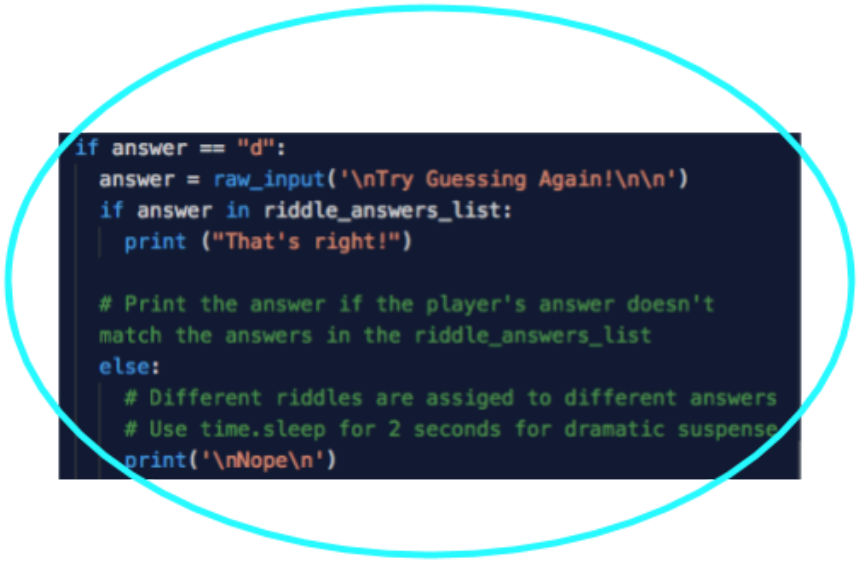
Gaganpreet Kaur
Mr. Orkney
Computer Science Principles P:3
March 1, 2019

Written Questions - Section 2

2. A. My code is programed as a riddle game that automatically prints a random riddle from my list of riddle. It is formatted in the coding language Python 2.7 and includes several if-else statements. The purpose of my video is to tell the player a riddle from my riddle-bank. The video demonstrates the player being asked a random riddle. After giving a false answer, the player decides he wants the code to give him the answer.

2. B. Throughout the coding process, even though I received much input/feedback from my peers, I started and completed my code independently. One problem I faced while completing my code was when I tried allowing the player to actually try coming up with an answer for the riddle. It required raw_input and a variable for their guess. At first I decided that the answers should be different once the riddle is picked. The biggest problem for me was comparing the guess (raw_input) to the riddle_answers. In the end, I decided to just create a list of riddle answers (riddle_answers_list) and compare the guess to the main list. In addition, I also had to repetitively write if-statements that verified which riddle was chosen and printed the right answer to the assigned riddle.

2. C.



```
if answer == "d":
    answer = raw_input('\nTry Guessing Again!\n\n')
    if answer in riddle_answers_list:
        print ("That's right!")

    # Print the answer if the player's answer doesn't
    match the answers in the riddle_answers_list
    else:
        # Different riddles are assiged to different answers
        # Use time.sleep for 2 seconds for dramatic suspense
        print('\nNope\n')
```

The algorithms aboves are crucial for my programm to work if the player tries guessing again if their first attempt was unsuccessful. The first if-statement checks if, to the previous question, the player responded with a “d”. It then prints another statement that assigns the variable ‘answer’ to the raw_input of the following statement, “Try Guessing Again!”. The second if-statement checks if the player’s answer (raw_input) matches in the possible correct answers in the riddle_answers_list. If the answer does, then the program prints “That’s right!”, indicating that the player’s guess was correct. However, in the else-statement, if the player’s answer was incorrect, the program prints “Nope”.

2. D.

```
# Print the answer of the riddle if they respond "c"
if answer == "c":
    # Different riddles are assigned different answer
    # Use time.sleep to create suspense
    if riddle == MarriageRiddle:
        print ("\n" * 1)
        print ('The answer is..........')
        time.sleep(2)
        print ("No, because he's dead")
```

The segment above is an abstract that occurs when the player types “c” as the raw_input. If the riddle chosen in the beginning is the “MarriageRiddle”, then the program will print it’s answer. First, the code will print the statement, “The answer is….”. I imported time, so I can use time.sleep. Before printing the answer, the program will wait for 2 seconds to create dramatic suspense. Right after, to end the game, the answer to “MarriageRiddle” will print across the game. The abstract is very essential because it concludes the game by printing the answer. By begin able to create the first if-statement that checked the assigned the riddle, I was able to repeat the action for allow of my riddles.