SEN4015 Advanced Programming with Python Term Project

Depression, Anxiety and Stress Scale Program

Group number: 22

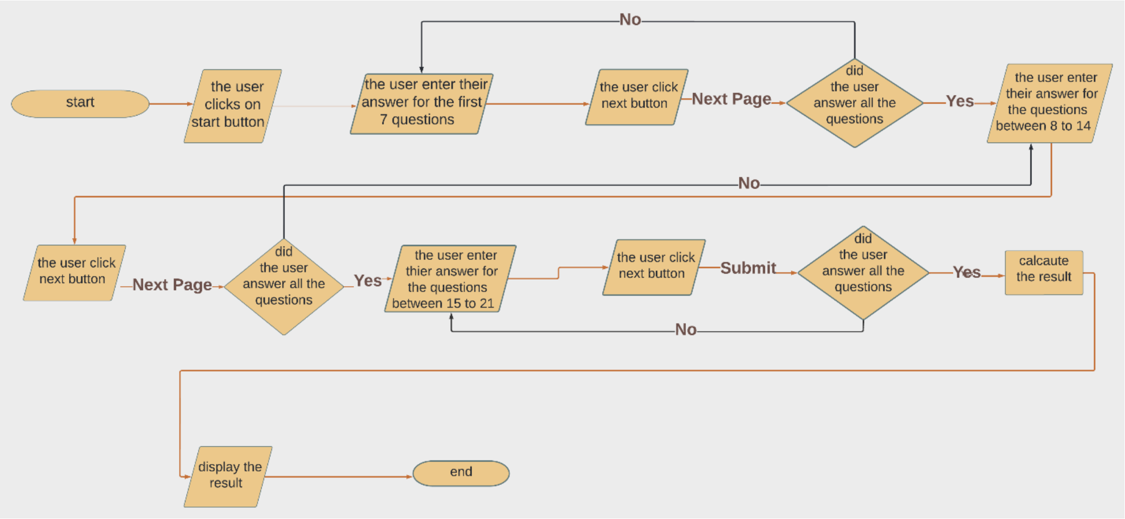
Student ID	Name-Surname	Responsiplites
1900812	Umniyah Sameer Haitham Abbood	Part of GUI & backend, report& slides
1900298	Charaf-Eddine M'rah	Part of backend, a small part of GUI
1900129	Mohanad El Masri	GUI implemnataion, class UML

The purpose of the project is to transfer a scientific psychology study which is “Depression, Anxiety, and Stress Scale” into a program. Our project is implemented using python and GUI using Tkinter toolkit.

The user will be asked to answer twenty-one questions to calculate their Depression, Anxiety, and Stress level.

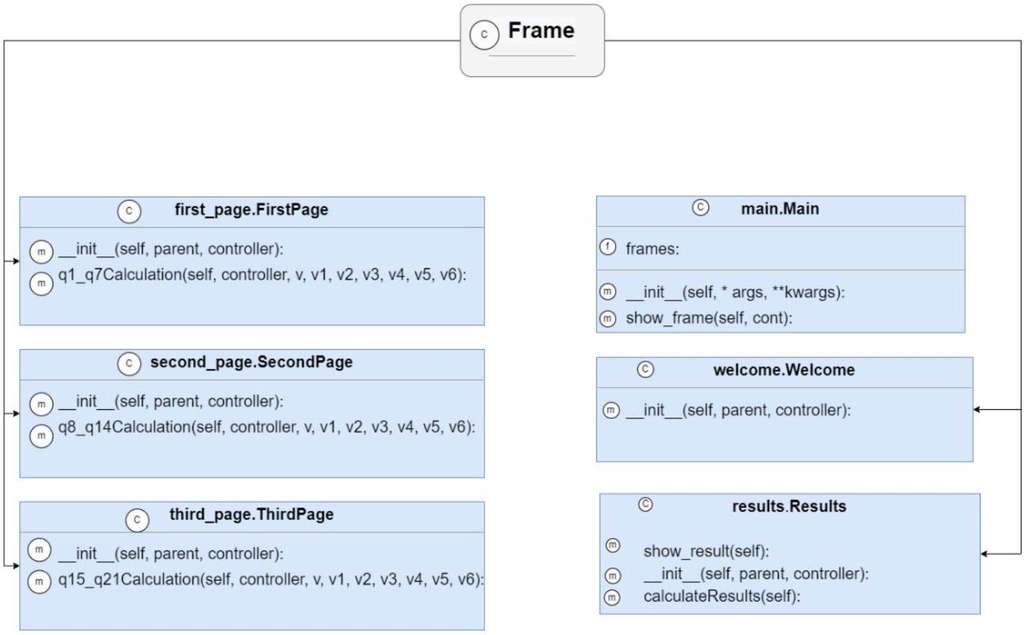
We will have three arrays store the answers to each question type and calculate the results and display the results.

1. Flow chart for DASS program



start
the user clicks on start button
the user enter their answer for the first 7 questions
the user click next button
Next Page
did the user answer all the questions
No
Yes
the user enter their answer for the questions between 8 to 14
No
the user click next button
Next Page
did the user answer all the questions
Yes
the user enter thier answer for the questions between 15 to 21
the user click next button
Submit
did the user answer all the questions
No
Yes
calcaute the result
display the result
end

2. UML Diagram for the code


c Frame
c first_page.FirstPage
m __init__(self, parent, controller):
m q1_q7Calculation(self, controller, v, v1, v2, v3, v4, v5, v6):
c second_page.SecondPage
m __init__(self, parent, controller):
m q8_q14Calculation(self, controller, v, v1, v2, v3, v4, v5, v6):
c third_page.ThirdPage
m __init__(self, parent, controller):
m q15_q21Calculation(self, controller, v, v1, v2, v3, v4, v5, v6):
c main.Main
f frames:
m __init__(self, * args, **kwargs):
m show_frame(self, cont):
c welcome.Welcome
m __init__(self, parent, controller):
c results.Results
m show_result(self):
m __init__(self, parent, controller):
m calculateResults(self):

3. Implementation

We have implemented five classes, each class in a .py file, and two more .py files

- class Main
- class Welcome
- class FirstPage
- class SecondPage
- class ThirdPage
- class Results
  - answers .py
  - constants .py

Our program will start executing with:

- ***class Main:*** in this class, we initialized our container where all other frames will be imported there, and use **mainloop ()** to run the Tkinter event loop, this class includes

    - **def __init__ (self, * args, **kwargs):** this is the initializer method where we have kept our other frames as a list, we added our frames of the other pages to the list by using for loop. Then we displayed them by using **show_frame ()** function
    - **def show_frame (self, cont):** this function aims to show the frames one by one, on top of each other using **tkraise ()** method
- ***answers .py****:* it contains three arrays which are (stressArray, depressionArray, anxiteyArray).
    - **def calculateSum(arr):**

IPO chart for the def calculateSum(arr):		
İnput	**processing**	**output**
İt takes an array as input	Sum all the items in the array	Return the summation multiped by two

- ***constants .py****:* it has a built-in dictionary data type which includes the options of the answer which we will be passed each time to the Radio buttons using for loop
- ***class Welcome****:* this is the first page (welcome page) that will be displayed.
- **def __init__ (self, parent, controller):** this is the initializer method where we initialized the title of our frame, label, and start button using Tkinter. When the user clicks on the start button it opens the "First Page" of questions, therefore we imported *FirstPage class* to welcome class
- ***class FirstPage (tk. Frame)****:* this is the second page that will be displayed after clicking the start button on the welcome page. It has the first seven questions out of twenty-one questions
    - **def __init__ (self, parent, controller):** this is the initializer method where we initialized seven labels and 7 Radio buttons, we initialized Radio Button using for loop getting the values of the

