

Analysis Report

Seven Segment Display

Analysis of Data Display from Human-Readable to Digital Displays in the Context of I/O Technology

Introduction:

In the realm of data representation and interaction with computers, the transition from human-readable formats to digital displays plays a pivotal role. This analysis explores the mechanisms of data representation and their relevance in the context of Input/Output (I/O) technology. It delves into whether the technology employed for data display is categorized as serial or parallel I/O.

Human-Readable Data:

Human-readable data is information presented in a format that can be easily understood by a human observer. Examples of human-readable data include text, numbers, symbols, and graphical elements. Such data is usually presented in natural language or visual representations like charts and graphs. Human-readable data is the most intuitive form of information presentation.

Digital Displays:

Digital displays, on the other hand, represent data in a format that can be interpreted by electronic systems. These displays are designed to convey information in binary code, which is the fundamental language of computers. The two most common types of digital displays discussed in this analysis are seven-segment displays and matrix displays.

Seven-Segment Display:

A seven-segment display is a digital display technology that is used to represent numbers, letters, and some special characters. It consists of seven individual segments, each of which can be turned on or off to create a combination of illuminated segments, forming the desired character. Seven-segment displays are a simple and cost-effective way to display numeric data and are often used in digital clocks, calculators, and various electronic devices.

Parallel I/O Technology:

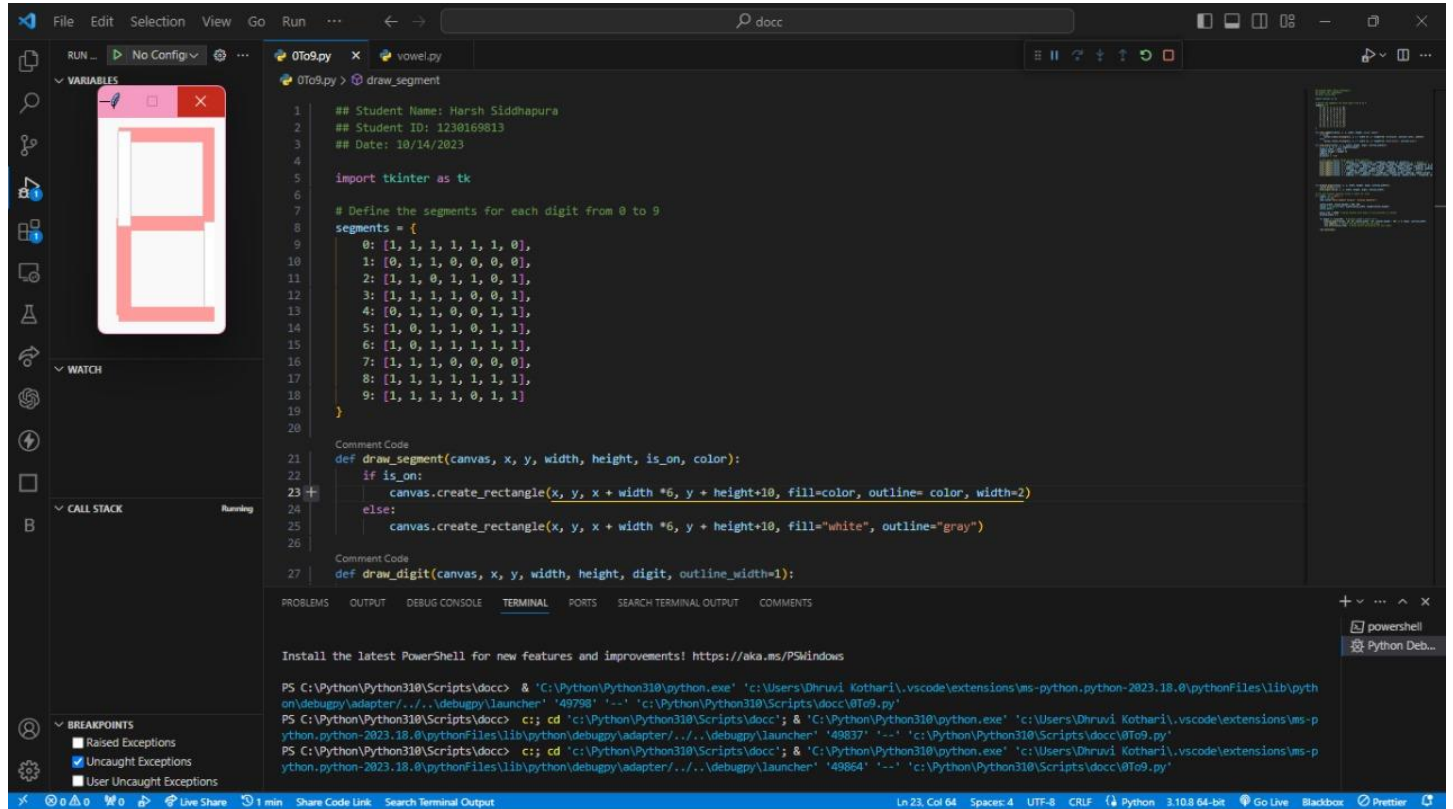
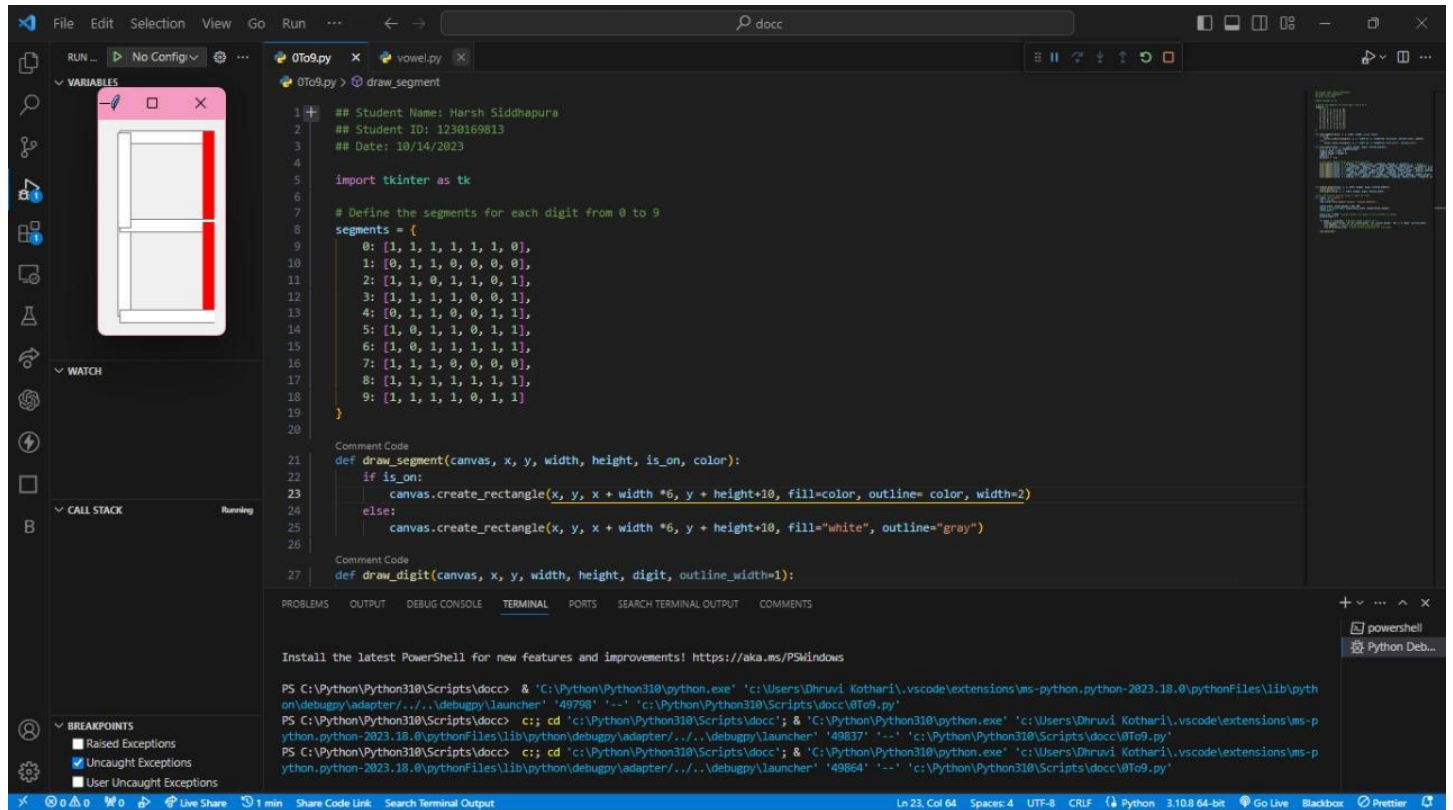
When discussing the connection between data sources and digital displays, the use of parallel I/O technology is prominent. Parallel I/O technology involves the simultaneous transmission of multiple bits (usually 8 or more) of data through multiple parallel lines. In the case of seven-segment displays, each of the seven segments is controlled by an individual line or bit. These parallel lines allow for precise control over each segment, enabling the display of specific characters. Parallel I/O is characterized by its capacity to transmit data in parallel, meaning that several bits of data are transmitted simultaneously.

