Computer science (code 30424) a.y. 2019-2020

Practice session 7 – Sequences and dictionaries

You are asked to solve the following exercises:

1) The file you need to use is called **top.py** and contains a program written in Python aimed to support the choice among different tourist destinations. The program is incomplete and contains a list called *destinations* with some possible tourist destinations.

You have to write the code necessary to complete the program by creating a function called **top_cities**. In particular, the function must:

- have a list of destinations (it is possible to use the list destinations but it must be possible to pass to the function any other list) as a mandatory parameter
- have as an optional parameter the number of cities to visit. If nothing is specified when calling the function, it must be used the number 5
- choose randomly the cities to visit from the list passed as an argument
- create a dictionary called *top* where to store the chosen destinations. The dictionary keys must be the integer numbers from 1 to the chosen number, while the values must be the cities
- return the dictionary *top*

Test the program calling the **top_cities** function passing a different number of cities as second argument.

Now create a new function called **top_cities2** that modifies the **top_cities** function so that, if the randomly chosen city has already been stored in the *top* dictionary, a new random choice is made, thus avoiding that the same city can be stored twice in the same dictionary. In any case, the number of cities to visit specified as the function argument must not change.

Test the program calling the **top_cities2** function passing a different number of cities as second argument.

2) Create a new Python file called digits.py in which you are asked to create a function called **sum_digits** that returns the sum of the digits of a number. In particular, the function must:

- ask the user to enter an integer number and store it as a string
- calculate the sum of all the digits of the number entered by the user
- return this sum

For example, if the user entered the number 2154, the function would return 12 (2 + 1 + 5 + 4).

Test the program and calling the **sum_digits** function several times, entering each time different integers.

3) The file you need to use is called **password.py** and contains two variables, *alpha_char* e *full_char*: in the first one is stored a text string with all alphanumeric characters, in the second one is stored a text string with the same alphanumeric characters plus a series of special characters.

You are asked to write the code needed to complete the program by creating a password generating function called **psw_generator**. In particular, the function must:

- ask the user if he wishes to generate a simple or complex password
- in the first case, an 8-character alphanumeric string must be generated by randomly choosing the characters of the string available in the *alpha_char* variable
- in the second case, a 20-character text string must be generated by randomly choosing the characters of the string contained in the *full_char* variable
- return the generated password

Test the program by calling the **psw_generator** function several times.

4) The file you need to use is called **crypto.py** and contains the list *cypher* in which we can find all lowercase and uppercase letters, integers from 0 to 9, plus some special characters. You are asked to create a function called **crypt** that uses a simple encryption system whereby each character of a text string is replaced with the character placed N positions ahead in the cipher.

In particular the function must:

- have a text string (it is the string that will be encrypted) as a mandatory parameter
- have as an optional parameter the number of positions needed to choose the characters in the cipher to encrypt the text string. For example, if this parameter were 10, the characters placed 10 positions ahead in the cipher would be chosen, if it were 20, the characters placed 20 positions ahead in the cipher would be chosen etc.
- replace each character in the text string passed to the function as first argument with the new characters
- if the character to be encrypted is not in the cypher list, the same character must be chosen also in the encrypted string
- return the encrypted text string

Note: We recommend using indexing to find and replace characters. If it is not possible to use a character placed N positions ahead in the cipher, use the character placed N positions behind in the cipher.

Test the program calling the **crypt** function several times.