CAPSTONE GROUP 5 BUSINESS REPORT

LIST OF CONTENTS

Report

Introduction

The aim of the project was to develop a software which sorted coins by certain currency and denominations. This process was split into three parts: creating the functions, making a text menu and forming a Graphical User Interface (GUI). The functionality of the program will be broken down in the three parts.

Methodology

Part 1 involves creating these functions by programming, with Python, how a given coin input is sorted into given denominations. The python script for coin sorting was designed using modules such as sys, functions and conditional statements. The modules were first imported, then loops within functions were defined to request user's input for the desired operation as illustrated in *Figure 1.1.* To create the currency converter, we visited **alphavantage** [1] to regenerate a personal API key, which is a unique identifier used to authenticate a user access a web service [2]. Next, we import the library as shown in *Figure 1.2.* After initialising the major variables and request users for their inputs, we use the user's API key to access alpha vantage to request the currency exchange rate for any desired calculations. Table 1 displays the front-end output.

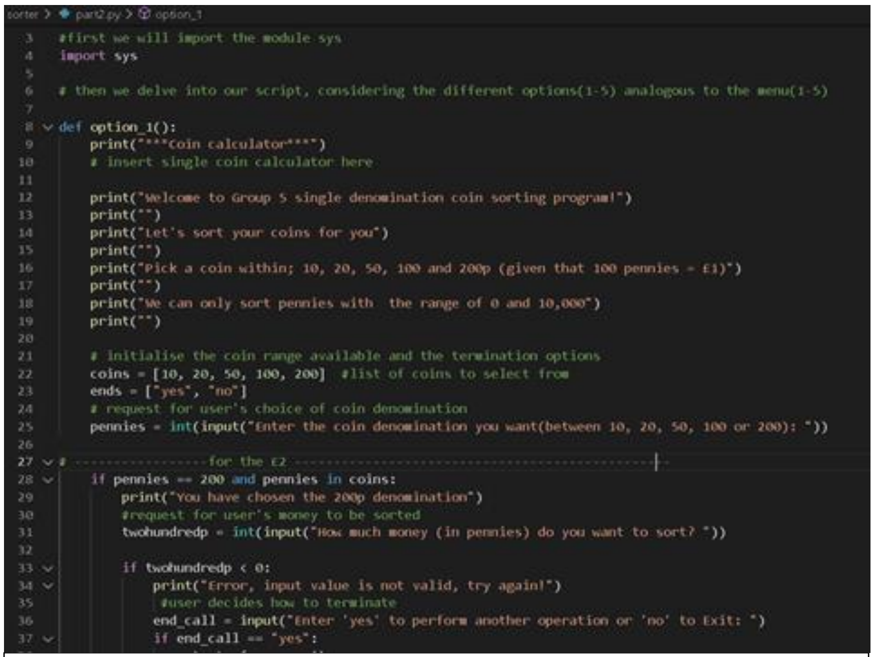
Part 2 of the problem involved taking the previous program and introducing a text menu to allow the user to navigate through the coin sorting program. To allow navigation, we created a main menu (*Figure 2.1*) that gave 6 different options. Each option calls a different function, that corresponds to an option for the user. Option 1 involves the use of an if statement as the user is prompted for a coin configuration. A variable 'pennies' is used to test if the user input for 'pennies' matches a given if/elif statement. Option 2 uses this same principle, but instead uses an if statement to test if the input statement is within a valid range before the modulus operator is applied. Option 3 (*Figure 2.2*) prints the available denominations, while Option 4 (*Figure 2.3*) allows the user to manually configure the program. These configurations are stored as global variables so are consistent throughout the program. If these variables have not been defined, then the program will use a default configuration (*Figure 2.4)* Option 5 (*Figure 2.5*) will print the current configuration, while Option 6 (*Figure 2.6)* will quit the program. Any invalid inputs throughout the program will return the user to the previous stage of navigation. Table 2 displays the front-end output.