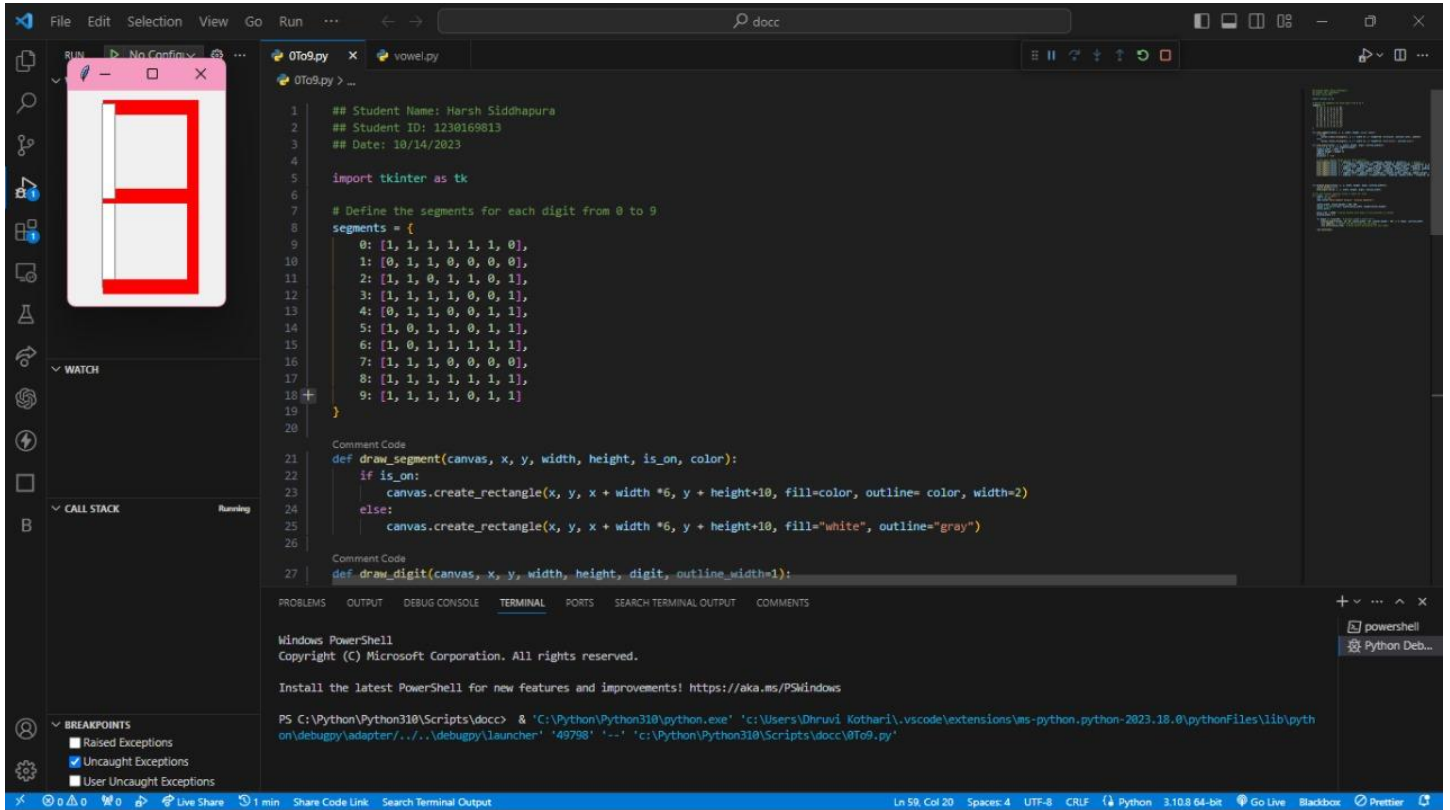
Conclusion:

In summary, the transition from human-readable data to digital displays, particularly seven-segment displays, involves a shift from an easily interpretable format to a binary representation. This shift is necessary for electronic systems to process and present data efficiently. The technology employed for driving seven-segment displays falls under the category of parallel I/O. Each of the seven segments is controlled in parallel through individual lines or bits, allowing for precise character representation. This analysis highlights the significance of digital displays and the role of parallel I/O technology in bridging the gap between human-readable data and digital representation in the computing domain.

Screenshot

0 to 9 - 7 Segment Display





The screenshot shows a Visual Studio Code editor with a Python file named `0To9.py`. The code defines a Tkinter window and a function to draw digits from 0 to 9 using segments. A small window titled `0To9.py` displays the digit '2' in red on a white background. The code includes comments for student information and a function `draw_digit` that calls `draw_segment` for each digit.

```

1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 10/14/2023
4
5  import tkinter as tk
6
7  # Define the segments for each digit from 0 to 9
8  segments = {
9      0: [1, 1, 1, 1, 1, 1, 0],
10     1: [0, 1, 1, 0, 0, 0, 0],
11     2: [1, 1, 0, 1, 1, 0, 1],
12     3: [1, 1, 1, 1, 0, 0, 1],
13     4: [0, 1, 1, 0, 0, 1, 1],
14     5: [1, 0, 1, 1, 0, 1, 1],
15     6: [1, 0, 1, 1, 1, 1, 1],
16     7: [1, 1, 1, 0, 0, 0, 0],
17     8: [1, 1, 1, 1, 1, 1, 1],
18     9: [1, 1, 1, 1, 0, 1, 1]
19 }
20
21 # Comment Code
22 def draw_segment(canvas, x, y, width, height, is_on, color):
23     if is_on:
24         canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill=color, outline= color, width=2)
25     else:
26         canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill="white", outline="gray")
27
28 # Comment Code
29 def draw_digit(canvas, x, y, width, height, digit, outline_width=1):

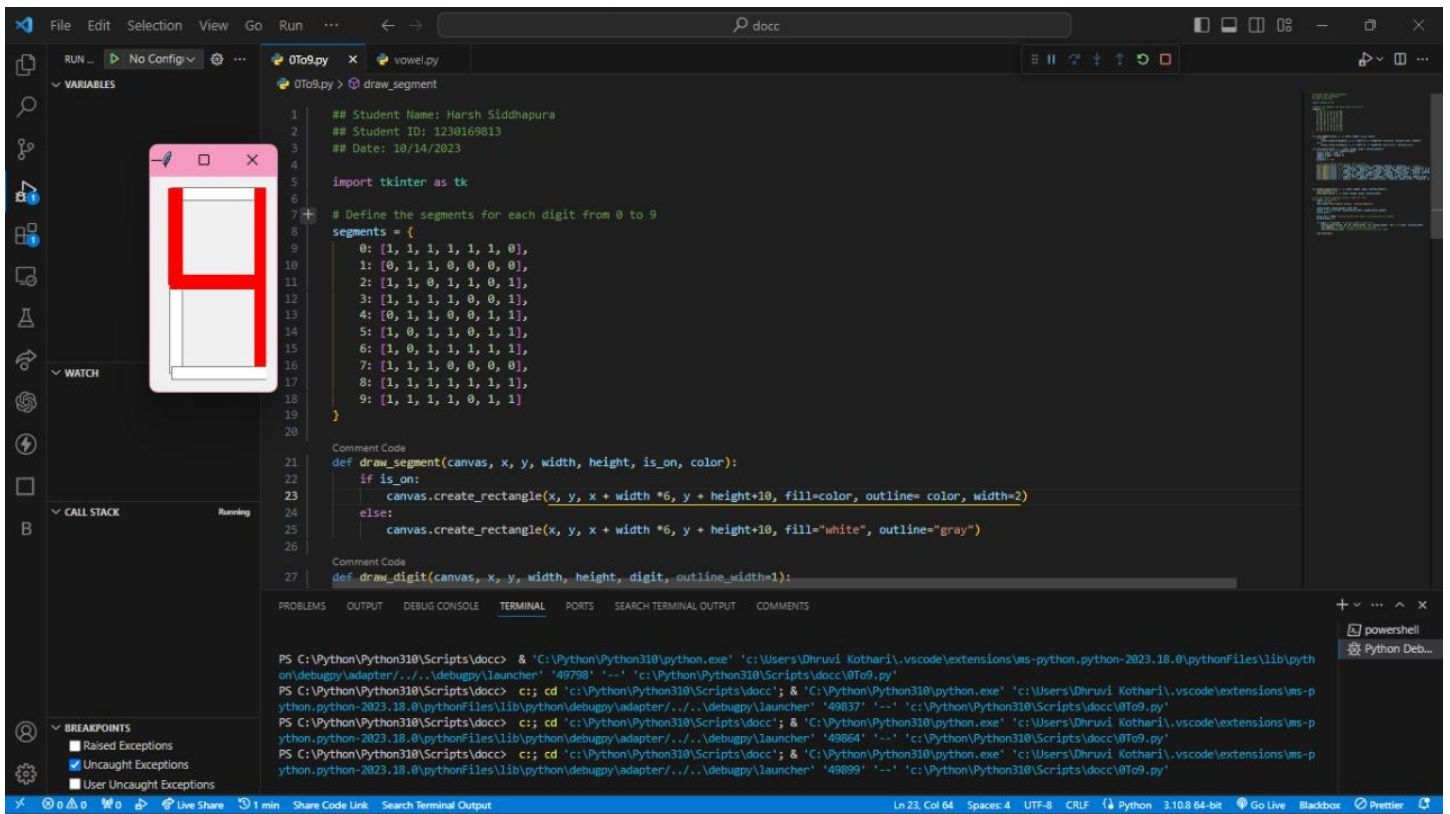
```

