SI@UCF Introduction to Programming in Python Test #2
6/22/2022

Name: _____

1) (5 pts) Write a program that reads in the number of seconds a game is played and the number of milliseconds a single frame of the game is shown (so for example, if a frame is shown for 25 milliseconds, in 100 milliseconds, four frames are shown), and calculates the number of frames shown over the course of the game. Have your program print this number out.

```
secondsPlay = int(input("How many seconds will you play?\n"))
width = int(input("How many milliseconds does one frame show?\n"))
```

2) (15 pts) In an art factory, balls of clay are made for sale. The packaging says that the ball of clay must weigh in between 45 grams and 55 grams. Write a program that reads in the current weight of a ball of clay and determines what action (if any) to take to make sure the ball of clay is ready for packaging. Your program must print out a message of one of the three forms: (a) You must add X grams of clay before packaging, (b) You must remove X grams of clay before packaging, or (c) Your clay is ready to be packages. If adding or removing clay, make sure to add or remove the minimal amount of clay necessary to get ready for packaging.

```
claySize = int(input("How many grams is the ball of clay?\n"))
```

3) (10 pts) Complete the program below so that it reads in a positive integer n from the user and then prints out the first n powers of 2, starting with 2. If the user enters n = 10, your program should print out

```
2 4 8 16 32 64 128 256 512 1024
```

```
n = int(input("Please enter n.\n"))
```

4) (10 pts) Write a segment of code that reads in integers from the user until they enter -1 and stores each of those integers in the list values. Do NOT add -1 to the list.

```
values = []

# Write your solution below.
```

5) (16 pts) Consider writing a pygame program where a sprite starts in the middle of the screen. Whenever the user presses the down arrow key (K_DOWN), the sprite moves to a random location that is below the current location of the sprite (with a new Y value in between the current one and 400) Whenever the user presses the up arrow key (K_UP), the sprite moves to a random location that is above the current location of the sprite (with a new Y value in between 0 and the current Y value). Similarly, when the user presses the left arrow key (K_LEFT), the sprite should move to a random location to its left (but with the same y value) and when the user presses the right arrow key (K_RIGHT), the sprite should move to a random location to its right.

```
import pygame, sys
from pygame.locals import *
import random

pygame.init()
DISPLAYSURF = pygame.display.set_mode((1000, 600))
sprite = pygame.image.load("sprite.gif")
sprite = pygame.transform.scale(sprite, (64, 64))

width  = 1000
height = 600
posX = 500
posY = 300

while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()

        if event.type == KEYDOWN:

            if event.key == _____________________________:

                posY = _____________________________________

            if _____________________________________________:

                posY = _____________________________________

            if _____________________________________________:

                posX = _____________________________________

            if _____________________________________________:

                posX = _____________________________________

    DISPLAYSURF.fill(pygame.Color(0, 0, 0))
    DISPLAYSURF.blit(sprite, (posX, posY))
    pygame.display.update()
```

6) (12 pts) Complete the code below so that it displays a red ball bouncing up and down at a constant rate of 5 pixels per frame. The ball should "hit the bottom of the screen" when the y-value of its center is equal to 590 and it should start coming back down when the y-value of its center is equal to 10. The x-value of the center should always remain at 400.

```
import pygame, sys
from pygame.locals import *

pygame.init()

width  = 800
height = 600

DISPLAYSURF = pygame.display.set_mode((width, height))
clock = pygame.time.Clock()

posX = 400
posY = 10
dY = 5
radius = 10

while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()

    # FILL IN ALL CODE HERE!!!









    DISPLAYSURF.fill(pygame.Color(0, 0, 0))
    pygame.draw.circle(DISPLAYSURF, pygame.Color("red"), (posX,
posY), radius)

    pygame.display.update()
    clock.tick(50)
```

7) (15 pts) Write a function that takes in a list of integers, vals, and returns True if it contains any duplicate values, and False if all the values in the list are unique. Don't worry about the efficiency of your method. As long as you solve the problem correctly, you'll get full credit.

```
def containsDuplicates(vals):
```

8) (15 pts) Write a segment of code that reads in 100,000 words from the user (not necessarily distinct) and the prints out a chart showing how many times each word appeared in the list. The chart should have one line per word, with each line having a word, followed by a space, followed by the number of times that word appeared.

9) (2 pt) Topper's Craft Creamery sells $1 cones on campus on Wednesdays (today). What are you allowed to add to the cone if you pay 50 cents more for each one? _____

Python Random

```
random.seed() - seeds the random number generator
random.randint(a,b) - returns a random integer in between a and
                      b, inclusive.
```

Python List Methods

```
list.append(x) - adds x to the end of list
list.clear() - empties the contents of list.
len(x) - returns the length of list x
How to create an empty list: items = []
```

Python Dictionary Information

```
How to create an empty dictionary: mydictionary = {}
To add/change entry: mydictionary[key] = value
```

PyGame Event Types

```
QUIT
KEYDOWN
KEYUP
MOUSEMOTION
MOUSEBUTTONUP
MOUSEBUTTONDOWN
```

PyGame event.key values

```
K_DOWN
K_UP
K_LEFT
K_RIGHT
```