PROGRAMMING FOR ALL – SPRING 2021 - WEEK #3 – 210412 & 210414

Topics for the week:

Read in Chapter 3:

- Decision structure - Logical operators – refresh
- Boolean variables & Flag variables
- Menu driven programs

Read in CHAPTER 3 and 8.1, 8.2:

- Strings basics – comparison of strings & characters in strings
- Strings, the function `len`, and characters in strings - refresh
- String slicing
- String operations

Read in CHAPTER 4:

- Repetition Structures – introduction

- The `for` loop introduction
- The `for` loop with lists
- The `for` loop with ranges

LOGICAL OPERATORS REFRESH

- **Color Mixer**

The colors red, blue, and yellow are known as the primary colors because they cannot be made by mixing other colors. When you mix two primary colors, you get a secondary color, as shown here:

- When you mix red and blue, you get purple.
- When you mix red and yellow, you get orange.
- When you mix blue and yellow, you get green.

Design a program that prompts the user to enter the names of two primary colors to mix. The program should display the name of the secondary color that results.

Example of a possible program output:

```
Enter two different primary colors.
Your color#1 (red, blue or yellow) is:  red
Your color#2 (red, blue or yellow) is:  blue

Your colors are: red and blue .

By mixing these two colors you will get:  purple
>>>
```

BOOLEAN VARIABLES AND FLAG VARIABLES

- **Grading**

Write a program that asks a user to provide scores (in percentage) from three tests. The program should display:

- An average score in percentage;
- A grade, which follows the rules:

Average Test Score in %	Grade
score>=95	A
85 <= score < 95	B
70 <= score <85	C
60 <= score < 70	D
Below 60	F

- An alert message "`You have to improve your scores!`" if the average score is below 70% or at least one test was below 60%.

Specification: for the alert message use a flag Boolean variable `alert`.

Menu driven programs

- **Cube menu**

Write a program that asks the user to provide a length of the base of a cube in centimeters (including a decimal numbers).

Based on the wish of the user the program displays the surface area of the cube or the volume of the cube, or both.

Specification:

- All displayed values should be rounded to two decimal places.
- The wish of the user should be done by offering a menu and receiving the user choice.

STRINGS intro

- **String work**

Write a program that asks a user to enter a string and then displays:

- the number of characters in the string;
- the first and the last character of the string with dots between;
- info whether the first and last character are same.

- **String, Alphabet Comparison**

Write a program that asks the user to enter two names and then displays those two names in alphabetical order.

- **Initials**

Write a program that asks a user to enter his/her first and last name (as two inputs) and then displays initials of the user.

Example: If the users enters Paul Smith, the computer displays P.S.

- **Strings with seven characters**

Write a program that asks a user to enter a string with 7 characters and then displays:

- the string without the first and last character;
- a character in the middle of the string;
- a new string created from the entered string such way that middle character is replaced by 'X'.

If the user doesn't enter 7 character string, the program displays: "wrong input"

Example:

```
This program displays portions of a string.
Enter a string with 7 characters: SEATTLE
The string with first and last characters missing is:  EATTL
The middle character is:  T
The modified string is:  SEAXTLE
>>>
```

- **Middle characters in strings**

Write a program that asks a user to enter a string and then displays:

- The middle character, if the entered string has an odd number of characters;
- The two middle characters, if the entered string has an even number of characters.

Example:

```
This program displays middle characters of a string.
Enter a string: 1234567
The middle character is:  4
>>>
 RESTART: C:\Users\horaki\Desktop\TCSS141_WINTER2021\WEEK
_21Q1_W4_CW_strings.py

This program displays middle characters of a string.
Enter a string: 12345678
The middle characters are:  45
>>>
```

- **New integer**

Write a program that asks a user to enter an at least two digits integer and then creates a **new integer** following way:

- the first part of the new integer is a square of the first digit of the entered integer;
- the second part of the new integer is a square of the last digit of the entered integer.

Example:

```
This program displays a new ineteger created from an entered integer.
Enter a positive integer with at least two digits: 45
1625
>>>
 RESTART: C:\Users\horaki\Desktop\TCSS141_WINTER2021\WEEK #4\2_CLASSWOR
_21Q1_W4_CW_strings.py
This program displays a new ineteger created from an entered integer.
Enter a positive integer with at least two digits: 91
811
>>>
```

Practice:

Update this program such way that it would work also for negative integers as inputs.

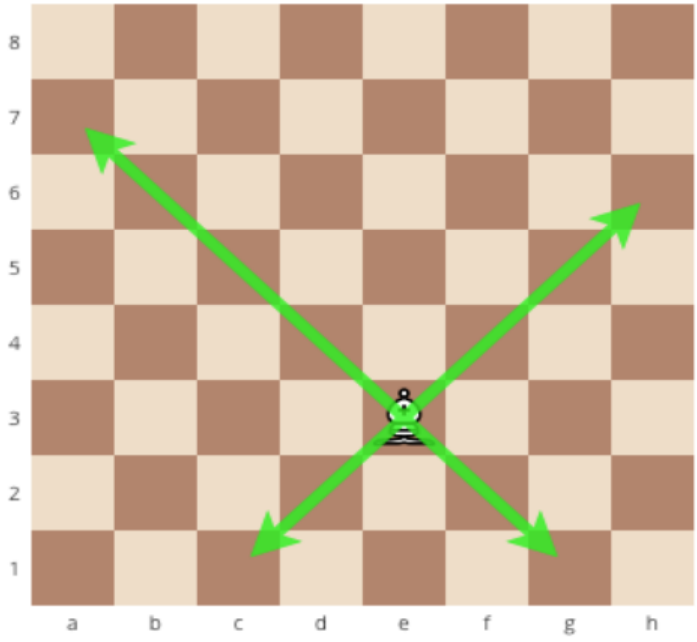
Practice:

- **Bishop moves**

In chess, the bishop moves diagonally, any number of squares.
Write a program that asks the user to provide two cells of the chessboard, the initial and the terminal cell, and displays whether the bishop can move from the initial to the terminal cell in one move.

The program should receive the input of four numbers from 1 to 8, which represent horizontal and vertical coordinates of two cells, first two - for the initial cell, and then the last two - for the terminal cell.

If the initial or terminal cell is out of the chess field, the program displays info about wrong positions of the cell, individually.
If the initial and terminal position is the same, program displays that the bishop didn't make any move.



- **Chess board - same color**

Write a program that asks the user to provide two different cells of the chessboard and displays whether they are printed in the same color, and if they are which color, black or white it is.

The program should receive the input of four numbers from 1 to 8, which represent horizontal and vertical coordinates of two cells, first two - for the first cell, and then the last two - for the second cell.

If the input is out of the chess field, the program displays info about wrong positions of the cell(s).

-