Benz Vrianne Beleber 202105039
CMSC 21 Lec-3

Operators in C
Lecture 2 Assignments

1. Code the following:
a. Prompt the user to enter a two-digit number
b. Display the number with the digits reversed
Example:

```
Please enter a 2-digit number: 75
Reverse: 57
```

Save your code as `as1.c`

CMSC 21 > CMSC 21 > Lecture2 > C as1.c > main(void)

```c
//reversing the order of a 2-digit number
#include <stdio.h>
    int main (void) {
        int num, ones;

        printf ("Please enter a 2-digit number:\n");
        scanf ("%d", &num);

        ones = num%10;                              //to isolate the ones digit
        num = (num/10)+(ones*10);                   //to move the tens digit to the ones place
                                                    //then added by the ones digit multiplied by 10 so it becomes the tens digit.

        printf("Reverse: %d", num);

        return 0;

    }

```

2. Extend the code in item 1, such that it reverses a 3-digit number.
Example:

```
Please enter a 3-digit number: 123
Reverse: 321
```

Save your code as `as2.c`

CMSC 21 > CMSC 21 > Lecture2 > C as2.c > main(void)

```c
//reversing the order of a 3-digit number
#include <stdio.h>
    int main (void) {
        int num, ones, tens;

        printf ("Please enter a 3-digit number:\n");
        scanf ("%d", &num);

        ones = num%10;                              //isolating the ones digit
        num = (num/10);                             //updating the user given to proceed
        tens = num%10;                              //isolating the tens digit
        num = (num/10)+(tens*10)+(ones*100);        //the last division of 10 leaves the original hundreds digit
                                                    //to be a single digit moving it to ones place
        /*the tens and ones digit will then be multiplied to the right x10 multiplier to put them in their new respective places*/

        printf("Reverse: %d", num);

        return 0;

    }

```

Benz Vrianne Beleber 202105039
CMSC 21 Lec-3

3. Provide the output of the following codes, given that i, j, and k are integer variables.

```
a) i = 3; j = 4; k = 5;
printf("%d", i < j || ++j < k);
```

output:**1**

```
b) i = 7; j = 8; k = 9;
printf("%d",i - 7 && j++ < k);
```

output:**0**

```
c) i = 7; j = 8; k = 9;
printf("%d", (i = j) || (j == k));
printf("%d %d %d", i, j, k);
```

output:**1 8 8 9**

```
d) i = j = k = 1;
printf("%d", ++i || ++j && ++k);
printf("%d %d %d", i, j, k);
```

output:**1 2 1 1**