COMPUTER PROGRAMMING LABORATORY
Experiment # 4:
Functions

OBJECTIVES

The main purpose of this experiment is to introduce you to Python Functions. In this experiment, firstly, void and return-value type functions are examined. Then, some examples are studied.

QUESTIONS

1) Consider we have a football league that consists of 3 teams. Each team makes 30 matches. Each match scored as follows:

Won (W): Winner team takes 3 points
Draw (D): Both teams take 1 point.
Lost (L): Losing team takes 0 point.

An example data of the league is given in the following table.

Team	Won	Draw	Lost	Points
A	13	3	14	42
B	22	6	2	72
C	15	4	11	49

The champion of the league is Team B.

Write a Python program to determine champion of the league. In *main* function, for each team, initialize number of won games randomly in the interval of [15-25], draw games randomly in the interval of [3-6]. Number of lost games must be equal to 30-(won+draw) games. **The program must include the *points* function that receives number of won**, **draw and lost games for each team and returns the points of the team.** In *main* function, determine and print champion of the league according to points of teams.

2) Repeat Question 1. **The program must include the *generate* function.** In *generate* function, for each team, initialize and return number of won games randomly in the interval of [15-25], draw games randomly in the interval of [3-6]. Number of lost games must be equal to 30-(won+draw) games. **The program must include the *points* function that receives number of won**, **draw and lost games for each team and returns the points of the team.** In *main* function, determine and print champion of the league according to points of teams.

3) Write a Python program to determine the quotient and the remainder of the division when two integers are divided. In *main* function, prompt the (n1 and n2) integer values. **The program must include the *division* function that receives n1 and n2 and returns quotient and the remainder.** In main function, print (n1 and n2) integer values, the quotient and the remainder. Test your program with

i) n1=146 and n2=9.
ii) n1=130 and n2=3.

E-mail : burakaleci@gmail.com

4) Write a Python program to calculate square root of a number. In *main* function, prompt the integer value (i.e x). **The program must include the *root* function that receives x and returns square root of the x and number of iterations.** In main function, print the x, its square root and number of iterations. In order to calculate square root of a number, use Newton's method.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{1}$$

For square root special case $f(x) = x^2 - a$ formula is converted into following version

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right) \tag{2}$$

If $|x_n - \frac{a}{x_n}| < \varepsilon$, stop the iteration. Consider ε is a global constant and its value is 10^{-15}. Test your program with $x_0 = 4$.

5) Repeat Question 4 for $f(x) = x^2 - 11x + 28$. Use formula 1. (**Hint: Use SymPy and NumPy modules**). If $|\frac{f(x_n)}{f'(x_n)}| < \varepsilon$, stop the iteration. Test your program with $x_0 = 3$.

6) A prime number (or a prime) is a natural number greater than 1 that cannot be formed by multiplying two smaller natural numbers.

Write a Python program that takes a number and determines whether the number is prime number or not. The program must include ***main*** function. In main function prompt a number. In main function print the number. The program must include the ***primes*** function to determine whether the number is prime number or not. In primes function, return 1 if the number is prime number. Otherwise, return 0. Print the result in *main* function after call the function.

7) The prime factorization of an integer is the multiset of primes whose product is the integer.

For example, 3757208 = 2*2*2*7*13*13*397.

Write a Python program to determine and print prime factorization of a number. In *main* function, prompt the integer value (i.e x). **The program must include the *prime_factor* function that receives x and prints prime factorization of x.**

(Hint: We can stop looking for factors when `factor*factor` is greater than `n` because if an integer `n` has a factor, it has one less than or equal to the square root of `n`.)

Test your program with x= 3757208 and x= 287994837222311

8) Write a Python program that takes a number (i.e n) and prints an n-by-n table such that there is an * in row i and column j if the gcd of i and j is 1 (i and j are relatively prime), and a space in that position otherwise.

The program must include ***main*** function. In main function prompt a number.In main function print the number.The program must include the ***primes*** function to print an n-by-n table such that there is an * in row i and column j if the gcd of i and j is 1 (i and j are relatively prime), and a space in that position otherwise. The output of the program must be following example. Test your program with n=5, n=11.

```
Enter number 5
5
1 2 3 4 5
* * * * * 1
*   *   * 2
* *   * * 3
*   *   * 4
* * * *   5
```

```
Enter number 11
11
1 2 3 4 5 6 7 8 9 10 11
* * * * * * * * * * * 1
*   *   *   *   *   * 2
* *   * *   * *   * * 3
*   *   *   *   *   * 4
* * * *   * * * *   * 5
*       *   *       * 6
* * * * * *   * * * * 7
*   *   *   *   *   * 8
* *   * *   * *   * * 9
*   *       *   *   * 10
* * * * * * * * * *   11
```