PROGRAMMING FOR ALL – SPRING 2021 - WEEK #9 – 200524 & 200526

Topics for the week: Functions

- Introduction to lists - refresh **read 7.1, 7.2**
- List slicing **read 7.3**
- List methods **read 7.4, 7.5**
- Processing lists read 7.7
- **Tuples** **read 7.9**

Class exercises:

LISTS

- **Recursive List**

Write a value returning function `rec_list` with two parameters representing a first element in a list and a number of element in a list. Element in the list are created following way: each next element in the list is obtained from the previous by multiplying that element by two and adding the value three.

Then write a program that prompts a user to enter a value of the first element and the number of element to create the list recursively.

- **Grading with lists and functions**

Write following value returning functions:

A value returning function `ave_score` with a parameter representing a list of scores, which returns an average value of scores.

A value returning function `grade` with a parameter representing an average score from scores which returns a grade. Grades are assigned following way:

Ave score in %	Letter Grade
90–100	A
80–89	B
70–79	C
60–69	D
Below 60	F

Then write a program that prompts a user to enter scores from his/her exams separated by commas and displays a grade received.

- **Grading with lists and functions – update – worst score dropped**

Update the previous program such way that one worst score is dropped (should be done in the function `ave_score`).

Write following value returning functions:

A value returning function `ave_score` with a parameter representing a list of scores, which returns an average value of scores.

A value returning function `grade` with a parameter representing an average score from scores which returns a grade. Grades are assigned following way:

Ave score in %	Letter Grade
90–100	A
80–89	B
70–79	C
60–69	D
Below 60	F

Then write a program that prompts a user to enter scores from his/her exams separated by commas and displays a grade received.

- **Counting function**

Write a value returning function `count_value` with a parameter representing a value, which executes following:

- Creates and displays a list of 10 random integers from 1 to 6;
- Counts and returns a number of appearances of the value in the list.

- **Removing element**

STEP 1:

Write following functions:

A value returning function `work_list` which generates and returns a list of 20 random integers from 1 to 10.

A value returning function `removal` with two parameters representing a list and a value, which

- returns the list with all occurrences of the value removed;
- the number of removed elements.

STEP 2:

Write a program which is using the created functions and prompts a user to enter an integer. Then the program displays:

- the generated list;
- the generated list with all occurrences of the entered integer removed;
- the number of elements removed.

If the entered integer is not in the list, the program displays "The entered integer is not in the list.

- **Initials**

STEP 1:

Write a value returning function `intials` with a parameter representing a name which generates and returns all initials of the name.

Example: If the name is Anna Marie Quinn, the function returns A.M.Q.

STEP 2:

Write a program which is using the created function and prompts a user to enter a full name. The program displays initials of the name.

The user can enter names repeatedly, by entering "$" ends the program.

- **Sum of digits list**

STEP 1:

Write following functions:

A value returning function `work_list` which generates and returns a list of 10 random integers from 100 to 500.

A value returning function `sum_dig_list` with a parameter representing a list on numbers which returns a list of sum of digits of each number in the list.

STEP 2:

Write a program which is using the created functions and displays a list of sums of digits of each number in the generated list.

GAMES

- **Number '3' in a die game**

STEP 1:

A value returning function `die_roll` with a parameter representing a number of rolls (r). The function generates and returns a list of r random integers from 1 to 6 and the number of occurrences of the value '3' in the list.

STEP 2:

Write a program which simulates a die game between a user and the computer. The user is prompted to enter a number of rolls and his/her guess about the number of occurrences of "3" in the game. Then the program displays a list of outputs. If the user's guess is correct, the user is the winner, otherwise the winner is the computer; the program displays info about it.

The user can play the game repeatedly, till enters for the number of rolls a positive integer. At the end the program displays the number of games and the number of winnings of the user.

- **Number of heads in a coin game**

STEP 1:

A value returning function `coin_toss` with a parameter representing a number of tosses (t). The function generates and returns a list of t random values 'H' and 'T' and the number of occurrences of the value 'H' in the list.

STEP 2:

Write a program which simulates a coin game between a user and the computer. The user is prompted to enter a number of tosses and his/her guess about the number of 'H' occurrences in the game. Then the program displays a list of outputs. If the user's guess is correct, the user is the winner and wins $1.50, otherwise the winner is the computer and the user lose $1; the program displays info about it.

The user can play the game repeatedly, till enters for the number of tosses a positive integer. At the end the program displays the number of games, the number of winnings of the user, and $ gain of the player.