Radio button from **constants .py.** And one button “next page”, when the user clicks that button, the selected value in the Radio buttons will be calculated using command lambda using **def q1_q7Calculation ()**

- **def q1_q7Calculation (self, controller, v, v1, v2, v3, v4, v5, v6):**

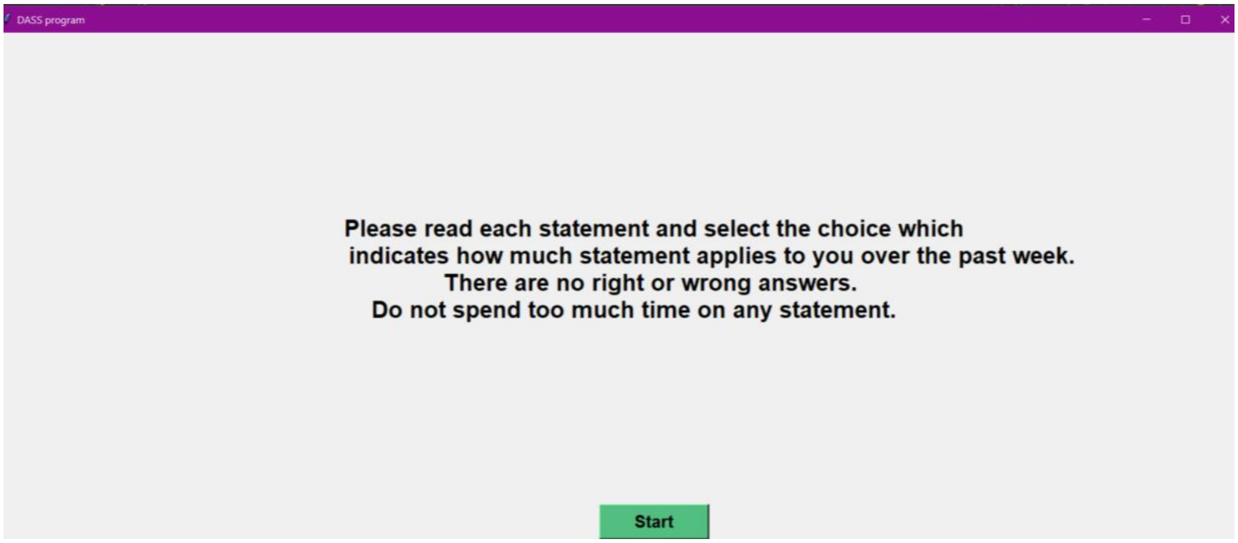
IPO chart for the def q1_q7Calculation		
İnput	**processing**	**output**
values of the Radio buttons which are seven	The function gets the answer to each question from the Radio buttons selected value and appends it to its corresponding array from answers .py. And it checks if the user left any Radio buttons empty	Appended value to the value stressArray, anxietyArray, depressionArray. If there are any empty values, it displays text “please answer all the questions” After the user answer all the questions and shows the second page frame which is the third page will be displayed

- ***class SecondPage (tk. Frame):*** this is the third page that will be displayed after clicking the next button on the first page. It has the seven questions, from 8 to 14 out of twenty-one questions
  - **def __init__ (self, parent, controller):** same exact implementation with the FirstPage class
  - **def q8_q14Calculation (self, controller, v, v1, v2, v3, v4, v5, v6):** same implementation with **def q1_q7Calculation ()** the only difference is that we will get and append the question from 8-14

to the corresponding array, and when the user clicks on the next page button the third page farm which is the fourth page will be displayed.

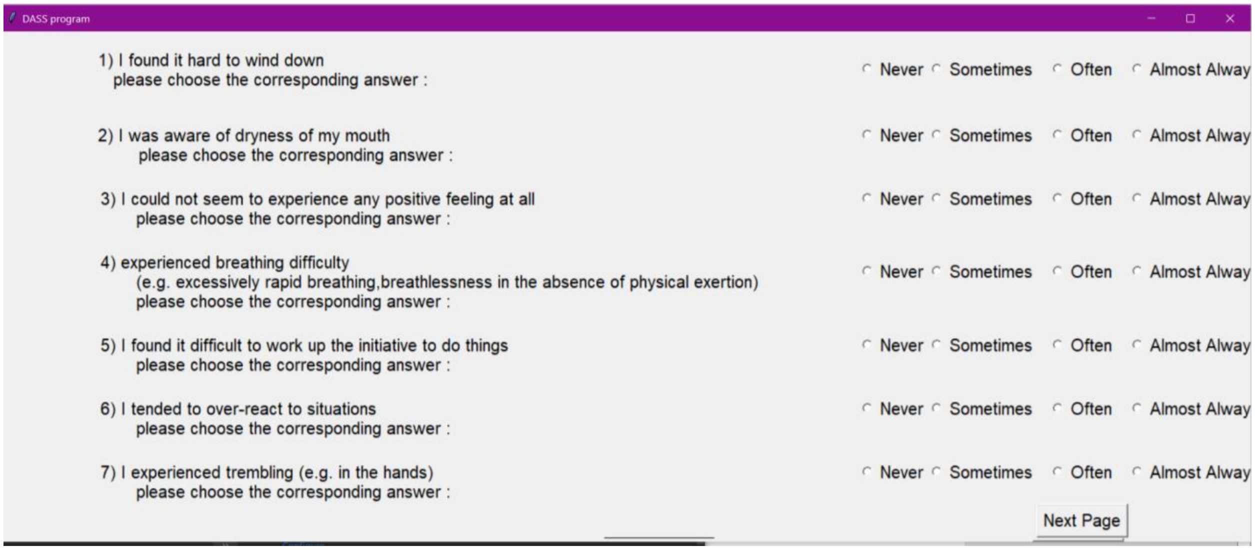
- *class ThirdPage (tk. Frame):* this is the fourth page that will be displayed after clicking the next button on the second page. It has the seven questions, from 15 to 21 questions
    - **def __init__ (self, parent, controller)*:*** same exact implementation with the *FirstPage* class and *SecondPage class*
    - **def q15_q21Calculation (self, controller, v, v1, v2, v3, v4, v5, v6):** same implementation with **def q1_q7Calculation ()** and **def q8_q14Calculation ()** the only difference is that we will get and append the question from 15-21 to the corresponding array, and when the user clicks on the next page button the Results frame which is the fifth page will be displayed.
- *class Results (tk. Frame):* this is the fifth page that will be displayed after clicking the next button on the ThirdPage. It has a button “get result” when the user clicks the button the results show up
    - **def __init__ (self, parent, controller):** this is the initializer method where we initialized one label and one button which is used to display the results using command lambda **show_result ()**
    - **def calculateResults(self):** this function calculates the sum of each array using **calculateSum ()** function from ***answers .py*** and makes the decision on which value should be displayed as a result and returns 3 string values, a result for (depression, anxiety, and stress)
    - **def show_result(self):** it gets three string values from **calculateResults ()** and assigns the value to result Label to display the results texts.

4. Program screenshots



Please read each statement and select the choice which
indicates how much statement applies to you over the past week.
There are no right or wrong answers.
Do not spend too much time on any statement.

