CSCI 111, Lab 7
SNAKE!

Due date: Midnight, Tuesday, November 1, on Canvas. No late work accepted.

File names: Names of files, functions, and variables, when specified, must be EXACTLY as specified. This includes simple mistakes such as capitalization.

Individual work: All work must be your own. Do not share code with anyone other than the instructor and teaching assistants. This includes looking over shoulders at screens with the code open. You may discuss ideas, algorithms, approaches, *etc.* with other students but NEVER actual code.

Pygame: First, make sure you know how pygame works. For our purposes, it's probably best to go through the tutorial that results in the original (buggy) snake game:

`https://www.edureka.co/blog/snake-game-with-pygame/`

It is a very good tutorial and explains all the concepts you need to get going. There are many other tutorials on the web that you can consult if you need more help:

`https://www.pygame.org/wiki/tutorials`

Snake: The modified snake game is provided in the lab folder, and includes my improvements to the original one on the web.

You will make the following improvements and additions to the game. Add the following features in the following order. Do not go on to the next feature without making sure the previous one works.

Game modes: Currently the game has three modes: playing, loser, and quit. These are controlled by the booleans `loser` and `quitGame` (why didn't I use the name `quit`?).

You will get rid of these booleans and have one, `gameMode` which can take on several values:

```
# Game states
PLAYING = 'Playing mode'
MENU = 'Menu mode'
LOSER = 'Loser mode'
QUIT = 'Quit mode'
```

The constants (in all caps) are easier to type than the strings, and also make sure you have fewer spelling mistakes.

Now, for example, where my code has:

```
if quitGame:
    return()
```

you can use

```
if gameMode == QUIT:
    return()
```

Starting snake: The game is more interesting after the snake has some length. Make it so that there is a global variable `snakeStartingLength` such that the snake starts with this length.

You can start the snake anywhere you like. What happens if you just start the existing game with a longer snake? For example, instead of:

```
1     snake = [(x1,y1)]
```

you pur:

```
1     snake = [(x1-40,y1), (x1-30,y1), (x1-20,y1), (x1-10,y1), (x1,y1)]
```

what happens? Why?

How can you fix this? A simple way is to start the game with the snake moving (try it). However, for this lab, make it so the snake does not move until the player enters the first direction. You will have to modify some code.

Make the length of the snake one of the things you can choose in the menu. The default should be 5. The `+` and `-` (plus and minus) keys should change the starting length.