Lab 1: After Environment Setup; test your programming basics

Note: ***Students must demonstrate their lab in the following week of September 28th, 2020 to get the grade. This lab is worth 20marks. A Missed demonstration will result in 25% grade deduction. Late submission and late demonstration to the lab is welcomed but with a loss of marks. That is, you do not earn full marks for late submission. 10% penalty for every day late up to 50%. Work will not be accepted after 10 days.***
You must talk to your lab professor in case of any unprecedented situation which may result in late submission.

Program #1:

Write a small program: {(00_Numbers.c)} that:-
1. Prints the numbers from 1 to 100
2. If the number is a multiple of three, it should print instead "`I'm a multiple of 3!`"
3. If the number is a multiple of five, it should print instead "`I'm a multiple of 5!`"
4. If the number is a multiple of three and five, it should print instead "`I'm a multiple of 3 && 5!`"

This program should not take you more than 10 minutes to write.

The following demonstrates the execution of the program:

```
#./00_Numbers
...
8
9 I'm multiple of 3!!!
10 I'm multiple of 5!!!
11
12 I'm multiple of 3!!!
13
14
15 I'm multiple of 3 && 5!!!
                    Excerpt SAMPLE TEST OUTPUT:  00_Numbers
```

Program #2:

Write a small C program: `digits.c` that:
1. Read an integer number from the command line using `scanf( )`
2. Find the number of digits in the number.
3. Prints the number of digits to the screen.
4. Finish with a value of `0`.

The following demonstrates the execution of the program:

```
root@bahris:01_Lab# ./digits
Enter an integer:  8765
Number of digits in 8765 is 4
root@bahris:01_Lab# ./digits
Enter an integer:  -98765
Number of digits in -98765 is 5
root@bahris:01_Lab# ./digits
Enter an integer:  123456789
Number of digits in 123456789 is 9
# echo $?
0
                          SAMPLE TEST OUTPUT:  digits
```

Program #3:

Write a small C program: `reverse.c` that:

5. Read an integer number from the command line using `scanf( )`.
6. Prints the number with its digits reversed to the screen.
7. Finish with a value of `0`.

```
root@bahris:01_Lab# ./reverse
Enter an integer:  7654
The reversed number is 4567
root@bahris:01_Lab# ./reverse
Enter an integer:  78
The reversed number is 87
root@bahris:01_Lab# ./reverse
Enter an integer:  9870
The reversed number is 789
                          SAMPLE TEST OUTPUT:  reverse
```

Program #4:

Write a small C program: `bin2dec.c` that:

1. Read a binary number (just `0` and `1`) from the command line using `scanf( )`.
8. Prints the decimal number equivalent to the binary number entered.
9. Finish with a value of `0`.

```
root@bahris:01_Lab# ./bin2dec
Enter a binary number: 1100
The decimal equivalent of 1100 is 12
root@bahris:01_Lab# ./bin2dec
Enter a binary number: 11101
The decimal equivalent of 11101 is 29
root@bahris:01_Lab# ./bin2dec
Enter a binary number: 10
The decimal equivalent of 10 is 2
                          SAMPLE TEST OUTPUT:  bin2dec
```