Pic (1) Welcome page



1) I found it hard to wind down
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Alway

2) I was aware of dryness of my mouth
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Alway

3) I could not seem to experience any positive feeling at all
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Alway

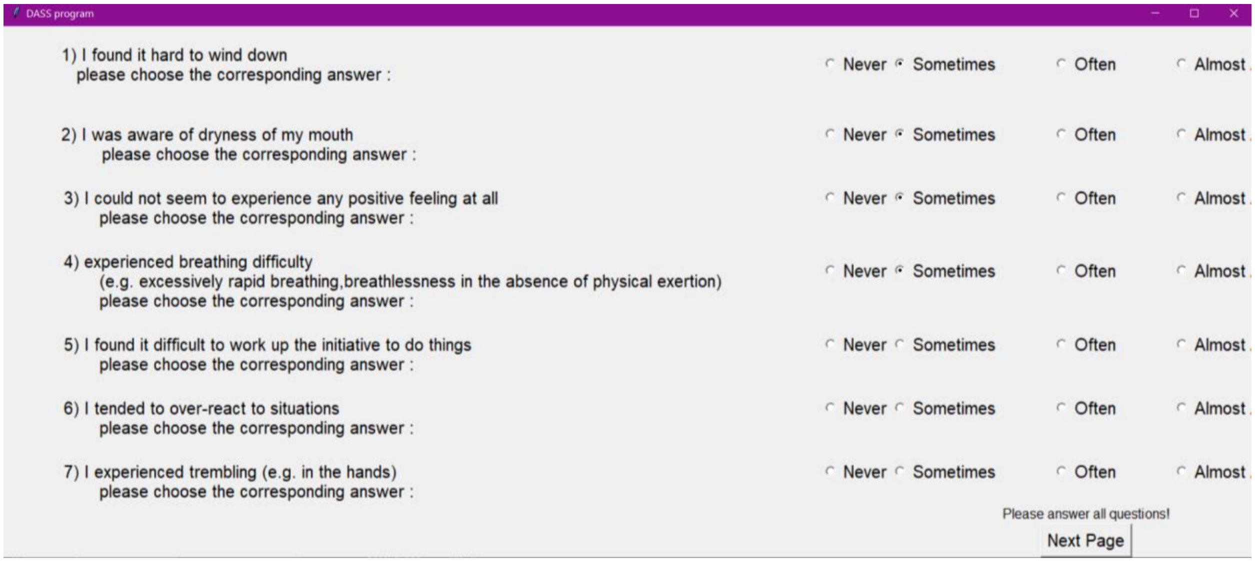
4) experienced breathing difficulty
(e.g. excessively rapid breathing,breathlessness in the absence of physical exertion)
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Alway

5) I found it difficult to work up the initiative to do things
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Alway

6) I tended to over-react to situations
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Alway

7) I experienced trembling (e.g. in the hands)
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Alway

Pic (2) First page



DASS program

1) I found it hard to wind down
please choose the corresponding answer :
○ Never ● Sometimes ○ Often ○ Almost

2) I was aware of dryness of my mouth
please choose the corresponding answer :
○ Never ● Sometimes ○ Often ○ Almost

3) I could not seem to experience any positive feeling at all
please choose the corresponding answer :
○ Never ● Sometimes ○ Often ○ Almost

4) experienced breathing difficulty
(e.g. excessively rapid breathing,breathlessness in the absence of physical exertion)
please choose the corresponding answer :
○ Never ● Sometimes ○ Often ○ Almost

5) I found it difficult to work up the initiative to do things
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost

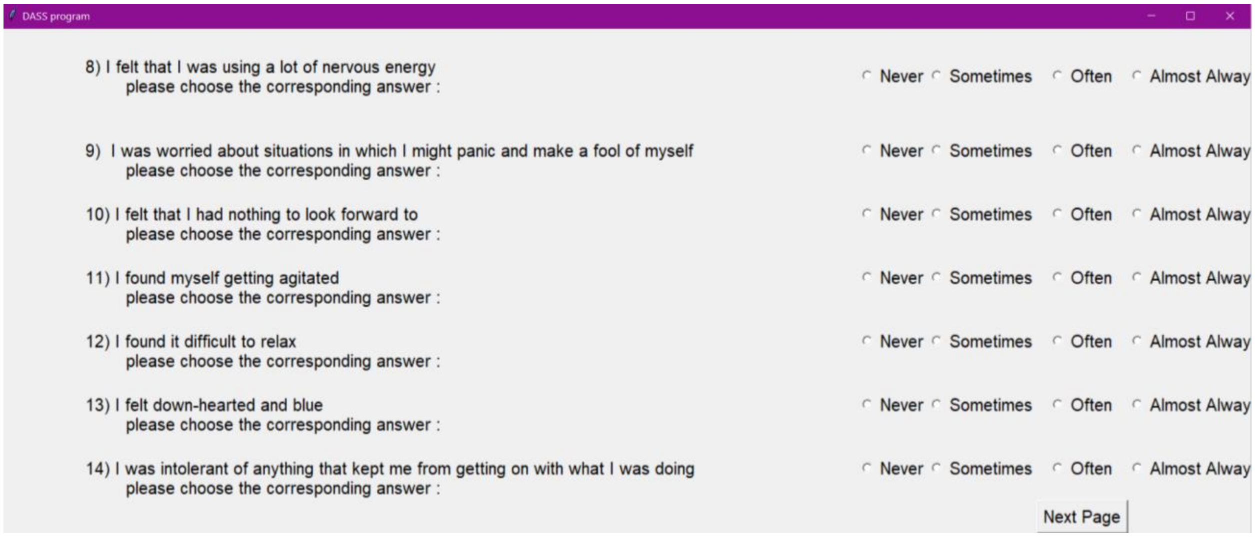
6) I tended to over-react to situations
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost

7) I experienced trembling (e.g. in the hands)
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost

Please answer all questions!

Pic (3) First page- if the user did not answer all the questions,

"Please answer all the questions" message will pop up and will not let the user go to the next page unless all the questions are answered



DASS program

8) I felt that I was using a lot of nervous energy
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost Alway

9) I was worried about situations in which I might panic and make a fool of myself
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost Alway

10) I felt that I had nothing to look forward to
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost Alway

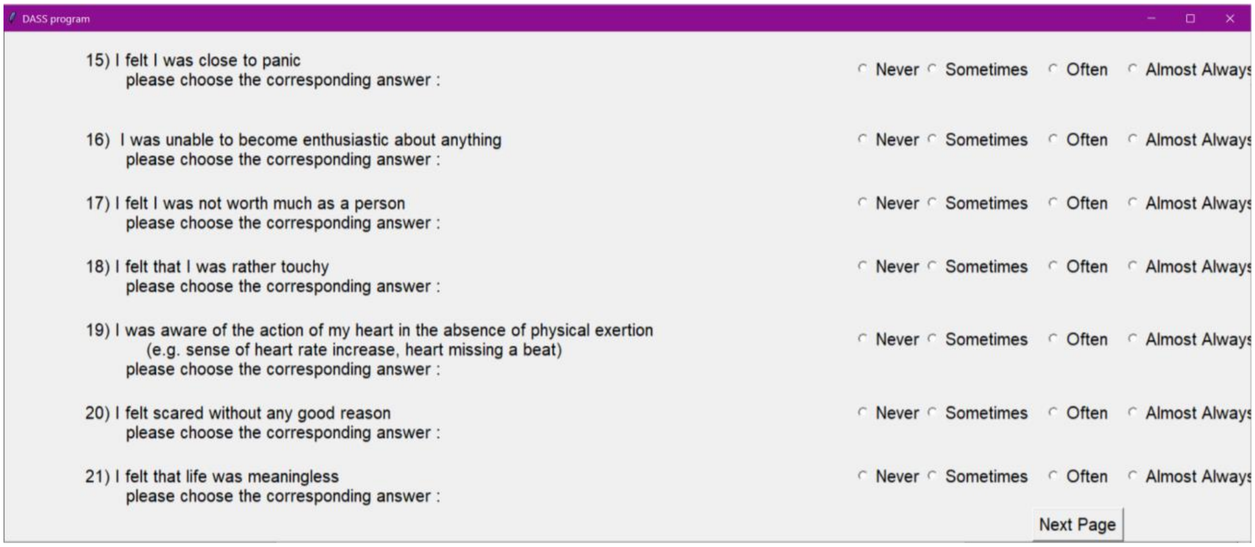
11) I found myself getting agitated
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost Alway

