**Intro to Python**

**Exercise : Imhotep Part 2**

**Assignment**

Download, unpack, and complete as many tasks as possible from the TODO.txt file's list.

**Imhotep**

In the game of Imhotep, the player tries to build a step pyramid for the Pharaoh Zoser. He is given resources, like land to plant grain, people and workers, storehouses of grain, etc. The player tries to keep people and workers from starving, by planting grain and assigning grain to use as food, assign enough people as workers to complete levels of the pyramid, etc. The player is trying to complete 20 levels of the pyramid in 12 years or less.

Each round of the game, the play can choose the number of people to use on the workforce, the number of storehouses to use to feed the workers, the number of storehouses to use to feed the rest of the population, and the number flooded tels of land to plant with grain.

In this exercise, you will practice Python programming with function definitions, arithmetic, and printing text to the console, getting input from the user, making decisions based on user input, and repeating actions until the user gives a good input.

The functions you create will control getting the user's choices to control the game progress.

A function usually has this structure:

```python
def function_name(input_parameter1, input_parameter2):
    output_value = input_parameter1 + input_parameter2
    return output_value
```

The function_name is chosen by the programmer. The list of input parameters can have any number of parameters, and can have any name the programmer chooses. The calculation in the function can also be whatever the programmer chooses.

Arithmetic calculations can use any of the basic math operations you are familiar with. For example we could use `4 + 5`, `4 - 5`, `4 * 5` and `4 / 5`, to do addition, subtraction, multiplication and division.

We can print text to the console using the print command. For example, this prints a friendly message to the console:

```python
print "Hello!"
```

Random numbers can be created using the random module. This is an example of creating a random number from 1 to 6, inclusive.

```python
x = random.randint(1, 6)
```

Your program can get numbers from the user by calling the input function. For example:

```python
x = input(''
```

Your program can make decisions using if statements. The following code will display a mean message if the number is larger than 100 or less than 0, or a nice message if it is between 0 and 100.

```python
if x > 100 or x < 0:
    print "I don't like that number."
else:
    print "That's a good number."
```

Your program can repeat actions using the while statement. For example, this code repeats an action until the user gives a number from 0 to 100.

```python
repeat = True
```
while repeat:
    x = input("")
    if x > 100 or x < 0:
        print "I don't like that number."
    else:
        repeat = False
    print "That's a good number."

**Tasks**

For this exercise we need to create the following functions:

get_current_workers:

```python
def get_current_workers(current_population):
    This function asks the user how many people to put on
    the workforce. If the number is larger than the population,
    or if the number is less than 0, then the function prints
    a warning message. The function asks again until the user
    gives a valid number. Then, it returns current_workers.
```

Example messages:

```
Number of people you wish on the
workforce? 1000000
Zoser heard your foolishness. Now...
Number of people you wish on the
workforce? -1
Zoser heard your foolishness. Now...
Number of people you wish on the
workforce? 90
```

get_storehouses_to_feed_workers:

```python
def get_storehouses_to_feed_workers(current_storehouses, current_workers):
    This function asks the user how many storehouses to use to feed
    workers. If the number is more than the current number of storehouses
    or the number is less than 0, the function displays a warning message.
    The function continues to ask until the user gives a valid number.
    The function returns the storehouses_to_feed_workers.
```

Example messages:

```
From 10 storehouses owned by Ra, how many
will feed workers? 11
Do not jest, the Hawk's ears are sharp.
From 10 storehouses owned by Ra, how many
will feed workers? -1
Do not jest, the Hawk's ears are sharp.
From 10 storehouses owned by Ra, how many
will feed workers? 5
```

get_storehouses_to_feed_population:

```python
def get_storehouses_to_feed_population(current_storehouses, current_population):
    This function asks the user how many storehouses to use to feed
    remaining population. If the number is more than the current number of
    storehouses or the number is less than 0, the function displays a warning
    message. The function continues to ask until the user gives a valid number.
    The function returns the storehouses_to_feed_population.
```

Example messages:

```
From 10 remaining storehouses, how many
will feed 90 remaining people? 11
I warn you not to mock Pharaoh Zoser,
his flail is swift.
From 10 remaining storehouses, how many
```
def get_tels_to_plant(current_tels, current_storehouses, current_population)
    This function asks the user how many of the current_tels should be planted. If the number is more than the current tels, or is less than 0, the function displays a warning message. It takes 1 storehouse of grain per 100 tels to plant. So, if the number is more than 100 times the current storehouses, the function displays a warning message. Each non-worker population can plant up to 10 tels. So, if the number is more than 10 times the current_population, the function displays a warning message. The function continues, until a valid number is given. The function returns the tels_to_plant.

Example messages:

From 4000 tels, how many do you wish to plant? 4001
Zoser does not appreciate your foolishness. Now...
From 4000 tels, how many do you wish to plant? -1
Zoser does not appreciate your foolishness. Now...
From 4000 tels, how many do you wish to plant? 1001
There is only enough grain to plant 1000 tels.
From 4000 tels, how many do you wish to plant? 1000
There are only enough people to plant 900 tels.
From 4000 tels, how many do you wish to plant? 900

The test_player_choices() function is used to test your functions.

Download

- Imhotep Part2