The terminal shows the PowerShell prompt and the command to run the script:

```

PS C:\Python\Python310\Scripts\docc> & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49798' '--' 'c:\Python\Python310\Scripts\docc\0To9.py'

```

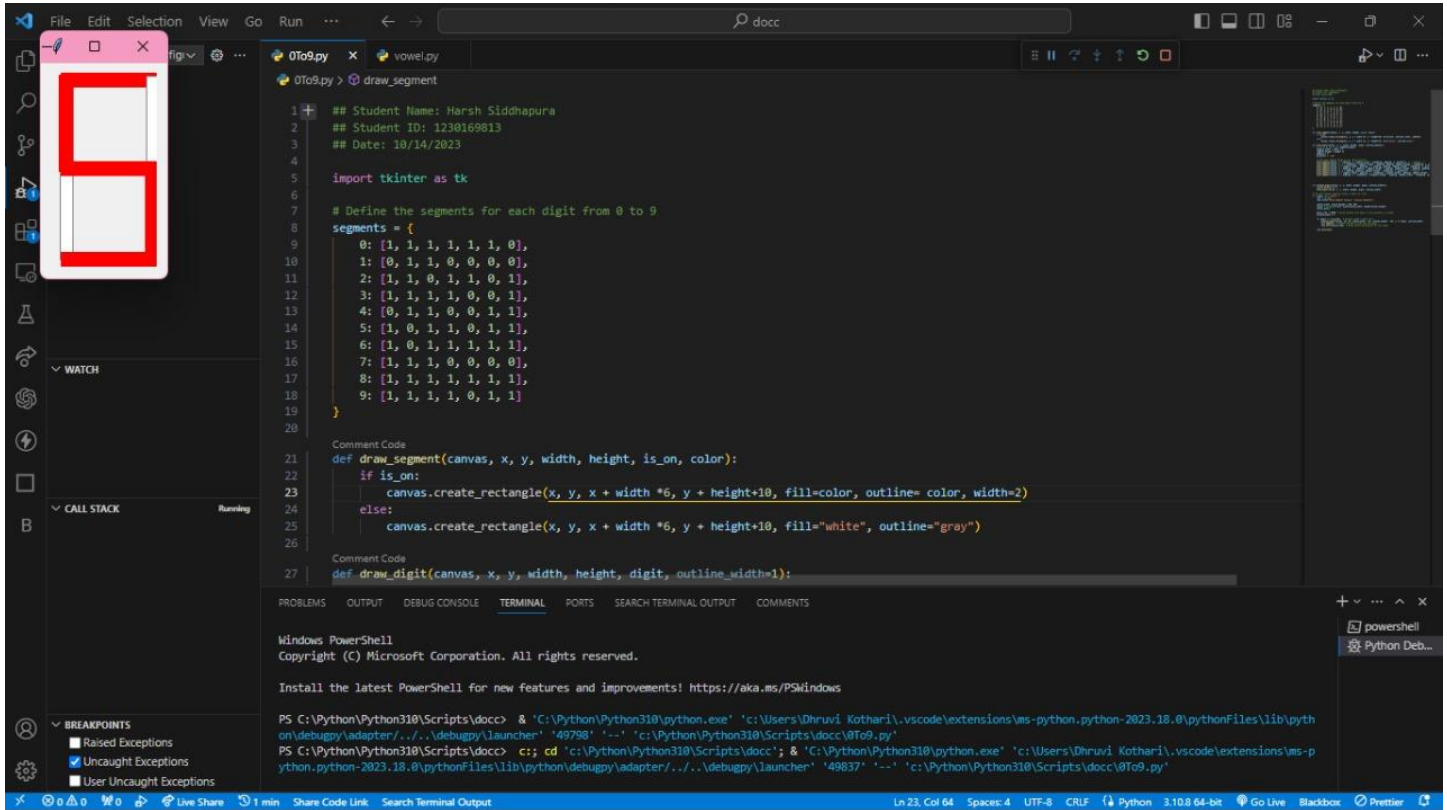


The screenshot shows the same Visual Studio Code editor with the `0To9.py` file. The small window now displays the digit '4' in red. The code is identical to the previous screenshot. The terminal shows the command to run the script:

```

PS C:\Python\Python310\Scripts\docc> & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49798' '--' 'c:\Python\Python310\Scripts\docc\0To9.py'

```

VS Code interface showing the initial state of the digit drawing program. The code defines segments for digits 0-9 and a draw_digit function. The terminal shows the command to run the program.

```

1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 10/14/2023
4
5  import tkinter as tk
6
7  # Define the segments for each digit from 0 to 9
8  segments = {
9      0: [1, 1, 1, 1, 1, 1, 0],
10     1: [0, 1, 1, 0, 0, 0, 0],
11     2: [1, 1, 0, 1, 1, 0, 1],
12     3: [1, 1, 1, 1, 0, 0, 1],
13     4: [0, 1, 1, 0, 0, 1, 1],
14     5: [1, 0, 1, 1, 0, 1, 1],
15     6: [1, 0, 1, 1, 1, 1, 1],
16     7: [1, 1, 1, 0, 0, 0, 0],
17     8: [1, 1, 1, 1, 1, 1, 1],
18     9: [1, 1, 1, 1, 0, 1, 1]
19 }
20
21 Comment Code
22 def draw_segment(canvas, x, y, width, height, is_on, color):
23     if is_on:
24         canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill=color, outline= color, width=2)
25     else:
26         canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill="white", outline="gray")
27
28 Comment Code
29 def draw_digit(canvas, x, y, width, height, digit, outline_width=1):

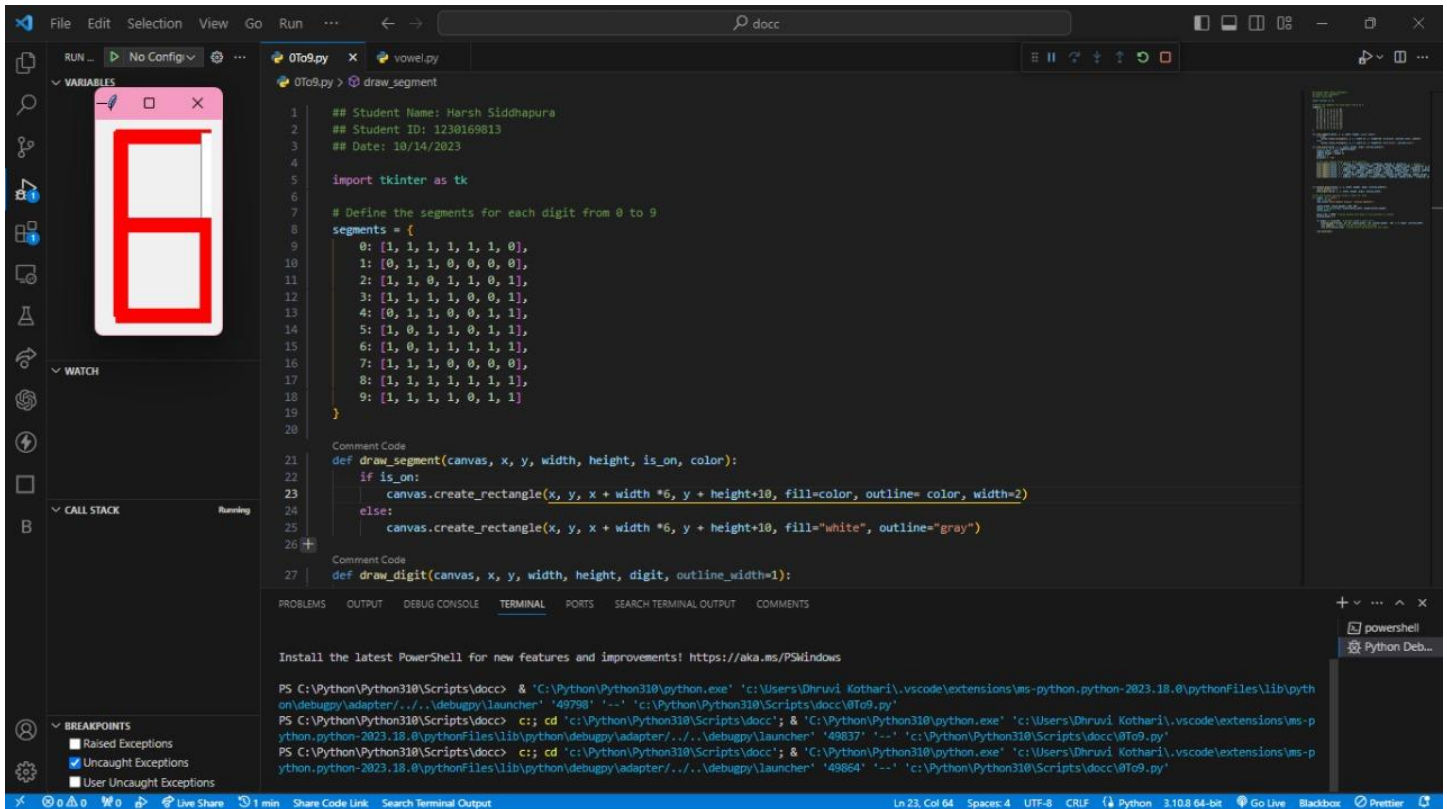
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Python\Python310\Scripts\docc> & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49798' '-' 'c:\Python\Python310\Scripts\docc\0to9.py'

PS C:\Python\Python310\Scripts\docc> c; cd 'c:\Python\Python310\Scripts\docc'; & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49837' '-' 'c:\Python\Python310\Scripts\docc\0to9.py'



VS Code interface showing the program running. The digit '0' is displayed on the canvas. The terminal shows the command to run the program.