12) I found it difficult to relax
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost Alway

13) I felt down-hearted and blue
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost Alway

14) I was intolerant of anything that kept me from getting on with what I was doing
please choose the corresponding answer :
○ Never ○ Sometimes ○ Often ○ Almost Alway

Pic (4) Second page



15) I felt I was close to panic
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Always

16) I was unable to become enthusiastic about anything
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Always

17) I felt I was not worth much as a person
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Always

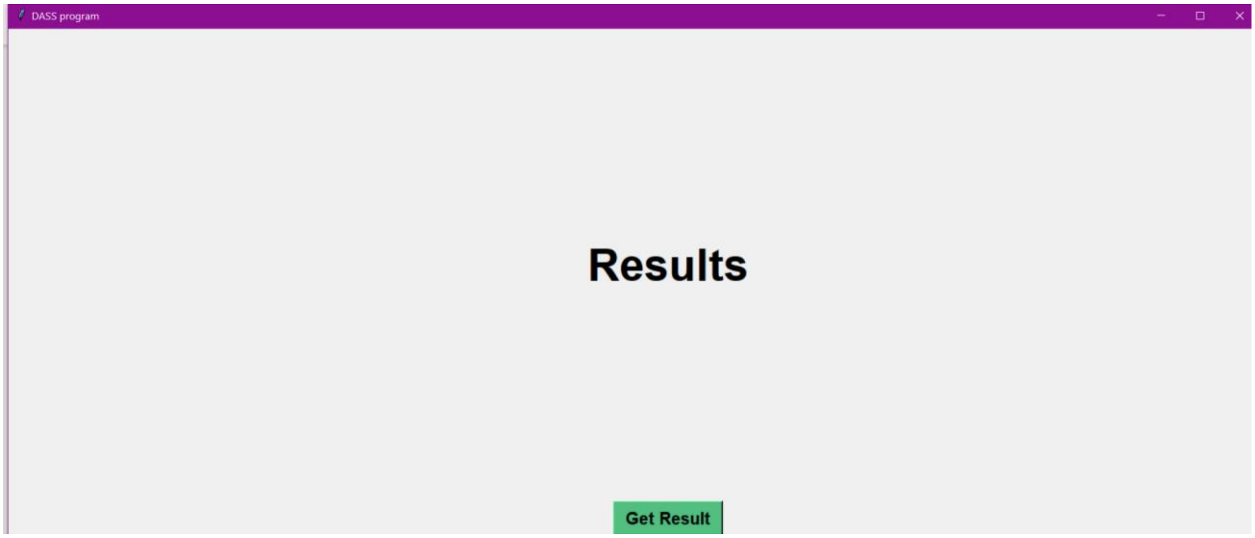
18) I felt that I was rather touchy
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Always

19) I was aware of the action of my heart in the absence of physical exertion
(e.g. sense of heart rate increase, heart missing a beat)
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Always

20) I felt scared without any good reason
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Always

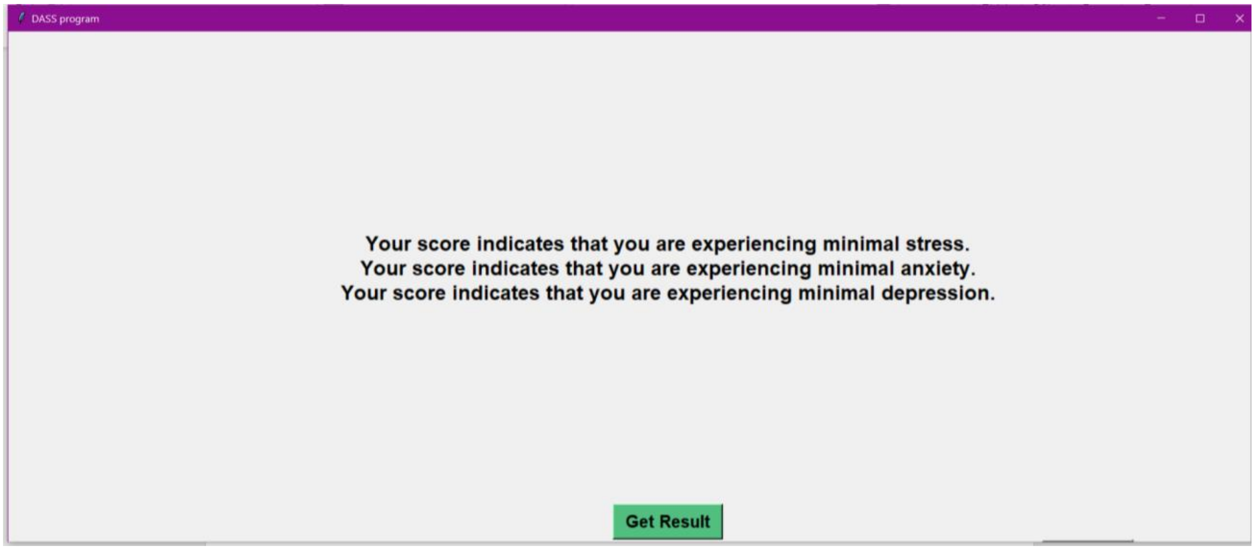
21) I felt that life was meaningless
please choose the corresponding answer : ○ Never ○ Sometimes ○ Often ○ Almost Always

Pic (5) Third page



Results

Pic (6-A) result page before clicking the button



Your score indicates that you are experiencing minimal stress.
Your score indicates that you are experiencing minimal anxiety.
Your score indicates that you are experiencing minimal depression.

Pic (6-B) result page after clicking the button

❖ This project contains the following concepts that we had learn throughout SEN4015 such as:

- Decision Structures (if, if-elif-else Statements)
- Repetition Structures (for Loop: a Count-Controlled Loop)
- Functions
- List & Dictionary
- OOP concepts such as class definition, initializer methods, importing other classes, *args, **kwargs

Main.py

```
import tkinter as tk
from welcome import Welcome
from first_page import FirstPage
from second_page import SecondPage
from third_page import ThirdPage
from results import Results
class Main(tk.Tk):
    def __init__(self, * args, **kwargs):
        tk.Tk.__init__(self, * args, **kwargs)
        container = tk.Frame(self)

        container.pack(fill="both", expand=True)
        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)

        self.frames = {}
        for F in (Welcome, FirstPage, SecondPage, ThirdPage, Results):
            frame = F(container, self)
            self.frames[F] = frame
            frame.grid(row=4, column=5)

        self.show_frame(Welcome)

    def show_frame(self, cont):
      frame = self.frames[cont]
      frame.tkraise()

app= Main()
app.mainloop()
```