Part 3 involved making a user-friendly GUI to represent the coin sorter on PyQT designer, a program to create a GUI. [3]. We design the GUI shown in *Figure 3.1.* The design stage involved creating a series of pushbuttons or input boxes that allow the user to interact with the GUI. On interaction, the functions created in Part 2 will be programmed to respond to the user's interaction in the form of a form of navigation through the GUI or perform a calculation. After designing the GUI, there is a conversion of the UI file from Qt designer to a python file. (*Figure 3.2*). The GUI output is illustrated in Table 3.

<u>**Conclusion**</u>

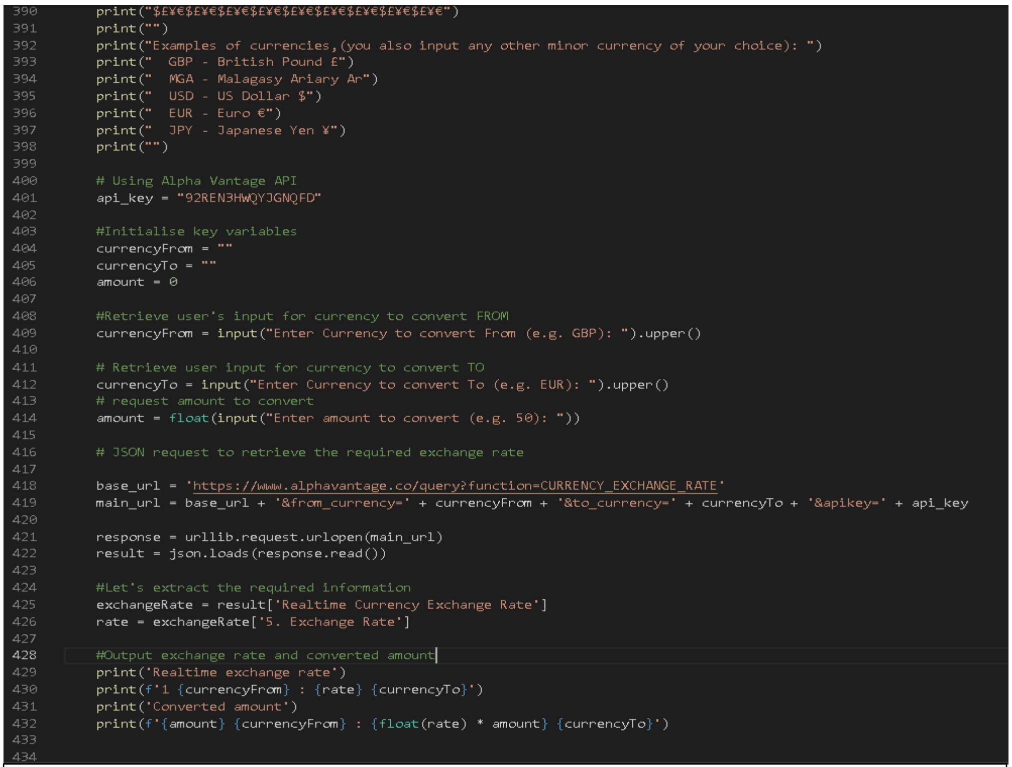
We submit this project in the belief that it meets the requirement outlined and the functionality delivers on the outcomes required. We look forward to your feedback before we move to the next stage of production.