```

1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 10/14/2023
4
5  import tkinter as tk
6
7  # Define the segments for each digit from 0 to 9
8  segments = {
9      0: [1, 1, 1, 1, 1, 1, 0],
10     1: [0, 1, 1, 0, 0, 0, 0],
11     2: [1, 1, 0, 1, 1, 0, 1],
12     3: [1, 1, 1, 1, 0, 0, 1],
13     4: [0, 1, 1, 0, 0, 1, 1],
14     5: [1, 0, 1, 1, 0, 1, 1],
15     6: [1, 0, 1, 1, 1, 1, 1],
16     7: [1, 1, 1, 0, 0, 0, 0],
17     8: [1, 1, 1, 1, 1, 1, 1],
18     9: [1, 1, 1, 1, 0, 1, 1]
19 }
20
21 Comment Code
22 def draw_segment(canvas, x, y, width, height, is_on, color):
23     if is_on:
24         canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill=color, outline= color, width=2)
25     else:
26         canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill="white", outline="gray")
27
28 Comment Code
29 def draw_digit(canvas, x, y, width, height, digit, outline_width=1):

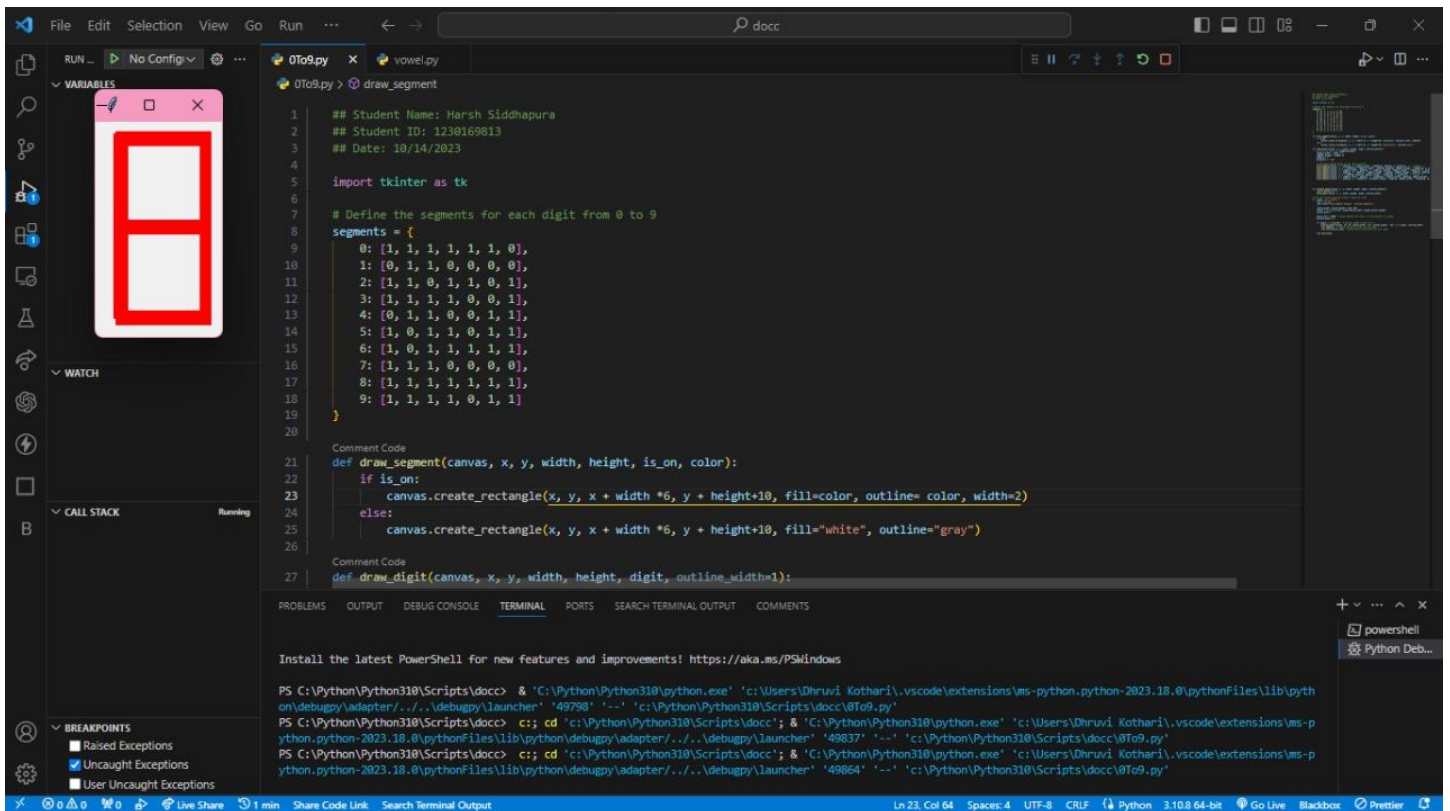
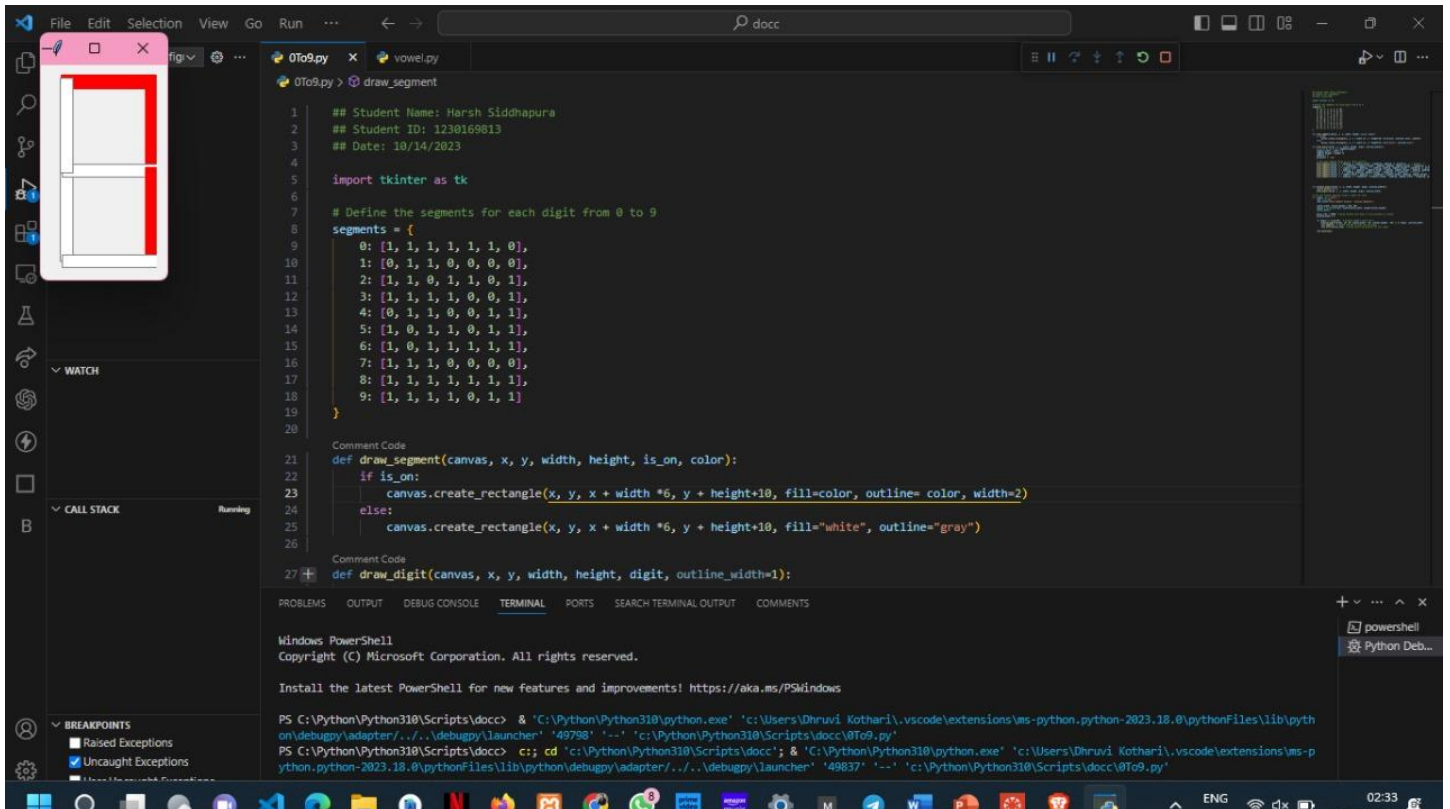
```

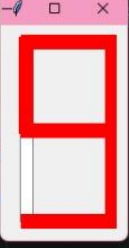
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Python\Python310\Scripts\docc> & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49798' '-' 'c:\Python\Python310\Scripts\docc\0to9.py'

PS C:\Python\Python310\Scripts\docc> c; cd 'c:\Python\Python310\Scripts\docc'; & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49837' '-' 'c:\Python\Python310\Scripts\docc\0to9.py'

PS C:\Python\Python310\Scripts\docc> c; cd 'c:\Python\Python310\Scripts\docc'; & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49864' '-' 'c:\Python\Python310\Scripts\docc\0to9.py'





File Edit Selection View Go Run ...

0To9.py x vowel.py

0To9.py > draw_segment