welcome.py

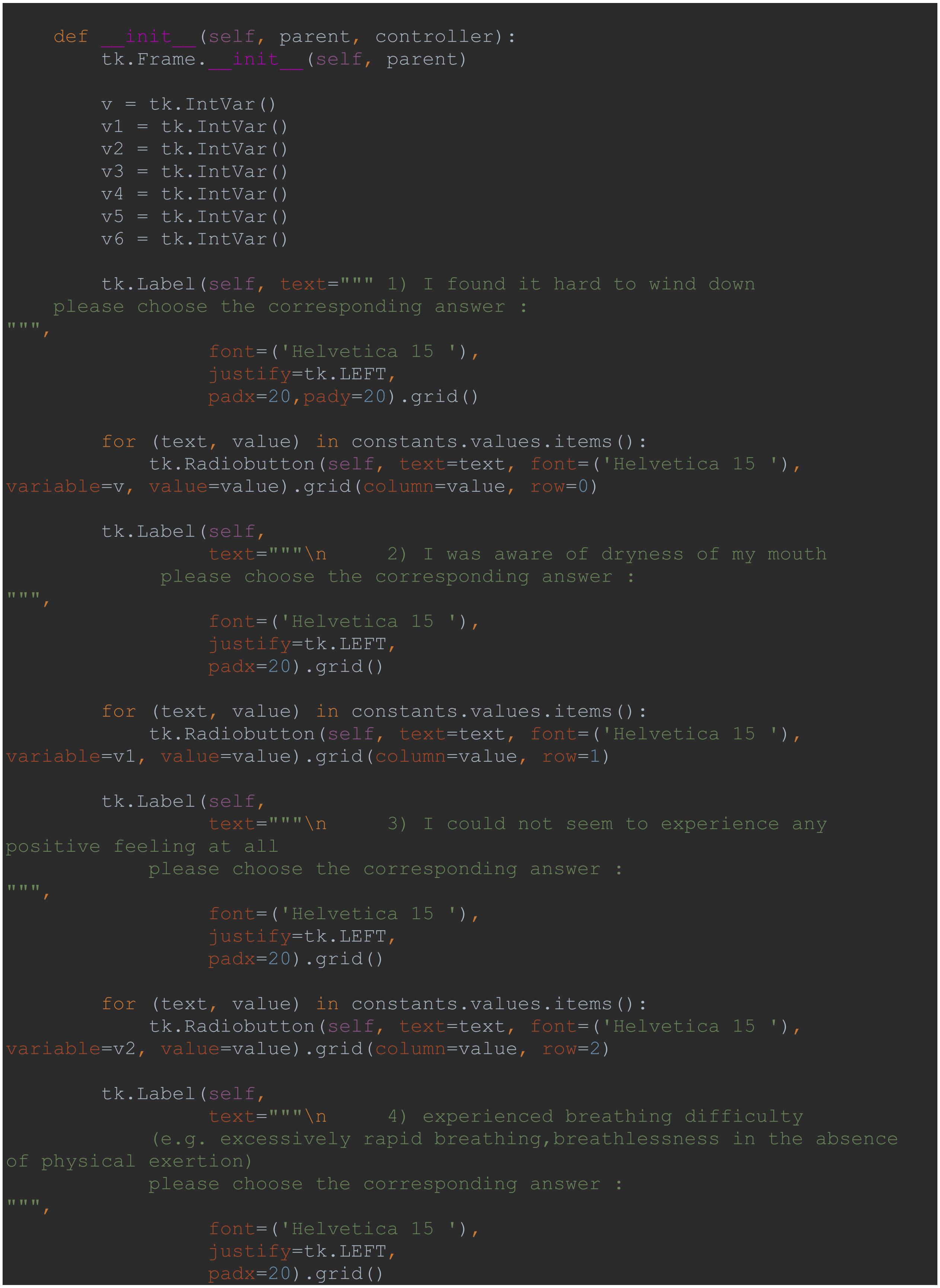
```
import tkinter as tk
from first_page import FirstPage
class Welcome(tk.Frame):
     def __init__(self, parent, controller):
         tk.Frame.__init__(self, parent) #the start is the parent
         self.winfo_toplevel().title("DASS program")
         label = tk.Label(self, text='''Please read each statement and select
the choice which
                 indicates how much statement applies to you over the past
week.
There are no right or wrong answers.
         Do not spend too much time on any statement.
''',font=('Helvetica 20 bold'))
         label.pack(padx=280, pady=210)
         buttonStart = tk.Button(self, text="Start", font=('Helvetica 15
bold'),

command=lambda:controller.show_frame(FirstPage) ,bg="#52BE80", height=0,
width=10)
         buttonStart.pack()
```

constants.py

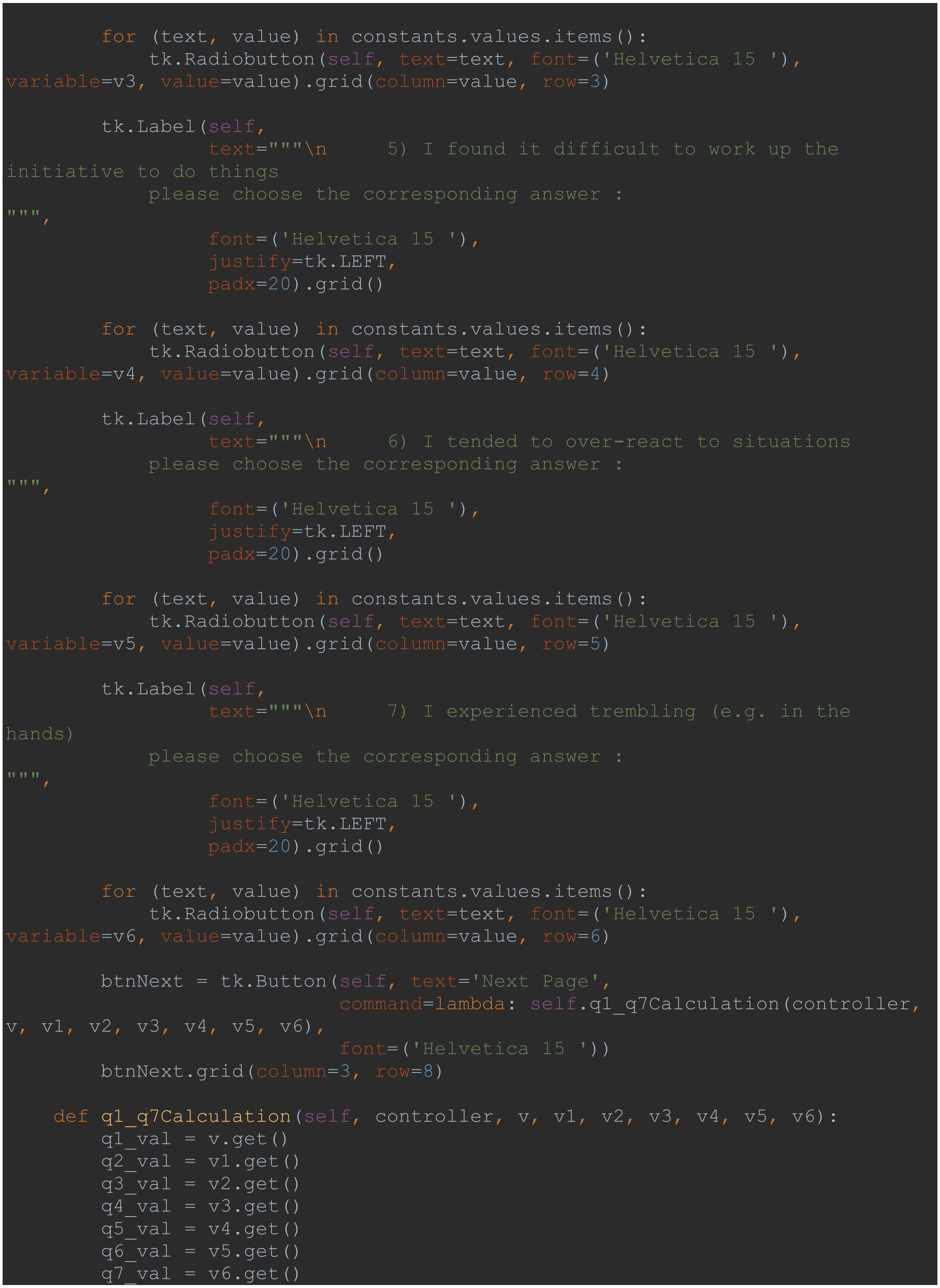
```
values = {"Never": "1",
          "Sometimes": "2",
          "Often": "3",
          "Almost Always": "4"}
```

