Computer
Science

COMPSCI 101 2020 S2

Assignment 5

Due: 4:30pm, Friday 30th October 2020.
Worth: This assignment is marked out of 30 and is worth 3% of your final mark.

- In your program there must not be any variables used outside any of the functions.
- From the CompSci 101 Assignment website: https://www.cs.auckland.ac.nz/courses/compsci101s2c/assignments/, download the skeleton Python program and rename the SkeletonA5.py file to "YourUsername_A5.py", e.g., kng001_A5.py.
- Your program must include a docstring at the top of the file containing your name, your username and a correct description of the program.

Submission

Submit your Python program, your own palette text file and your own pattern text file using the Assignment Dropbox: https://adb.auckland.ac.nz/.

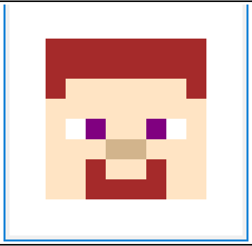
This assignment first reads a list of colours from a palette text file and creates a list of colour strings. For example, the `steve_palette.txt` contains the following lines:

```
b:brown
c:bisque
w:white
p:purple
t:tan
```

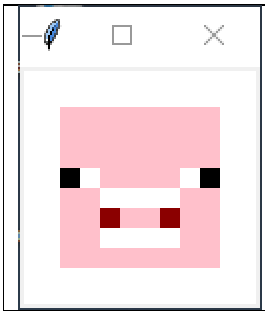
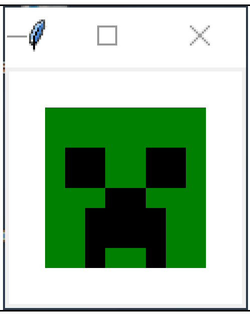
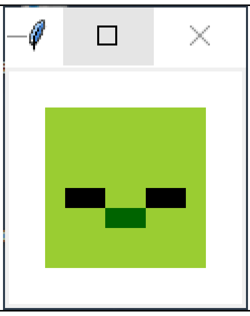
and returns a dictionary of colours:

```
{'b': 'brown', 'c': 'bisque', 'w': 'white', 'p': 'purple', 't': 'tan'}
```

Next, your program reads lines of digits from a pattern text file and draws a grid of coloured rectangles based on the information stored in the text file. Each row of the picture corresponds to a line in the file, and each rectangle along the row corresponds to a code in the line. For example, if the colour codes that we read from the above palette file are `b: brown, c: bisque, w: white, p: purple, t: tan`, the following table shows the contents of the pattern file, "`steve.txt`" and the corresponding pixel art. The first row (bbbbbbbb) represents 8 brown rectangles.

steve.txt	Pixel Art
bbbbbbbb bbbbbbbb bccccccb cccccccc cwpccpwc ccctt ccc ccbccbcc ccbbbbcc	

Here are some more examples:

	pig.txt pppppppp pppppppp pppppppp bwppppwb ppwwwwpp ppdppdpp ppwwwwpp pppppppp		creeper1.txt gggggggg gggggggg gbbggbbg gbbggbbg gggbbggg ggbbbbgg ggbbbbgg ggbggbgg		zombie.txt yyyyyyyy yyyyyyyy yyyyyyyy yyyyyyyy ybbyybby yyyddyyy yyyyyyyy yyyyyyyy

Steps:

1. Add your username to the title bar of the window **(2 marks)**
Currently the title bar of the program window displays "A5 by". Add your username to the title bar of the program, i.e., the title bar should display the string, "A5 by yourUsername", e.g., A5 by kng001.