```

1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 10/14/2023
4
5  import tkinter as tk
6
7  # Define the segments for each digit from 0 to 9
8  segments = {
9      0: [1, 1, 1, 1, 1, 1, 0],
10     1: [0, 1, 1, 0, 0, 0, 0],
11     2: [1, 1, 0, 1, 1, 0, 1],
12     3: [1, 1, 1, 1, 0, 0, 1],
13     4: [0, 1, 1, 0, 0, 1, 1],
14     5: [1, 0, 1, 1, 0, 1, 1],
15     6: [1, 0, 1, 1, 1, 1, 1],
16     7: [1, 1, 1, 0, 0, 0, 0],
17     8: [1, 1, 1, 1, 1, 1, 1],
18     9: [1, 1, 1, 1, 0, 1, 1]
19 }
20
21 Comment Code
22 def draw_segment(canvas, x, y, width, height, is_on, color):
23     if is_on:
24         canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill=color, outline= color, width=2)
25     else:
26         canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill="white", outline="gray")
27
28 Comment Code
29 def draw_digit(canvas, x, y, width, height, digit, outline_width=1):

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TERMINAL OUTPUT COMMENTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

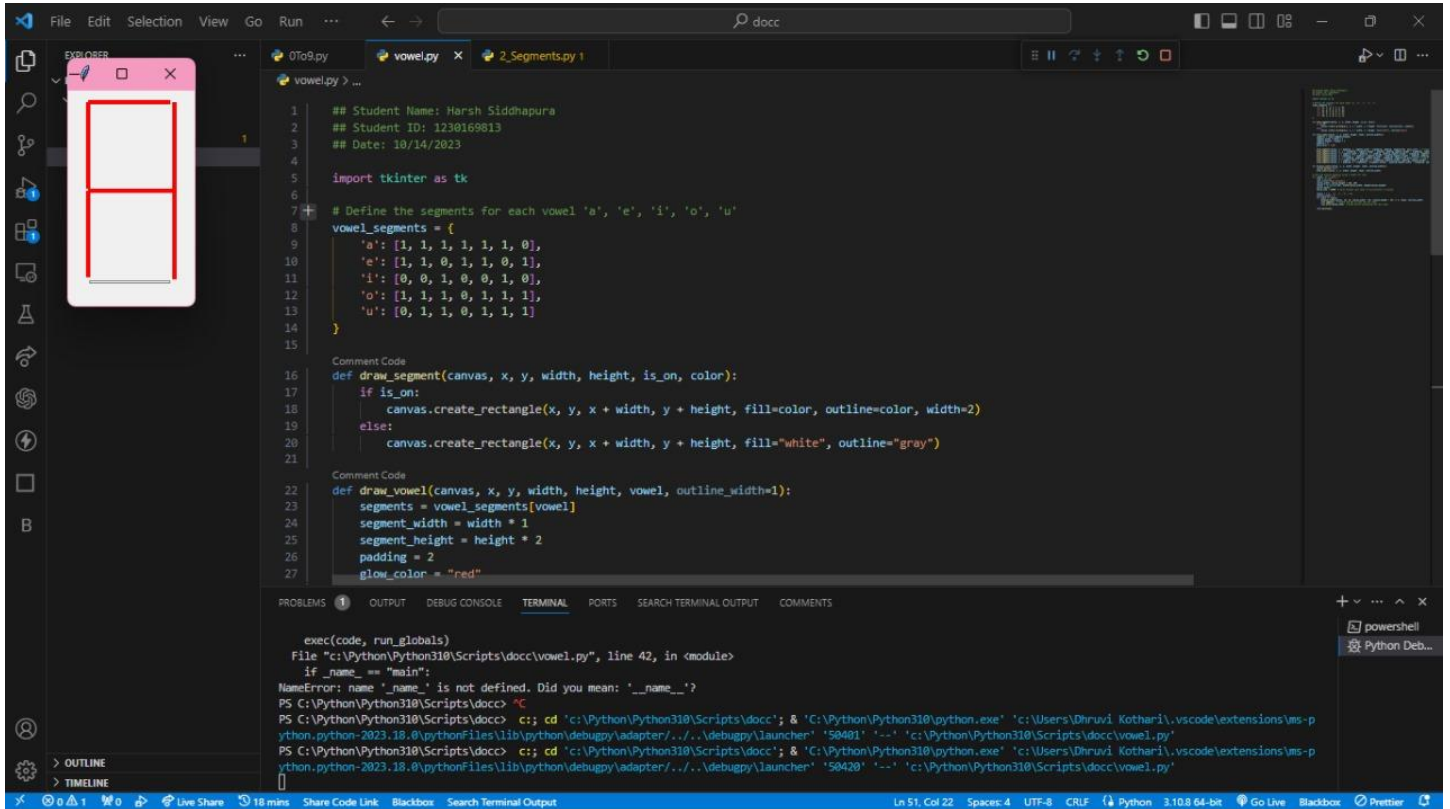
PS C:\Python\Python310\Scripts\docc> & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49798' '--' 'c:\Python\Python310\Scripts\docc\0To9.py'

PS C:\Python\Python310\Scripts\docc> c; cd 'c:\Python\Python310\Scripts\docc'; & 'C:\Python\Python310\python.exe' 'c:\Users\Dhruvi Kothari\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '49837' '--' 'c:\Python\Python310\Scripts\docc\0To9.py'

BREAKPOINTS
☐ Raised Exceptions
☒ Uncaught Exceptions
☐ User Uncaught Exceptions

Ln 23, Col 64 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit Go Live Blackbox Premier