first_page.py

```
import tkinter as tk
from second_page import SecondPage
import constants
import answers

class FirstPage(tk.Frame):
```



```
    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)

        v = tk.IntVar()
        v1 = tk.IntVar()
        v2 = tk.IntVar()
        v3 = tk.IntVar()
        v4 = tk.IntVar()
        v5 = tk.IntVar()
        v6 = tk.IntVar()

        tk.Label(self, text=""" 1) I found it hard to wind down
    please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20,pady=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v, value=value).grid(column=value, row=0)

        tk.Label(self,
                 text="""\n     2) I was aware of dryness of my mouth
             please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v1, value=value).grid(column=value, row=1)

        tk.Label(self,
                 text="""\n     3) I could not seem to experience any
positive feeling at all
            please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v2, value=value).grid(column=value, row=2)

        tk.Label(self,
                 text="""\n     4) experienced breathing difficulty
            (e.g. excessively rapid breathing,breathlessness in the absence
of physical exertion)
            please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()
```



```
        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v3, value=value).grid(column=value, row=3)

        tk.Label(self,
                 text="""\n     5) I found it difficult to work up the
initiative to do things
            please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v4, value=value).grid(column=value, row=4)

        tk.Label(self,
                 text="""\n     6) I tended to over-react to situations
            please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v5, value=value).grid(column=value, row=5)

        tk.Label(self,
                 text="""\n     7) I experienced trembling (e.g. in the
hands)
            please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v6, value=value).grid(column=value, row=6)

        btnNext = tk.Button(self, text='Next Page',
                            command=lambda: self.q1_q7Calculation(controller,
v, v1, v2, v3, v4, v5, v6),
                            font=('Helvetica 15 '))
        btnNext.grid(column=3, row=8)

    def q1_q7Calculation(self, controller, v, v1, v2, v3, v4, v5, v6):
        q1_val = v.get()
        q2_val = v1.get()
        q3_val = v2.get()
        q4_val = v3.get()
        q5_val = v4.get()
        q6_val = v5.get()
        q7_val = v6.get()
        answers.stressArray.append(q1_val)
        answers.anxietyArray.append(q2_val)
        answers.depressionArray.append(q3_val)
        answers.anxietyArray.append(q4_val)
        answers.depressionArray.append(q5_val)
        answers.stressArray.append(q6_val)
        answers.anxietyArray.append(q7_val)
        if q1_val > 0 and q2_val > 0 and q3_val > 0 and q4_val > 0 and q5_val
> 0 and q6_val > 0 and q7_val > 0:
            controller.show_frame(SecondPage)
        else:
            errorLabel = tk.Label(self, text='Please answer all questions!',
font=('Helvetica 12'))
            errorLabel.grid(row=7, column=3)
```

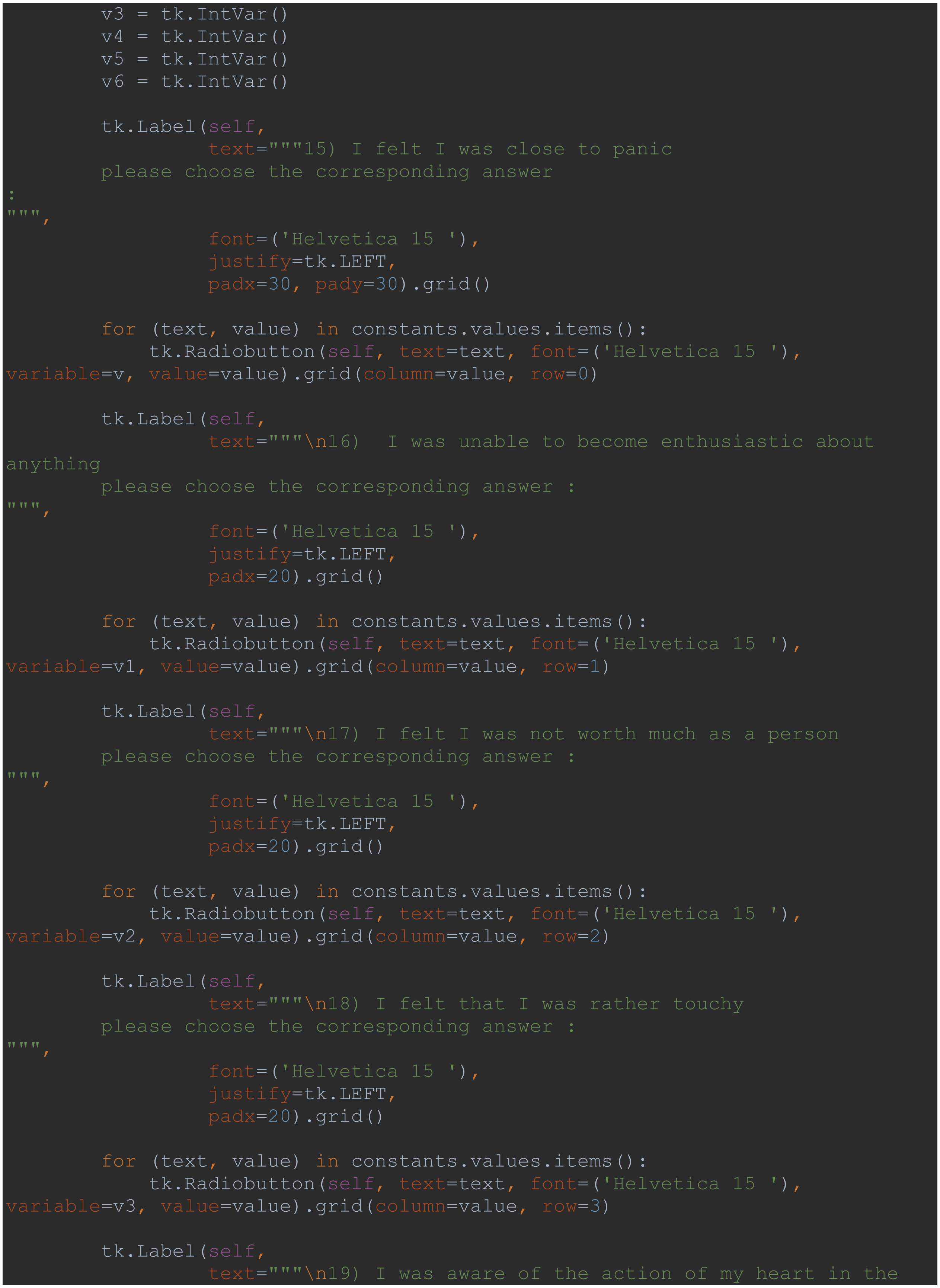
second_page.py

```
import tkinter as tk

from third_page import ThirdPage
import constants
import answers

class SecondPage(tk.Frame):
    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        v = tk.IntVar()
        v1 = tk.IntVar()
        v2 = tk.IntVar()
        v3 = tk.IntVar()
        v4 = tk.IntVar()
        v5 = tk.IntVar()
        v6 = tk.IntVar()

        tk.Label(self, text="""8) I felt that I was using a lot of nervous
energy
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=35, pady=30).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v, value=value).grid(column=value, row=0)

        tk.Label(self,
                 text="""\n9)  I was worried about situations in which I
might panic and make a fool of myself
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v1, value=value).grid(column=value, row=1)

        tk.Label(self,
                 text="""\n10) I felt that I had nothing to look forward to
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v2, value=value).grid(column=value, row=2)

        tk.Label(self,
                 text="""\n11) I found myself getting agitated
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v3, value=value).grid(column=value, row=3)

        tk.Label(self,
                 text="""\n12) I found it difficult to relax
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v4, value=value).grid(column=value, row=4)

        tk.Label(self,
                 text="""\n13) I felt down-hearted and blue
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v5, value=value).grid(column=value, row=5)

        tk.Label(self,
                 text="""\n14) I was intolerant of anything that kept me from
getting on with what I was doing
```

