



Purpose:

In this work sheet, you will implement different algorithms by using loops and nested loop structures.

Experiments:

- a. Consider the following C program (and type it in!):

```
#include <stdio.h>

int main()
{
    float number = 1.0;
    float stepSize = 0.05;

    while (number != 0.0)
    {
        printf("%f\n", number);
        number = number - stepSize;
    }
}
```

What do you observe when you run this program? What is wrong? How can you fix the problem?

Tasks:

- a. Calculate the value of π from the infinite series:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Print a table that shows π approximated by one term of this series, by two terms, by three terms, and so on. How many terms of this series do you have to use before you first get 3.14? 3.141? 3.1415?

- b. Fibonacci numbers are defined as follows:

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}$$

The following table shows the first nine Fibonacci numbers:

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
0	1	1	2	3	5	8	13	21	34

Write a C program that asks the user to enter a number n and **iteratively** computes the n^{th} Fibonacci number.

Sample Run:

```
Enter n: 7
7th Fibonacci number is 13
```

- c. For a nonnegative integer n , the factorial of n , written as $n!$ is defined by

$$0! = 1 \quad n! = n(n-1)\dots 3..2..1 \text{ for } n > 0$$

Write a C program that takes an n value from the user and **iteratively** calculates the $n!$, for example, if user enters 4, then your program needs to return 24 which is $4*3*2*1$.

Sample Run:

```
Enter n: 5
5! Is 120
```

- d. The value of the mathematical constant e can be expressed as an infinite series:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

Write a program that approximates e by computing the value of

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!} \text{ where } n \text{ is an integer entered by the user.}$$

Sample Run:

```
Enter n: 3
e is 2.67
```

- e. Write a nested loop that displays the following output:

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

- f. Assume that there are 2 players (player 1 and player 2) who will play for the tennis championship. 5 matches will take place to conclude which player will win the tennis championship. For each single match, players will play 3 sets. A player should win at least 2 sets to be considered as the winner of that single match. The player who wins >2 matches will be the champion. Write a C program that gets match scores and find the champion. Use nested loop.

Sample Run:

```
Tennis Championship!
Match 1 starts!
Enter set 1 score of player 1: 1
Enter set 1 score of player 2: 0
Player 1 wins set 1
Enter set 2 score of player 1: 1
Enter set 2 score of player 2: 0
Player 1 wins set 2
Enter set 3 score of player 1: 0
Enter set 3 score of player 2: 1
Player 2 wins set 3
Player 1 is the winner of match 1

Match 2 starts!
Enter set 1 score of player 1: 1
Enter set 1 score of player 2: 0
Player 1 wins set 1
Enter set 2 score of player 1: 0
Enter set 2 score of player 2: 1
Player 2 wins set 2
Enter set 3 score of player 1: 0
```

Enter set 3 score of player 2: 1
Player 2 wins set 3
Player 2 is the winner of match 2

Match 3 starts!
Enter set 1 score of player 1: 1
Enter set 1 score of player 2: 0
Player 1 wins set 1
Enter set 2 score of player 1: 0
Enter set 2 score of player 2: 1
Player 2 wins set 2
Enter set 3 score of player 1: 0
Enter set 3 score of player 2: 1
Player 2 wins set 3
Player 2 is the winner of match 2

.....

At the end of 5 matches, Player 2 is the champion!

Recommended Reading: Chapter 5,

Recommended Exercises: All practical exercises in Chapter 5.