AEIOU - 7 Segment Display



```

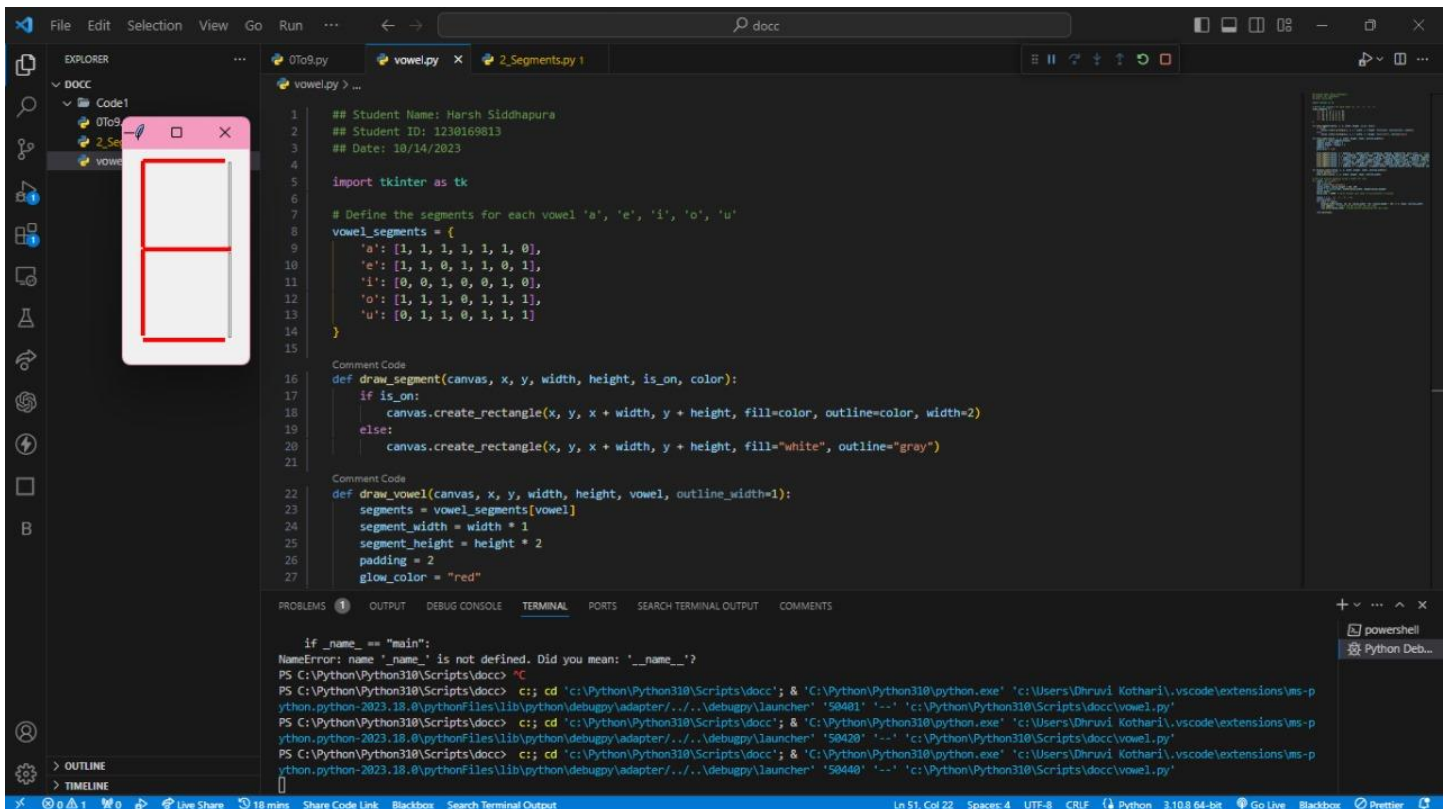
1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 10/14/2023
4
5  import tkinter as tk
6
7  # Define the segments for each vowel 'a', 'e', 'i', 'o', 'u'
8  vowel_segments = {
9      'a': [1, 1, 1, 1, 1, 1, 0],
10     'e': [1, 1, 0, 1, 1, 0, 1],
11     'i': [0, 0, 1, 0, 0, 1, 0],
12     'o': [1, 1, 1, 0, 1, 1, 1],
13     'u': [0, 1, 1, 0, 1, 1, 1]
14 }
15
16 Comment Code
17 def draw_segment(canvas, x, y, width, height, is_on, color):
18     if is_on:
19         canvas.create_rectangle(x, y, x + width, y + height, fill=color, outline=color, width=2)
20     else:
21         canvas.create_rectangle(x, y, x + width, y + height, fill="white", outline="gray")
22
23 Comment Code
24 def draw_vowel(canvas, x, y, width, height, vowel, outline_width=1):
25     segments = vowel_segments[vowel]
26     segment_width = width * 1
27     segment_height = height * 2
28     padding = 2
29     glow_color = "red"

```

```

exec(code, run_globals)
File "C:\Python\Python310\Scripts\docc\vowel.py", line 42, in <module>
  if __name__ == "__main__":
NameError: name 'name' is not defined. Did you mean: 'name__'?
PS C:\Python\Python310\Scripts\docc> cd "C:\Python\Python310\Scripts\docc"; & 'C:\Python\Python310\python.exe' 'C:\Users\Dhruvi Kothari\.vscode\extensions\ms-p
ython.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '58481' '--' 'C:\Python\Python310\Scripts\docc\vowel.py'
PS C:\Python\Python310\Scripts\docc> cd "C:\Python\Python310\Scripts\docc"; & 'C:\Python\Python310\python.exe' 'C:\Users\Dhruvi Kothari\.vscode\extensions\ms-p
ython.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '58420' '--' 'C:\Python\Python310\Scripts\docc\vowel.py'