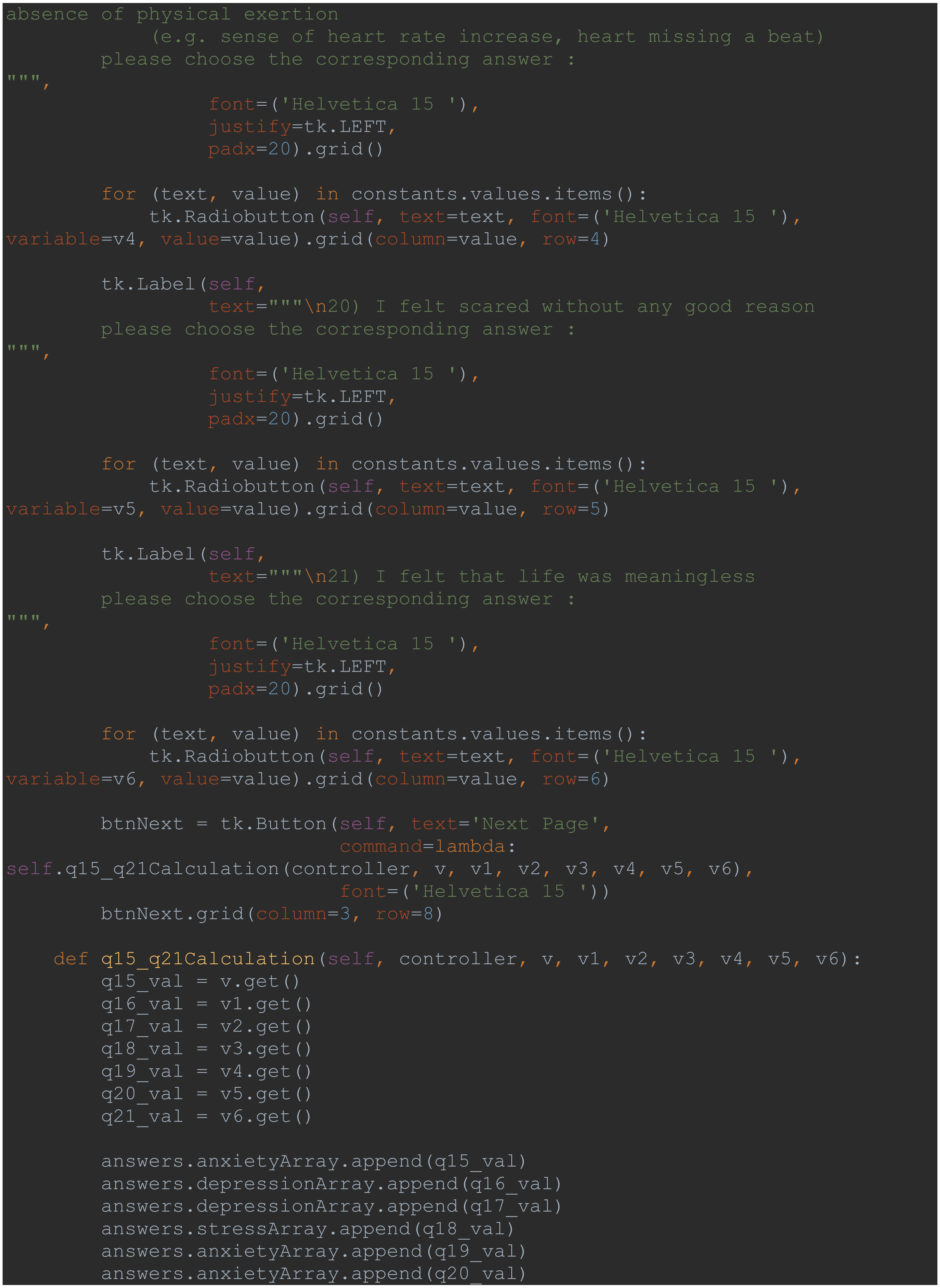
```
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v6, value=value).grid(column=value, row=6)

        btnNext = tk.Button(self, text='Next Page',
                            command=lambda:
self.q8_q14Calculation(controller, v, v1, v2, v3, v4, v5, v6),
                            font=('Helvetica 15 '))
        btnNext.grid(column=3, row=8)

    def q8_q14Calculation(self, controller, v, v1, v2, v3, v4, v5, v6):
        q8_val = v.get()
        q9_val = v1.get()
        q10_val = v2.get()
        q11_val = v3.get()
        q12_val = v4.get()
        q13_val = v5.get()
        q14_val = v6.get()

        answers.stressArray.append(q8_val)
        answers.anxietyArray.append(q9_val)
        answers.depressionArray.append(q10_val)
        answers.stressArray.append(q11_val)
        answers.stressArray.append(q12_val)
        answers.depressionArray.append(q13_val)
        answers.stressArray.append(q14_val)

        if q8_val > 0 and q9_val > 0 and q10_val > 0 and q11_val > 0 and
q12_val > 0 and q13_val > 0 and q14_val > 0:
          controller.show_frame(ThirdPage)
        else:
          errorLabel = tk.Label(self, text='Please answer all
questions!',font=('Helvetica 12'))
          errorLabel.grid(row=7, column= 3)
```

third_page.py

```
import tkinter as tk

from results import Results
import constants
import answers
class ThirdPage(tk.Frame):
    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        v = tk.IntVar()
        v1 = tk.IntVar()
        v2 = tk.IntVar()
```



```
        v3 = tk.IntVar()
        v4 = tk.IntVar()
        v5 = tk.IntVar()
        v6 = tk.IntVar()

        tk.Label(self,
                 text="""15) I felt I was close to panic
        please choose the corresponding answer
:
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=30, pady=30).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v, value=value).grid(column=value, row=0)

        tk.Label(self,
                 text="""\n16)  I was unable to become enthusiastic about
anything
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v1, value=value).grid(column=value, row=1)

        tk.Label(self,
                 text="""\n17) I felt I was not worth much as a person
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v2, value=value).grid(column=value, row=2)

        tk.Label(self,
                 text="""\n18) I felt that I was rather touchy
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v3, value=value).grid(column=value, row=3)

        tk.Label(self,
                 text="""\n19) I was aware of the action of my heart in the
```



