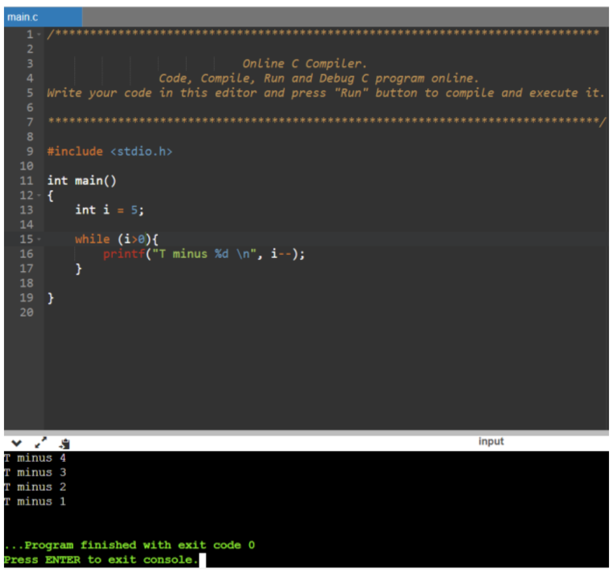
6. **False** – sign qualifiers can only be used in int and cha data types.

7. **False** – analyzing the expression logically or by running a code will give you a true return, because both conditions were satisfied ([1]a is equal to b or b is greater than a; [2]d is less than a).

8. **False** – the default case is used when no above cases were met or true, hence no longer needs to use a break statement on it.

9. **False** – the logical operator used (&&) indicates an 'and' which means that both conditions have to be met, otherwise the function will have a false return.

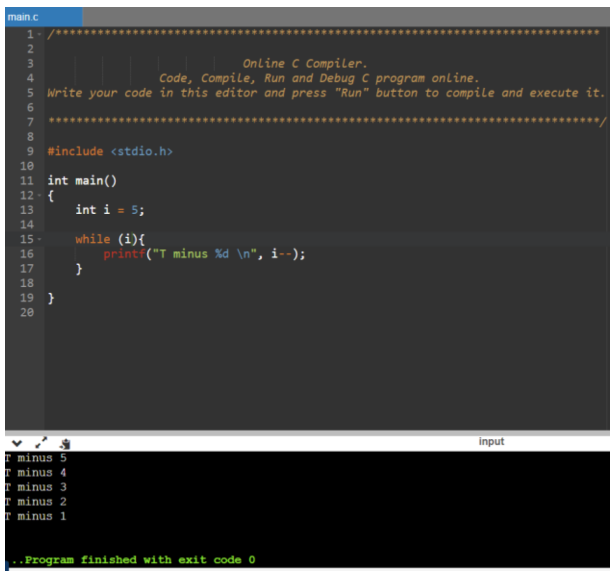
10. **True –**



main.c

```c
/******************************************************************************

                            Online C Compiler.
                Code, Compile, Run and Debug C program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/

#include <stdio.h>

int main()
{
    int i = 5;

    while (i>0){
        printf("T minus %d \n", i--);
    }

}

```

```
T minus 4
T minus 3
T minus 2
T minus 1


...Program finished with exit code 0
Press ENTER to exit console.
```



main.c

```c
/******************************************************************************

                            Online C Compiler.
                Code, Compile, Run and Debug C program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/

#include <stdio.h>

int main()
{
    int i = 5;

    while (i){
        printf("T minus %d \n", i--);
    }

}

```

```
T minus 5
T minus 4
T minus 3
T minus 2
T minus 1


..Program finished with exit code 0
```

MARK VINCENT LIMSON **202150328**
CMSC 21 – LECTURE **FIRST LONG EXAM**

II. FINDING ERRORS IN THE GIVEN PROGRAM. INDICATE CORRECTIONS.

1. **Errors**: ^{1}Data type of x undeclared, ^{2}no opening bracket on while loop, ^{3}incrementation improperly executed.

Possible correction:

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

2. **Errors**: ^{1}data specifier used incorrectly

Possible correction:

```
for (double y = .1; y != 1.0; y += .1) {
 printf("%lf\n", y); } //used %lf instead of %f
```

3. **Errors**: ^{1}Case 1 has no break, ^{2}break in default is no loner necessary.

Possible correction:

```
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");
}
```

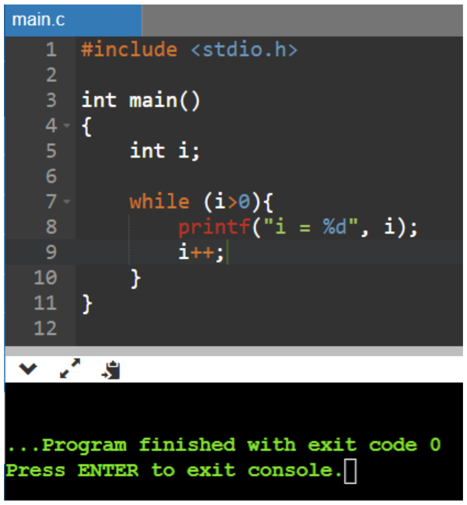
4. **Errors**: ^{1}data type for variable n is undeclared, ^{2}operator excludes 10.

Possible correction:

```
int n = 1;
while (<=10){
        printf(“%d”, n++);
}
```

III. ANSWER THE FOLLOWING QUESTIONS

1. Uninitialized variables can sometimes make the code not work as anticipated, though it would not cause any syntax error. One example is listed below:



```c
#include <stdio.h>

int main()
{
    int i;

    while (i>0){
        printf("i = %d", i);
        i++;
    }
}

```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

2. Technically, return values at the end of the program indicates exit from the main function. However, C has the capability to return to its main function without the return statement and end the program. Basically, there's not much influence on the whole program if a return statement is not indicated in the end.

3. There is not much difference between %i and %d, both can function as specifiers for integers. However, %i is more specific with signed decimal integers.

4. a = 10, b = 5, c = 0.30000

5. a = 12.3000, b = 0.6, c = 45

6. **Listed below:**

6a. (a*b) – (c*d) + e

6b. (a/b) % (c/d)

6c. (-a – b) + (c-d)

6d. (a * (-b/c)) – d

7.

MARK VINCENT LIMSON **202150328**
CMSC 21 – LECTURE **FIRST LONG EXAM**

IV. CODING APPLICATIONS

8a. The output of the code given that a = 2 and b =3 will be "*****", However, the provided code will give an error because the way the conditional is nested is not properly executed.

8b:

a. https://github.com/mackkk-n/CMSC-21-Lecture-/blob/master/Long%20Exam%201/LE1_Code1a.c

b. https://github.com/mackkk-n/CMSC-21-Lecture-/blob/master/Long%20Exam%201/LE1_Code1b.c

c. https://github.com/mackkk-n/CMSC-21-Lecture-/blob/master/Long%20Exam%201/LE1_Code1c.c

9. https://github.com/mackkk-n/CMSC-21-Lecture-/blob/master/Long%20Exam%201/LE1_Code2.c

10. https://github.com/mackkk-n/CMSC-21-Lecture-/blob/master/Long%20Exam%201/LE1_Code3.c