```



```

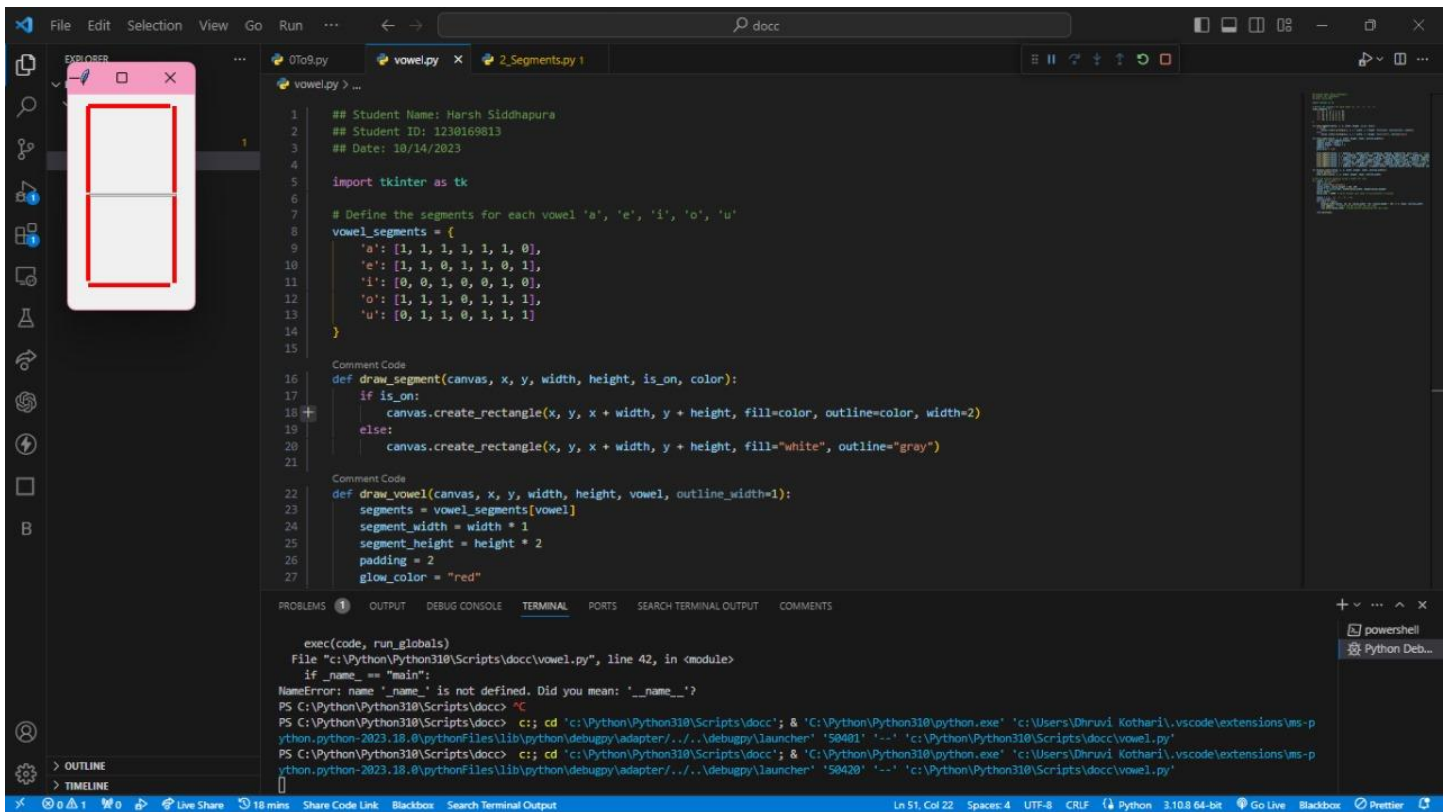
1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 10/14/2023
4
5  import tkinter as tk
6
7  # Define the segments for each vowel 'a', 'e', 'i', 'o', 'u'
8  vowel_segments = {
9      'a': [1, 1, 1, 1, 1, 1, 0],
10     'e': [1, 1, 0, 1, 1, 0, 1],
11     'i': [0, 0, 1, 0, 0, 1, 0],
12     'o': [1, 1, 1, 0, 1, 1, 1],
13     'u': [0, 1, 1, 0, 1, 1, 1]
14 }
15
16 Comment Code
17 def draw_segment(canvas, x, y, width, height, is_on, color):
18     if is_on:
19         canvas.create_rectangle(x, y, x + width, y + height, fill=color, outline=color, width=2)
20     else:
21         canvas.create_rectangle(x, y, x + width, y + height, fill="white", outline="gray")
22
23 Comment Code
24 def draw_vowel(canvas, x, y, width, height, vowel, outline_width=1):
25     segments = vowel_segments[vowel]
26     segment_width = width * 1
27     segment_height = height * 2
28     padding = 2
29     glow_color = "red"

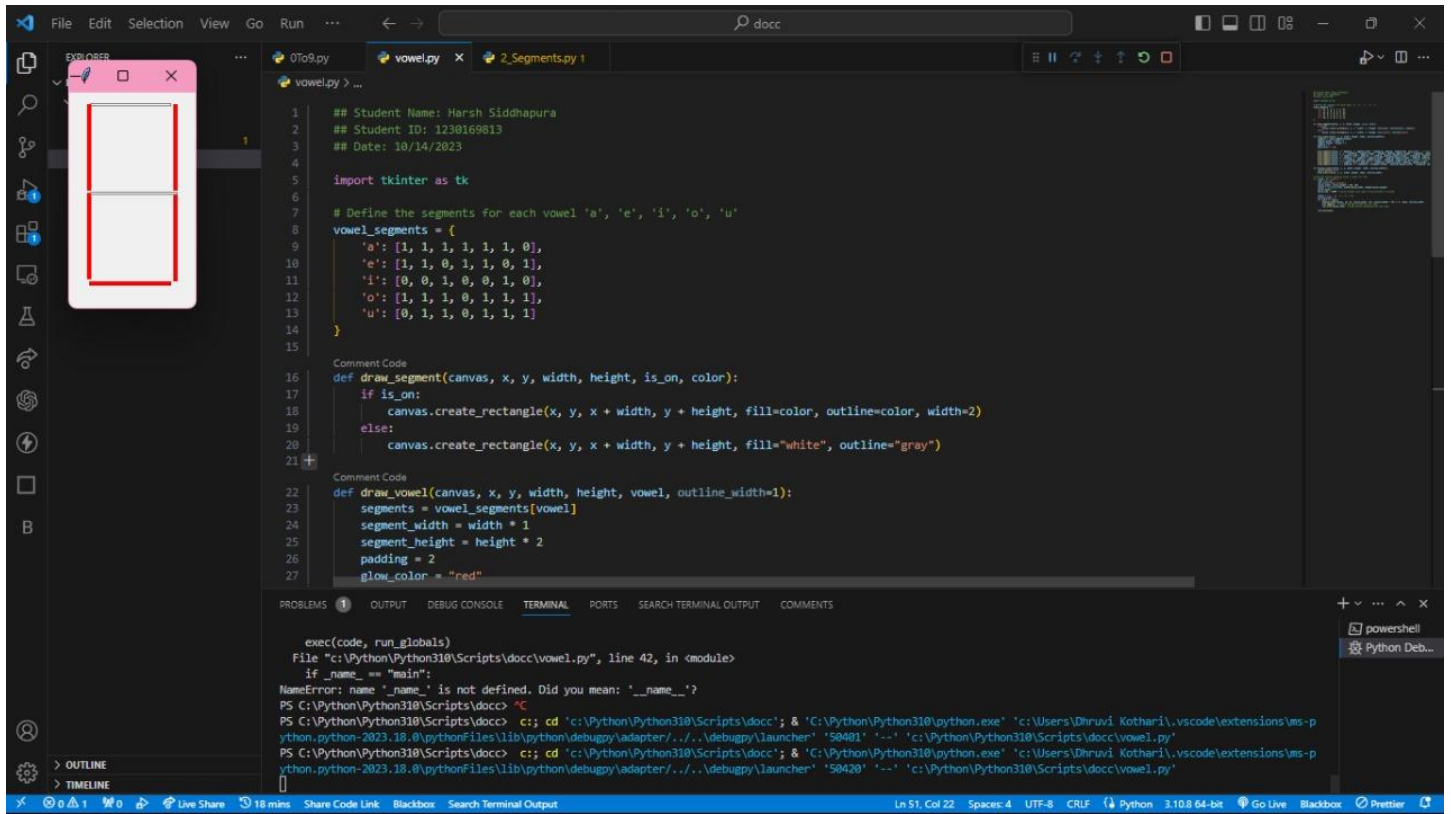
```

```

if __name__ == "__main__":
NameError: name 'name' is not defined. Did you mean: 'name__'?
PS C:\Python\Python310\Scripts\docc> cd "C:\Python\Python310\Scripts\docc"; & 'C:\Python\Python310\python.exe' 'C:\Users\Dhruvi Kothari\.vscode\extensions\ms-p
ython.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '58481' '--' 'C:\Python\Python310\Scripts\docc\vowel.py'
PS C:\Python\Python310\Scripts\docc> cd "C:\Python\Python310\Scripts\docc"; & 'C:\Python\Python310\python.exe' 'C:\Users\Dhruvi Kothari\.vscode\extensions\ms-p
ython.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '58420' '--' 'C:\Python\Python310\Scripts\docc\vowel.py'
PS C:\Python\Python310\Scripts\docc> cd "C:\Python\Python310\Scripts\docc"; & 'C:\Python\Python310\python.exe' 'C:\Users\Dhruvi Kothari\.vscode\extensions\ms-p
ython.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '58448' '--' 'C:\Python\Python310\Scripts\docc\vowel.py'

```



The image shows a VS Code editor with a Python script named `vowel.py` open. The script contains the following code:

```

1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 10/14/2023
4
5  import tkinter as tk
6
7  # Define the segments for each vowel 'a', 'e', 'i', 'o', 'u'
8  vowel_segments = {
9      'a': [1, 1, 1, 1, 1, 1, 0],
10     'e': [1, 1, 0, 1, 1, 0, 1],
11     'i': [0, 0, 1, 0, 0, 1, 0],
12     'o': [1, 1, 1, 0, 1, 1, 1],
13     'u': [0, 1, 1, 0, 1, 1, 1]
14 }
15
16 # Comment Code
17 def draw_segment(canvas, x, y, width, height, is_on, color):
18     if is_on:
19         canvas.create_rectangle(x, y, x + width, y + height, fill=color, outline=color, width=2)
20     else:
21         canvas.create_rectangle(x, y, x + width, y + height, fill="white", outline="gray")
22
23 # Comment Code
24 def draw_vowel(canvas, x, y, width, height, vowel, outline_width=1):
25     segments = vowel_segments[vowel]
26     segment_width = width * 1
27     segment_height = height * 2
28     padding = 2
29     glow_color = "red"

```

The terminal output shows the following error:

```

exec(code, run_globals)
File "c:\Python\Python310\Scripts\docc\vowel.py", line 42, in <module>
    if _name_ == "main":
NameError: name '_name_' is not defined. Did you mean: '___name___'?
PS C:\Python\Python310\Scripts\docc> c:\cd "c:\Python\Python310\Scripts\docc"; & "c:\Python\Python310\python.exe" "c:\Users\Dhruvi Kothari\vscode\extensions\ms-p
python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher" "58481" "--" "c:\Python\Python310\Scripts\docc\vowel.py"
PS C:\Python\Python310\Scripts\docc> c:\cd "c:\Python\Python310\Scripts\docc"; & "c:\Python\Python310\python.exe" "c:\Users\Dhruvi Kothari\vscode\extensions\ms-p
python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher" "58428" "--" "c:\Python\Python310\Scripts\docc\vowel.py"

```

A small window with a red border is visible in the top-left corner of the editor.

2 Digit - 7 Segment Display

```

1: 4: [1, 0, 1, 1, 0, 1, 1],
2: 5: [1, 0, 1, 1, 1, 1, 1],
3: 6: [1, 0, 1, 1, 1, 1, 1],
4: 7: [1, 1, 1, 0, 0, 0, 0],
5: 8: [1, 1, 1, 1, 1, 1, 1],
6: 9: [1, 1, 1, 1, 0, 1, 1]

def draw_segment(canvas, x, y, width, height, is_on, color):
    if is_on:
        canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill=color, outline=color, width=2)
    else:
        canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill="white", outline="gray")

def draw_digit(canvas, x, y, width, height, digit, outline_width=1):
    a, b, c, d, e, f, g = segments[digit]
    segment_width = width * 1
    segment_height = height * 2
    padding = 2
    glowColor = "red"

    # Draw each segment based on its state (on/off)
    draw_segment(canvas, x + padding, y, segment_width - 2 * padding, padding, a, glowColor) # Segment 'a'
    draw_segment(canvas, x, y + 2 * padding + segment_height - padding, padding, segment_height - 2 * padding, e, glowColor) # Segment 'e'
    draw_segment(canvas, x + segment_width + padding, y + padding, padding, segment_height - 2 * padding, b, glowColor) # Segment 'b'
    draw_segment(canvas, x + padding, y + padding + segment_height - padding, segment_height - padding, padding, g, glowColor) # Segment 'g'
    draw_segment(canvas, x, y + padding, padding, segment_height - 2 * padding, f, glowColor) # Segment 'f'
    draw_segment(canvas, x + segment_width + padding, y + 2 * padding + segment_height, padding, segment_height - 2 * padding, c, glowColor) # Segment 'c'

def display_two_digits(canvas, x, y, width, height, digit1, digit2, outline_width=1, spacing=10):
    canvas.delete("all")
    draw_digit(canvas, x, y, width, height, digit1, outline_width)
    draw_digit(canvas, x + width + spacing, y, width, height, digit2, outline_width)

if __name__ == "__main__":
    root = tk.Tk()
    root.title("Two-Digit Seven-Segment Display")

    canvas_width, canvas_height = 220, 200
    canvas = tk.Canvas(root, width=canvas_width, height=canvas_height)
    canvas.pack()

    delay_time = 500 # Delay between each digit in milliseconds (1 second)
    outline_width = 2

    for digit1 in range(10): # Display the first digit from 0 to 9
        for digit2 in range(10): # Display the second digit from 0 to 9
            display_two_digits(canvas, 10, 10, (canvas_width - 30) // 2, (canvas_height - 30) // 4, digit1, digit2, outline_width)
            root.update() # Update the GUI to show the new digits
            root.after(delay_time) # Delay before displaying the next digits

    root.mainloop()

```

File "C:\Python\Python310\Scripts\docc\2_Segments.py", line 44, in display_two_digits
 canvas.delete("all")
 File "C:\Python\Python310\lib\tkinter_init_.py", line 2852, in delete
 self.tk.call((self._w, 'delete') + args)
 _tkinter.TclError: invalid command name ".lcanvas"

```

def draw_segment(canvas, x, y, width, height, is_on, color):
    if is_on:
        canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill=color, outline=color, width=2)
    else:
        canvas.create_rectangle(x, y, x + width * 6, y + height * 10, fill="white", outline="gray")

def draw_digit(canvas, x, y, width, height, digit, outline_width=1):
    a, b, c, d, e, f, g = segments[digit]
    segment_width = width * 1
    segment_height = height * 2
    padding = 2
    glowColor = "red"

    # Draw each segment based on its state (on/off)
    draw_segment(canvas, x + padding, y, segment_width - 2 * padding, padding, a, glowColor) # Segment 'a'
    draw_segment(canvas, x, y + 2 * padding + segment_height - padding, padding, segment_height - 2 * padding, e, glowColor) # Segment 'e'
    draw_segment(canvas, x + segment_width + padding, y + padding, padding, segment_height - 2 * padding, b, glowColor) # Segment 'b'
    draw_segment(canvas, x + padding, y + padding + segment_height - padding, segment_height - padding, padding, g, glowColor) # Segment 'g'
    draw_segment(canvas, x, y + padding, padding, segment_height - 2 * padding, f, glowColor) # Segment 'f'
    draw_segment(canvas, x + segment_width + padding, y + 2 * padding + segment_height, padding, segment_height - 2 * padding, c, glowColor) # Segment 'c'

def display_two_digits(canvas, x, y, width, height, digit1, digit2, outline_width=1, spacing=10):
    canvas.delete("all")
    draw_digit(canvas, x, y, width, height, digit1, outline_width)
    draw_digit(canvas, x + width + spacing, y, width, height, digit2, outline_width)

if __name__ == "__main__":
    root = tk.Tk()
    root.title("Two-Digit Seven-Segment Display")

    canvas_width, canvas_height = 220, 200
    canvas = tk.Canvas(root, width=canvas_width, height=canvas_height)
    canvas.pack()

    delay_time = 500 # Delay between each digit in milliseconds (1 second)
    outline_width = 2

    for digit1 in range(10): # Display the first digit from 0 to 9
        for digit2 in range(10): # Display the second digit from 0 to 9
            display_two_digits(canvas, 10, 10, (canvas_width - 30) // 2, (canvas_height - 30) // 4, digit1, digit2, outline_width)
            root.update() # Update the GUI to show the new digits
            root.after(delay_time) # Delay before displaying the next digits

    root.mainloop()

```

File "C:\Python\Python310\Scripts\docc\2_Segments.py", line 62, in <module>
 display_two_digits(canvas, 10, 10, (canvas_width - 30) // 2, (canvas_height - 30) // 4, digit1, digit2, outline_width)
 File "C:\Python\Python310\Scripts\docc\2_Segments.py", line 44, in display_two_digits
 canvas.delete("all")
 File "C:\Python\Python310\lib\tkinter_init_.py", line 2852, in delete
 self.tk.call((self._w, 'delete') + args)
 _tkinter.TclError: invalid command name ".lcanvas"

The image shows a VS Code editor with a Python script named `2_Segments.py`. A small window titled "Two-Dig..." is overlaid on the editor, displaying a red-outlined 2x2 grid. The script defines a function `display_two_digits` that takes a canvas, coordinates, and two digits as input. It uses `draw_digit` to render each digit on the canvas. The main logic is in the `if __name__ == '__main__':` block, which creates a Tkinter window titled "Two-Digit Seven-Segment Display" and uses a loop to display all combinations of two digits from 0 to 9. The terminal at the bottom shows an error: `TclError: invalid command name ".lcanvas"`, indicating a problem with the Tkinter canvas widget.

```

def display_two_digits(canvas, x, y, width, height, digit1, digit2, outline_width=1, spacing=10):
    canvas.delete("all")
    draw_digit(canvas, x, y, width, height, digit1, outline_width)
    draw_digit(canvas, x + width + spacing, y, width, height, digit2, outline_width)

# Test the display sequence using a simple for loop
if __name__ == '__main__':
    root = tk.Tk()
    root.title("Two-Digit Seven-Segment Display")

    canvas_width, canvas_height = 220, 200
    canvas = tk.Canvas(root, width=canvas_width, height=canvas_height)
    canvas.pack()

    delay_time = 500 # Delay between each digit in milliseconds (1 second)
    outline_width = 2

    for digit1 in range(10): # Display the first digit from 0 to 9
        for digit2 in range(10): # Display the second digit from 0 to 9
            display_two_digits(canvas, 10, 10, (canvas_width - 30) // 2, (canvas_height - 30) // 4, digit1, digit2, outline_width)
            root.update() # Update the GUI to show the new digits
            root.after(delay_time) # Delay before displaying the next digits

    root.mainloop()

```

File "C:\Python\Python310\Scripts\docc\2_Segments.py", line 62, in <module>
display_two_digits(canvas, 10, 10, (canvas_width - 30) // 2, (canvas_height - 30) // 4, digit1, digit2, outline_width)
File "C:\Python\Python310\Scripts\docc\2_Segments.py", line 44, in display_two_digits
canvas.delete("all")
File "C:\Python\Python310\lib\tkinter_init_.py", line 2852, in delete
self.tk.call((self._w, 'delete') + args)
_tkinter.TclError: invalid command name ".lcanvas"
PS C:\Python\Python310\Scripts\docc> c:\cd "c:\Python\Python310\Scripts\docc"; & 'C:\Python\Python310\python.exe' 'c:\Users\Kothari\vscode\extensions\ms-p
ython.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '50514' '-.' 'c:\Python\Python310\Scripts\docc\2_Segments.py'