2. Add a docstring at the top of the file containing your name, your username and a correct description of the program. **(2 marks)**

3. Complete the `get_filename(names_list, index)` function **(3 marks)**
This function takes a list of names and an index as parameters and **returns** the filename of the colour palette file and the filename of the pattern based on the parameter index. The function should return the result as a tuple. The filename of the colour palette is a name in the list and "_palette.txt" at the end and the filename of the pattern is only the name and ".txt" at the end. For example, the output produced by the following call to the function is shown below:

```
names_list = ['steve', 'pig', 'zombie', 'skeleton', 'creeper1', 'creeper2']
print("3. ", get_filename(names_list, 1))
3. ('pig_palette.txt', 'pig.txt')
```

4. Complete the `split_codes(line)` function **(2 marks)**
This function takes a `String` of codes as a parameter and returns a list of `String` elements. For example, the output produced by the following call to the function is shown below:

```
print("4.", split_codes('bwppppwb'))
4. ['b', 'w', 'p', 'p', 'p', 'p', 'w', 'b']
```

5. Complete the `process_file(filename)` function **(3 marks)**
This function takes a filename as a parameter and returns a list of `String` elements read from the input file. For example, the output produced by the following call to the function is shown below:

```
lines = process_file("steve.txt")
print("5.", lines)
5. ['bbbbbbbb', 'bbbbbbbb', 'bccccccb', 'cccccccc', 'cwpccpwc', 'cccttccc', 'ccbccbcc',
'ccbbbbcc']
```

6. Complete the `create_colours_dictionary(colours_list)` function **(3 marks)**
This function takes a list of `String` elements as a parameter and returns a colours dictionary from the parameter string. For example, the output produced by the following call to the function is shown below:

```
colours_list = ['b:brown', 'c:bisque', 'w:white', 'p:purple', 't:tan']
colours_dictionary = create_colours_dictionary(colours_list)
print("6.", colours_dictionary)
6. {'b': 'brown', 'c': 'bisque', 'w': 'white', 'p': 'purple', 't': 'tan'}
```

7. Complete the `create_pattern_list(lines)` function **(4 marks)**

This function takes a list of `String` elements as a parameter. Each element of the parameter list is a string of colour codes for a line. Each element of the pattern list created by this function is a list of codes lists. The list of codes is produced by calling the `split_codes()` function above. For example, the output produced by the following call to the function is shown below:

```
lines = ['bbbbbbbb', 'bbbbbbbb', 'bccccccb', 'cccccccc', 'cwpccpwc', 'cccttccc',
'ccbccbcc', 'ccbbbbcc']
pattern_list = create_pattern_list(lines)
print("7.", pattern_list)
7. [['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b'], ['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b'],
['b', 'c', 'c', 'c', 'c', 'c', 'c', 'b'], ['c', 'c', 'c', 'c', 'c', 'c', 'c', 'c'], ['c',
'w', 'p', 'c', 'c', 'p', 'w', 'c'], ['c', 'c', 'c', 't', 't', 'c', 'c', 'c'], ['c', 'c',
'b', 'c', 'c', 'b', 'c', 'c'], ['c', 'c', 'b', 'b', 'b', 'b', 'c', 'c']]
```

8. Complete the `draw_pattern(a_canvas, colours_dictionary, pattern_list, size, left, top)` function **(4 marks)**

This function is passed **SIX** parameters: the Canvas object, a colours dictionary, a list of `String` elements, and followed by three integer parameters. The function draws a grid of coloured rectangles inside the canvas area. The left-top position of the grid of coloured rectangles is given by the last two parameter values and the size of each rectangle is given by the size parameter. Once you have completed this function you should see rows of coloured rectangles in the canvas area.

9. Complete the program (2 marks)

At the top of the `main()` function, add code to prompt the user to enter an index as in the example below:

```
Enter an index for the pattern: 0
```

The `main()` function will then call the `get_filename()` function to retrieve the filename of the colour palette and the filename of the pattern based on the parameter index. Then, it will also call the `process_file()` function to return a list of strings and then call the `create_colours_dict()` function to return a colours dictionary. Next, the `main()` function will then call the `process_file()` function again to return a list of strings and then call the `create_pattern_list()` function to create the corresponding pattern list. Use the colours dictionary and the pattern list to draw a specific pattern in the canvas area.

9. Create your own palette and pattern text files **(5 marks)**

Your pattern should show great preparation, creativity or effort. Submit your text files to the Assignment dropbox along with your A5.py file. Some examples are given as below: