**MCA171 Python Programming**

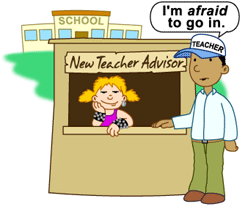
**Warm Up Exercises**

**Date: 23/8/2021 Time 2-4PM**

**Instructions:**

**Answer** **minimum of one question**. These are practice exercises and are ungraded but require mandatory submissions within 11:59PM. You need to upload your .py file and the screenshot of the output as an image, all testcases should be covered in the image you upload. You are discouraged from using search engines for obtaining code. Alternating can use any cheat sheet for developing your code.

1. You are a newbie teacher who is currently teaching MCA students in “**It’s your Risk**” College, “**Python Programming**”. You want to give your students some practice exercises but want to code yourself first so,

Write following python script named

* 1. AppleOrangeBannana : which gives output
     1. string “Apple” if the input is string
     2. string “Orange” if the input is number
     3. string “Bannana” for anything else
  2. Gerund - which gives output
     1. A string prefixed with “to” if it is ending with “ing”
     2. A string “Not a gerund” if does not end with “ing”

Example Gerund(“Painting”)-> to paint

* 1. DNAtoRNA: This script takes in a DNA string and gives as output RNA. Some information for your understanding is given below

Given a DNA strand, return its RNA complement (per RNA transcription).

Both DNA and RNA strands are a sequence of nucleotides.

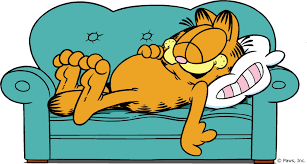
The four nucleotides found in DNA are adenine (**A**), cytosine (**C**), guanine (**G**) and thymine (**T**).

The four nucleotides found in RNA are adenine (**A**), cytosine (**C**), guanine (**G**) and uracil (**U**).

Given a DNA strand, its transcribed RNA strand is formed by replacing each nucleotide with its complement:

* G -> C
* C -> G
* T -> A
* A -> U

***lackadaisical teenager***



1. You had been a lackadaisical teenager. In conversation, your responses were very limited. You answered **'Sure.**' if asked a question, such as "How are you?". You answered **'Whoa, chill out!**' if somebody YELLED AT YOU (in all capitals).You answered **'Calm down, I know what I'm doing**!' if a question was yelled at you. You would say **'Fine. Be that way**!' if addressed without actually saying anything. You answered '**Whatever**.' to anything else.

Now that you have finished that phase of yours and are “Python Geek” in making, you want to write a script “MyTeenageResponses”, to mimic the lackadaisical teenager, you were!.

Following are few testcases for you to implement

Input => "WATCH OUT!" ,Output=>"Whoa, chill out!"

Input => ""You are, what, like 16?",Output => "Sure."

