CS1160
Lab 5: Nested Loops 2

In many cases we may use loop statement inside another looping statement. This type of looping is called nested looping.

Types of nested loop:

- Shapes and Main diagonal
- Working With Equations
- Factorial Number
- Prime Number
- Reverse of the Given Number Using Loops
- Palindrome number

NESTED FOR LOOP:

```
SYNTAX:
for (initialization; condition; increment/decrement)
{
for (initialization; condition; increment/decrement)
{
 body of the loop;
 }
}
```

The inner loop runs as many times as there is the limit of the condition of the external loop. This loop runs as long as the condition in the parenthesis is true.

Example:

#include<stdio.h>
int main() {
  int i, j;
for (i =1; i <=4 ; i++){
  for (j =1 ; j<= 4 ; j ++)
  printf("*");
  printf("\n");
} return 0;
}
	Output:
	
	***
	***
	***
	***
		Main Template For Shapes
		for (int i =1; i <=rows ; i++)
		{
		  for (int j =1 ; j<= column ; j ++)
		  printf(" * ");
		  printf(" \n ");
		}

Sara Abu Sa'aleek

- **Main diagonal:**

```
for (int i = 1; i <= rows; i++)   {
     for (int j = 1; j <= column; j++)    {
    if (i < j )
      printf ("*");   // above diagonal
   else if (i >  j )
     printf ("1"); // under diagonal
   else if (i == j )
      printf ("#"); // second diagonal
        }
printf ("\n");
}
```

Example:

```
#include<stdio.h>
int main() {
    int i,j;
     for (int i = 1; i <= 5; i++)    {
       for (int j = 1; j <= 5; j++)     {
      if (i < j )
        printf ("*");   // above diagonal
     else if (i >  j )
       printf ("1"); // under diagonal
     else if (i == j )
        printf ("#"); // second diagonal
           }
printf ("\n");
}

    return 0;
}
```

Output:

```
#****
1#***
11#**
111#*
1111#
```

Working with equation:

Example:

```
// conditions
```

$$\sum_{i=1}^{i=100} \sum_{j=1}^{j=i} j^2 * i^2$$

```
// inside loop
// initialize variables
```

```
double sum = 0;
 for (int i = 1; i <= 100; i++)
   for (int j = 1; j <= i; j++)
     sum = sum + ((j * j) * (i * i));
printf ("%lf \n", sum);
```

Factorial Number using do while

- Write a C program to read an integer number **num** and find it's factorial.
- **Note:** the factorial number is all the number multiplied from 1 to the number itself.

$$n! = n(n-1)(n-2)\dots(2)(1)$$

- Example :

5! = (5)(4)(3)(2)(1) = 120

```
int n,i=1,f=1;
printf("\n Enter The Number:");
scanf("%d",&n);
//LOOP TO CALCULATE FACTORIAL OF A NUMBER
do{
  f=f*i;
  i++;
} while(i<=n);
printf("\n The Factorial of %d is %d",n,f);
```

Prime Number using Do while

- Write a C program that reads a number and check whether its' prime or not.
- **Note: a prime number is the number that can only be divisible by 1 and it's self.**
- **Examples: 2, 3, 5, 7, 11, 13… etc**

```
int n, i=1, c = 0;
 printf("Enter any number n:");
 scanf("%d", &n);
 //logic
 do{ if (n % i == 0) {
       c++;
    }
 i++;}
 while ( i <= n);
 if (c == 2) {
      printf("n is a Prime number");
 }
 else {
       printf("n is not a Prime number");
 }
```

Count number of Digits

- Write a C program to input a number from user and find number of the digits using loops.
- **Example: supposedly the user entered the following number:**
  1247 then it's number of digits is 4

```c
int main()
{
    long long num; // use integer type
    int count = 0;

    printf("Enter any number: ");
    scanf("%lld", &num); // use %lld for long long

    // handle negative numbers
    if (num < 0)
        num = -num;

    // special case: when number is 0
    if (num == 0)
        count = 1;
    else {
        while (num != 0)
        {
            count++;
            num /= 10;
        }
    }

    printf("Total digits: %d\n", count);
```

Reverse of the Given Number Using While Loops

- **Write a C program to input a number from user and find reverse of the given number using loops.**
- **Example: supposedly the user entered the following number:**

1247 then it's reverse is 7421

```
int n, reverse = 0, remainder;

printf("Enter an integer: ");
scanf("%d", &n);

while (n != 0) {
  remainder = n % 10;
  reverse = reverse * 10 + remainder;
  n /= 10;
}

printf("Reversed number = %d", reverse);
```

Palindrome number using while loop

- Write a C program to input number from user and check number is palindrome or not using loop.
- **Note: *Palindrome number* is such number which when reversed is equal to the original number.**
- **Examples:**

121, 12321, 1001 … etc.

```
int n, reversed = 0, remainder, original;
  printf("Enter an integer: ");
  scanf("%d", &n);
  original = n;
// reversed integer is stored in reversed variable
  while (n != 0) {
      remainder = n % 10;
      reversed = reversed * 10 + remainder;
      n /= 10;
  }
  // palindrome if orignal and reversed are equal
  if (original == reversed)
      printf("%d is a palindrome.", original);
  else
      printf("%d is not a palindrome.", original);
```

Tasks

EXERCISE1: Write a program to check whether a number is prime or not. (not nested loops)

Note: A number is prime if it is divisible only by one and itself. Remember two is the only even and the smallest prime number. First few prime numbers are 2, 3, 5, 7, 11, 13, 17.... etc.

EXERCISE

Write a C program that:

1. ask the user to input X as an integer value.

2. if X is less than or equal to 0, print an error message and exit the program.

2. print the following pattern.

```
Enter the number of rows: 7
1
22
333
4444
55555
666666
7777777
```

EXERCISE: Write a program to evaluate the following equation using nested loops and print the result

Notes:

- **n** is a number that should be entered by the user

$$\sum_{i=4}^{i=20} \sum_{j=5}^{j=i} j^2 * (i^3 - 3) + n$$