PROGRAMMING FOR ALL – SPRING 2021 - WEEK #10 – 210531 & 210602

Monday: Holiday

Topics for the week: review

- **Binary to decimal**

STEP 1:

Write following functions:

- A value returning function `bin_dec` with a parameter representing a binary value, which returns a decimal representation of the binary value.

  Example: if the entered value is 1101, the function returns 13 because

$$1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 13$$

- A value returning function `input_validation` with a parameter representing a string, which returns a Boolean value whether the string is valid or no. A valid string can include values 0, 1 and spaces.
- A value returning function `bin_list` with a parameter representing a string of binary strings separated by a space, which returns a list of binary strings.

  Example: if the input is '1101 111 010 100' the function returns [1101,111,010,100]

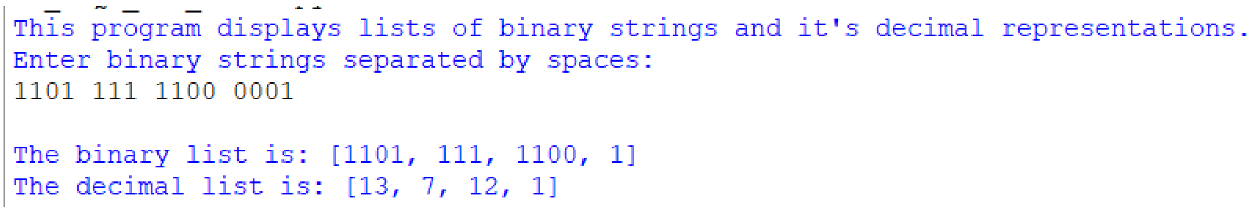
STEP 2:

Write a program which is using the created functions and prompts a user to enter a string with binary strings separated by a space. Then the program displays:

- the list of binary strings;
- the list of decimal values of all binary strings

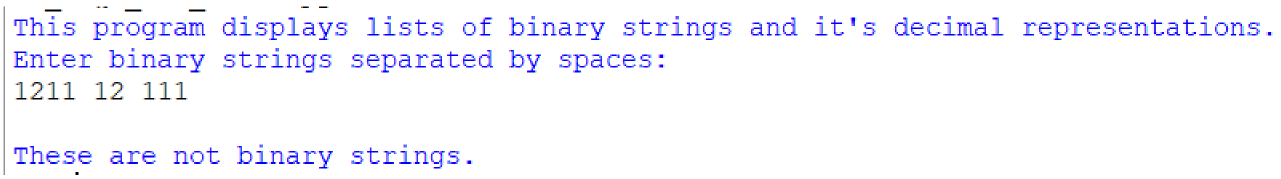
If the entered string of binary strings is not valid (it means, strings do not have only values 0 and 1) the program displays "These are not binary strings."

Example of possible outputs:



```
This program displays lists of binary strings and it's decimal representations.
Enter binary strings separated by spaces:
1101 111 1100 0001

The binary list is: [1101, 111, 1100, 1]
The decimal list is: [13, 7, 12, 1]
```



```
This program displays lists of binary strings and it's decimal representations.
Enter binary strings separated by spaces:
1211 12 111

These are not binary strings.
```

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

STEP 1:

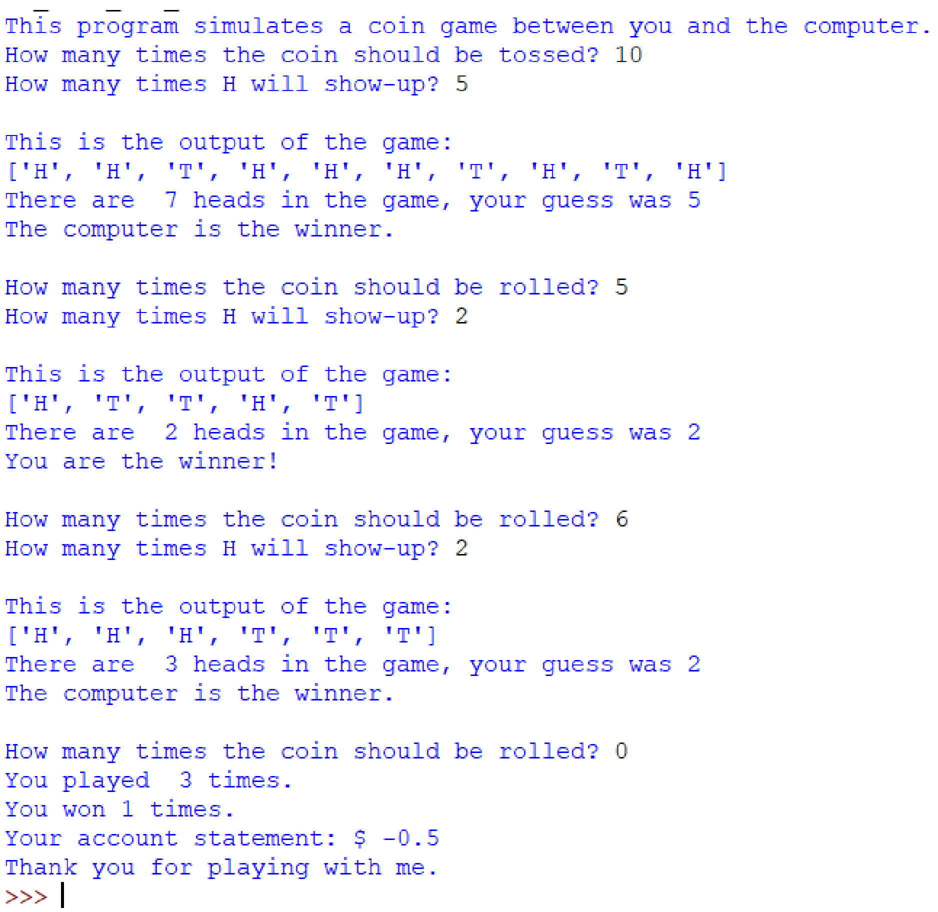
A value returning function `coin_toss` with a parameter representing a number of tosses (t). The function generates and returns a list of t random values 'H' and 'T' and the number of occurrences of the value 'H' in the list.

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.

Example of the output:



```
This program simulates a coin game between you and the computer.
How many times the coin should be tossed? 10
How many times H will show-up? 5

This is the output of the game:
['H', 'H', 'T', 'H', 'H', 'H', 'T', 'H', 'T', 'H']
There are  7 heads in the game, your guess was 5
The computer is the winner.

How many times the coin should be rolled? 5
How many times H will show-up? 2

This is the output of the game:
['H', 'T', 'T', 'H', 'T']
There are  2 heads in the game, your guess was 2
You are the winner!

How many times the coin should be rolled? 6
How many times H will show-up? 2

This is the output of the game:
['H', 'H', 'H', 'T', 'T', 'T']
There are  3 heads in the game, your guess was 2
The computer is the winner.

How many times the coin should be rolled? 0
You played  3 times.
You won 1 times.
Your account statement: $ -0.5
Thank you for playing with me.
>>> |
```