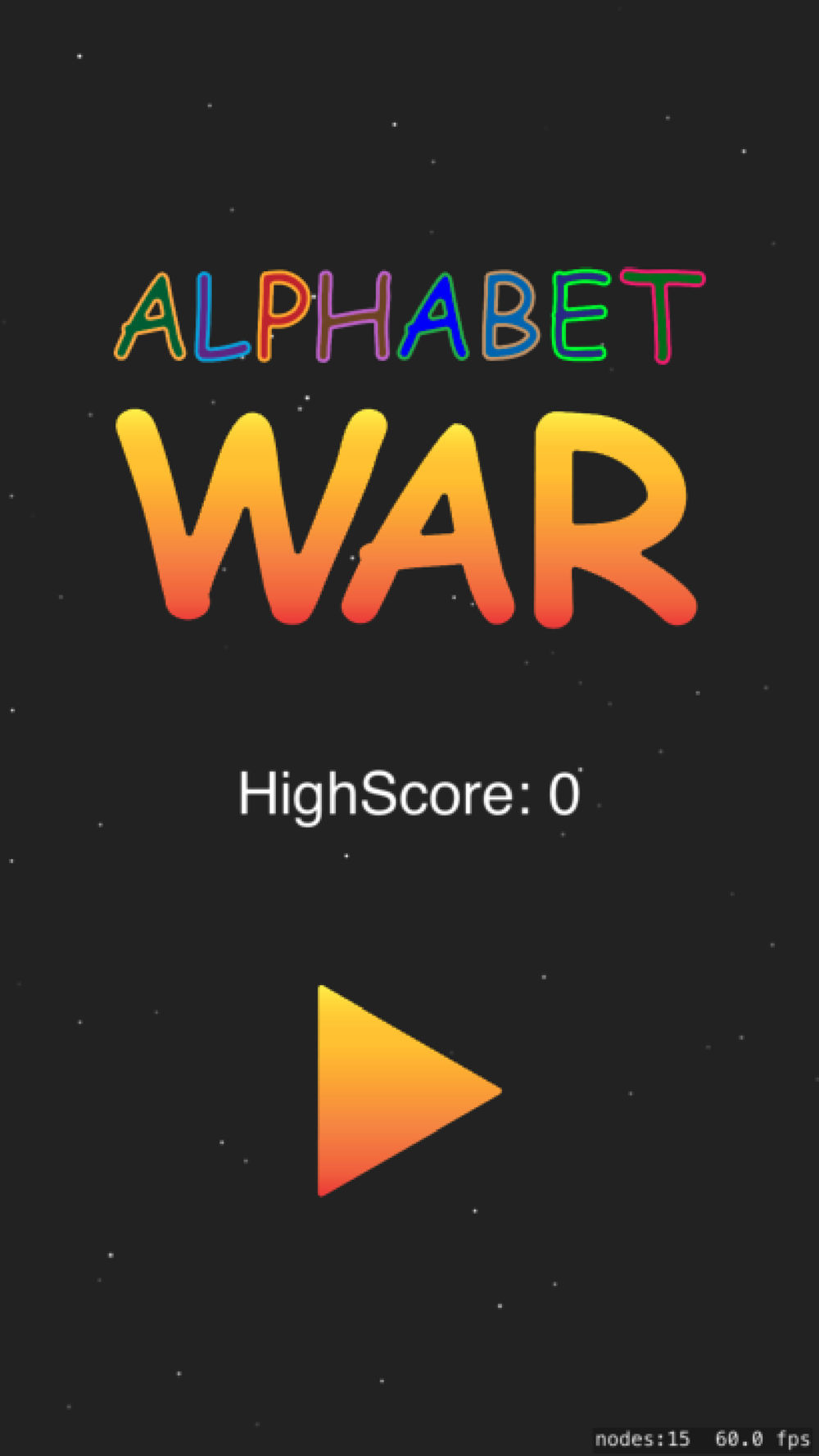
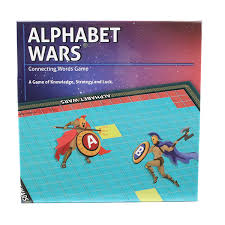
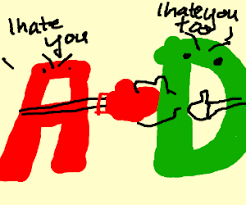
Input => ""Hi there!"),Output =>" "Whatever.”

Input => "WHAT'S GOING ON?" ,Output =>"Calm down, I know what I'm doing!"

Input => " " Output =>"Fine. Be that way!"

***Note:***

***lackadaisical teenager – Meaning : carelessly lazy***.

1. You are an ardent lover of games and “***CALL OF DUTY***” is your favourite. In addition, you are a “***Python Geek***”. You are given a task to code for a war game. In this version of the game, the war is between “**Alphabets**”-**T*he Alphabet war*!**

There are two groups of hostile letters. The tension between left-side letters and right-side letters was too high and the war began. Write a function that accepts fight string consists of only small letters and return who wins the fight. When the left-side wins return **Left side wins!** when the right-side wins return **Right side wins!** in other case return **Let's fight again!**

The left-side letters and their power:

w - 4

p - 3

b - 2

s - 1

The right-side letters and their power:

m - 4

q - 3

d - 2

z - 1

Following are few testcases for you to implement:

AlphabetWar("z") => Right side wins!

AlphabetWar("zdqmwpbs")=> Let's fight again!

AlphabetWar("wwwwwwz" )=> Left side wins!

1. You are staying in Bengaluru and are a student of CHRIST. You are also passionate gamer. Your commute on the outer ring road to reach the university (in your vehicle ) is long! To make the travel interesting you play a game.

The rules of the games are :

1. As you join the outer ring road the count is 0
2. Add 1 for every car that you overtake
3. Subtract 1 for every car that overtakes you
4. Stop counting when you reach your destination -CHRIST Main Gate.

Write a function “MyTravel\_MyGame” which takes in

* The total distance (in m)
* How fast you are going (mps)
* Information about other cars
* Their time (relative to you) as you join the outer ring road.
* For example,
* -1.5 means they already passed your starting point 1.5 minutes ago
  + 2.0 means they will pass your starting point 2.0 minutes from now
* How fast the other car is traveling (mps)

Note: Assume that all cars travel with constant speed.