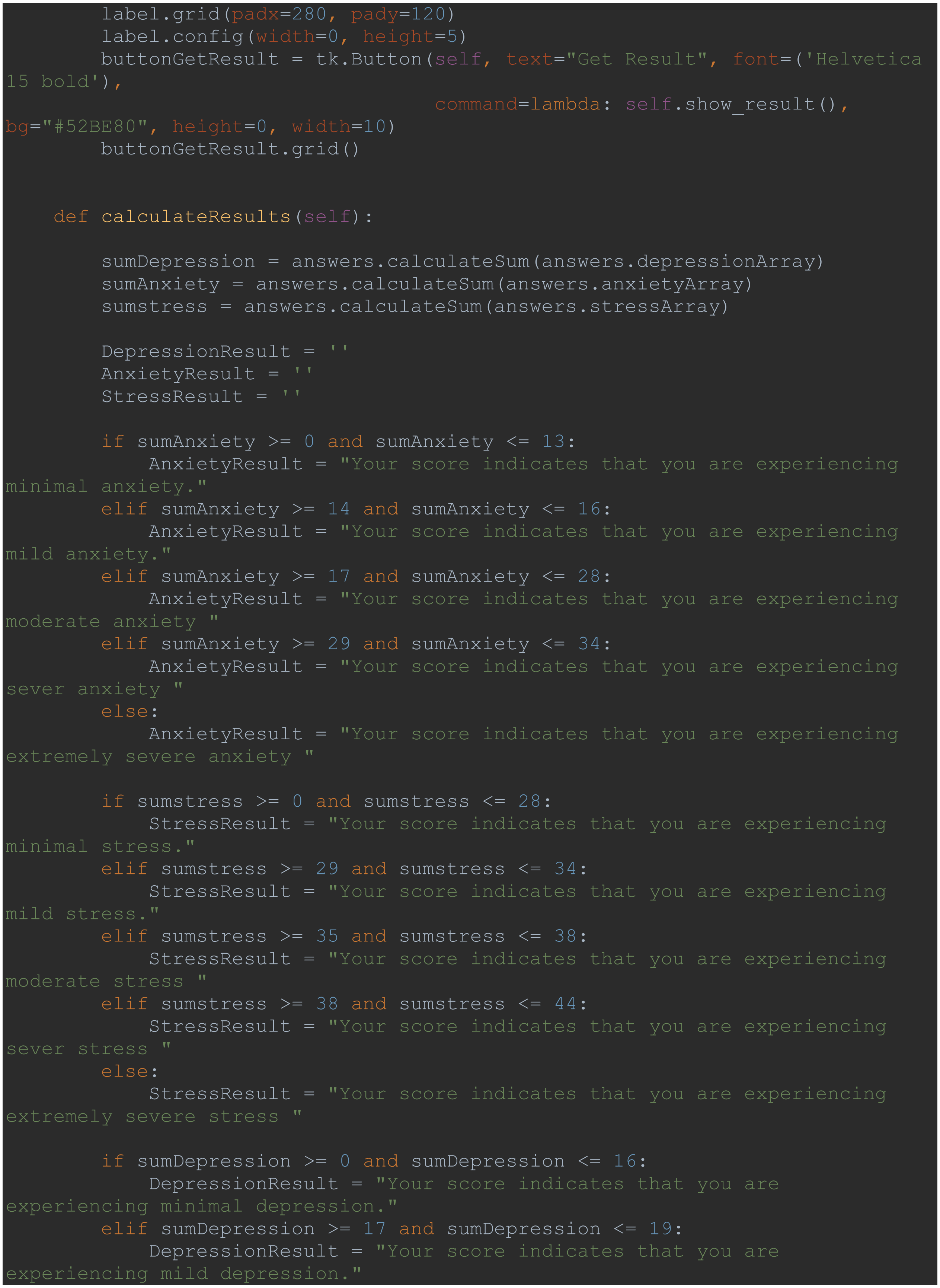
```
absence of physical exertion
            (e.g. sense of heart rate increase, heart missing a beat)
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v4, value=value).grid(column=value, row=4)

        tk.Label(self,
                 text="""\n20) I felt scared without any good reason
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v5, value=value).grid(column=value, row=5)

        tk.Label(self,
                 text="""\n21) I felt that life was meaningless
        please choose the corresponding answer :
""",
                 font=('Helvetica 15 '),
                 justify=tk.LEFT,
                 padx=20).grid()

        for (text, value) in constants.values.items():
            tk.Radiobutton(self, text=text, font=('Helvetica 15 '),
variable=v6, value=value).grid(column=value, row=6)

        btnNext = tk.Button(self, text='Next Page',
                            command=lambda:
self.q15_q21Calculation(controller, v, v1, v2, v3, v4, v5, v6),
                            font=('Helvetica 15 '))
        btnNext.grid(column=3, row=8)

    def q15_q21Calculation(self, controller, v, v1, v2, v3, v4, v5, v6):
        q15_val = v.get()
        q16_val = v1.get()
        q17_val = v2.get()
        q18_val = v3.get()
        q19_val = v4.get()
        q20_val = v5.get()
        q21_val = v6.get()

        answers.anxietyArray.append(q15_val)
        answers.depressionArray.append(q16_val)
        answers.depressionArray.append(q17_val)
        answers.stressArray.append(q18_val)
        answers.anxietyArray.append(q19_val)
        answers.anxietyArray.append(q20_val)
        answers.depressionArray.append(q21_val)

        if q15_val > 0 and q16_val > 0 and q17_val > 0 and q18_val > 0 and
q19_val > 0 and q20_val > 0 and q21_val > 0:
          controller.show_frame(Results)
        else:
          errorLabel = tk.Label(self, text='Please answer all
questions!',font=('Helvetica 12'))
          errorLabel.grid(row=7, column= 3)
```

answers.py

```
import array as arr

stressArray = arr.array('i')
depressionArray = arr.array('i')
anxietyArray = arr.array('i')

def calculateSum(arr):
  sum = 0
  for i in range(0, len(arr)):
    sum = sum + arr[i]
  return sum*2
```

results.py

```
import tkinter as tk
import welcome
import answers

class Results(tk.Frame):
    def show_result(self):

        depressionResult, anxietyResult, stressResult =
self.calculateResults()
        result = stressResult + "\n" + anxietyResult + "\n" +
depressionResult

        resultLabel = tk.Label(self, text=result,font=('Helvetica 18 bold'))

        resultLabel.grid(row=0, column= 0)
        result_shown = True

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        tk.Frame.configure(self)

        label = tk.Label(self,
                         text="\nResults\n
",
                         font=('Helvetica 40 bold'))
```



```
        label.grid(padx=280, pady=120)
        label.config(width=0, height=5)
        buttonGetResult = tk.Button(self, text="Get Result", font=('Helvetica
15 bold'),
                                    command=lambda: self.show_result(),
bg="#52BE80", height=0, width=10)
        buttonGetResult.grid()


    def calculateResults(self):

        sumDepression = answers.calculateSum(answers.depressionArray)
        sumAnxiety = answers.calculateSum(answers.anxietyArray)
        sumstress = answers.calculateSum(answers.stressArray)

        DepressionResult = ''
        AnxietyResult = ''
        StressResult = ''

        if sumAnxiety >= 0 and sumAnxiety <= 13:
            AnxietyResult = "Your score indicates that you are experiencing
minimal anxiety."
        elif sumAnxiety >= 14 and sumAnxiety <= 16:
            AnxietyResult = "Your score indicates that you are experiencing
mild anxiety."
        elif sumAnxiety >= 17 and sumAnxiety <= 28:
            AnxietyResult = "Your score indicates that you are experiencing
moderate anxiety "
        elif sumAnxiety >= 29 and sumAnxiety <= 34:
            AnxietyResult = "Your score indicates that you are experiencing
sever anxiety "
        else:
            AnxietyResult = "Your score indicates that you are experiencing
extremely severe anxiety "

        if sumstress >= 0 and sumstress <= 28:
            StressResult = "Your score indicates that you are experiencing
minimal stress."
        elif sumstress >= 29 and sumstress <= 34:
            StressResult = "Your score indicates that you are experiencing
mild stress."
        elif sumstress >= 35 and sumstress <= 38:
            StressResult = "Your score indicates that you are experiencing
moderate stress "
        elif sumstress >= 38 and sumstress <= 44:
            StressResult = "Your score indicates that you are experiencing
sever stress "
        else:
            StressResult = "Your score indicates that you are experiencing
extremely severe stress "

        if sumDepression >= 0 and sumDepression <= 16:
            DepressionResult = "Your score indicates that you are
experiencing minimal depression."
        elif sumDepression >= 17 and sumDepression <= 19:
            DepressionResult = "Your score indicates that you are
experiencing mild depression."
```

```
        elif sumDepression >= 20 and sumDepression <= 22:
            DepressionResult = "Your score indicates that you are 
experiencing moderate depression "
        elif sumDepression >= 23 and sumDepression <= 27:
            DepressionResult = "Your score indicates that you are 
experiencing sever depression "
        else:
            DepressionResult = "Your score indicates that you are 
experiencing extremely severe depression "

        return DepressionResult, AnxietyResult, StressResult
```