Lesson 11 Practice 3

Here are some examples of For & if commands

```
if(condition){
//the code block to be executed when the condition is true
}
```

In absence of brackets **{ }** only the first statement is considered part of the if-else block.

```
int x = 5;
if (x == 1);
        printf("Hello");
```

Here, the print statement will be executed as there is a semicolon after the if statement and thus, it will be considered in scope.

What does x==5 mean?

Compare the value of x to the number 5

What about x=5?

To assign the variable x a value of 5

```
If (number)
    printf("true");
```

Any integer other than 0 is true, hence the print statement will be executed.

Only one statement is executed by the for loop when there aren't curly braces { }.

```
for ( ;  ; ) {
//target statement(s)
}
```

```
for ( ;  ; );
        printf("Hello");
```

The print statement will be executed as there is a semicolon after the for statement.

```
int i = 0;
for (; i; i++)
        printf("Hello");
```

The print statement won’t be executed as the for command functions in the following order:

```
for (1; 2; 4) {
// 3
}
```

1: initialization step: it is executed first, and only once.

2: a conditional expression. It checks for a specific condition to be satisfied. If it is not, the loop is

terminated.

3: the body of the loop is executed.

4: increment or decrement to update the value of

the loop variable.

```
int i = 1;
for( ; i ; i++)
        printf("Hello");
```

The program will repeat **indefinitely** as i != 0 and thus, the condition is always true.

We can declare **multiple variables** in the initialization part and do **multiple operations** in the increment/decrement part.

Just separate the multiple operations/variables with **commas**:

```
int i = 5, j = 1;
for (; i; i--, j--)
        printf("%d %d \n ", i, j);
---------
int i, j;
for (i = 5, j = 1; i; i--, j--)
        printf("%d %d \n", i, j);
```

output :

```
5 1
4 0
```

```
3 -1
2 -2
1 -3
```

Multiple test conditions can be used, but they cannot be separated by commas**. AND operator (&&)** can be used to connect them. It evaluates two conditions and returns true only when both conditions are true. **OR operator (||)** can also be used. This means that if one or both of the conditions are true, we get a value of true returned to us.

true && true = true

true && false = false

```
int i, j;
for (i = 5, j = 1; i > 0 && j < 6; i--, j--)
        printf("%d %d \n ", i, j);
```

output :

```
5 1
 4 0
 3 -1
 2 -2
 1 -3
```

```
int i, j;
for (i = 5, j = 1; i > 0 || j < 6; i--, j++)
        printf("%d %d \n ", i, j);
```

output :

```
5 1
 4 2
 3 3
 2 4
 1 5
```

The body of the loop will be executed if one or both of the conditions are true.

Try to code yourself:

- - > click here: Lesson 11 Practice 3 c1 - Replit