Figure List



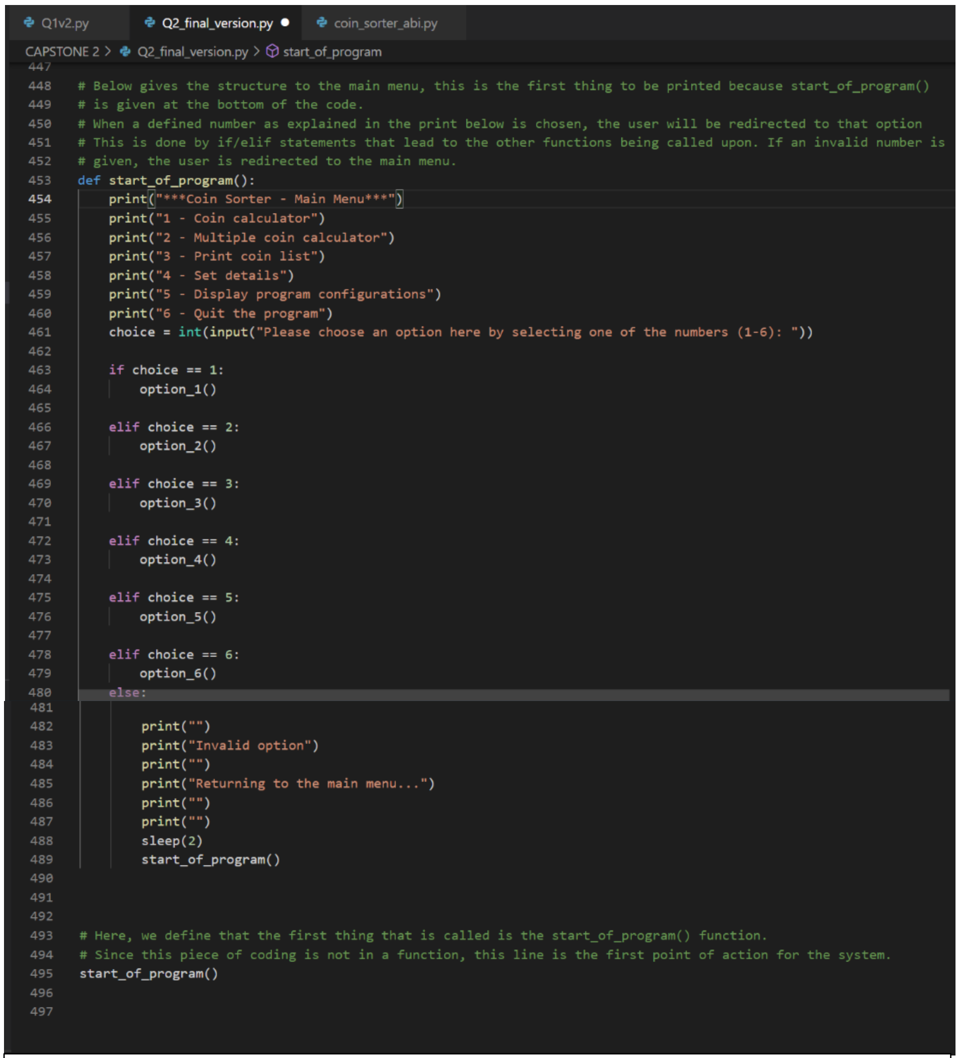
```python
#first we will import the module sys
import sys

# then we delve into our script, considering the different options(1-5) analogous to the menu(1-5)

def option_1():
    print("***Coin calculator***")
    # insert single coin calculator here

    print("Welcome to Group 5 single denomination coin sorting program!")
    print("")
    print("Let's sort your coins for you")
    print("")
    print("Pick a coin within; 10, 20, 50, 100 and 200p (given that 100 pennies = £1)")
    print("")
    print("We can only sort pennies with  the range of 0 and 10,000")
    print("")

    # initialise the coin range available and the termination options
    coins = [10, 20, 50, 100, 200]  #list of coins to select from
    ends = ["yes", "no"]
    # request for user's choice of coin denomination
    pennies = int(input("Enter the coin denomination you want(between 10, 20, 50, 100 or 200): "))

# ------------------for the £2 -----------------------------------------
    if pennies == 200 and pennies in coins:
        print("You have chosen the 200p denomination")
        #request for user's money to be sorted
        twohundredp = int(input("How much money (in pennies) do you want to sort? "))

        if twohundredp < 0:
            print("Error, input value is not valid, try again!")
             #user decides how to terminate
            end_call = input("Enter 'yes' to perform another operation or 'no' to Exit: ")
            if end_call == "yes":
```

Figure 1.1 - An image showing the start of the coding of Part 1. The coin sorting feature has been programmed via the use of an if statement allows the program to call a given function based on the user's integer input.



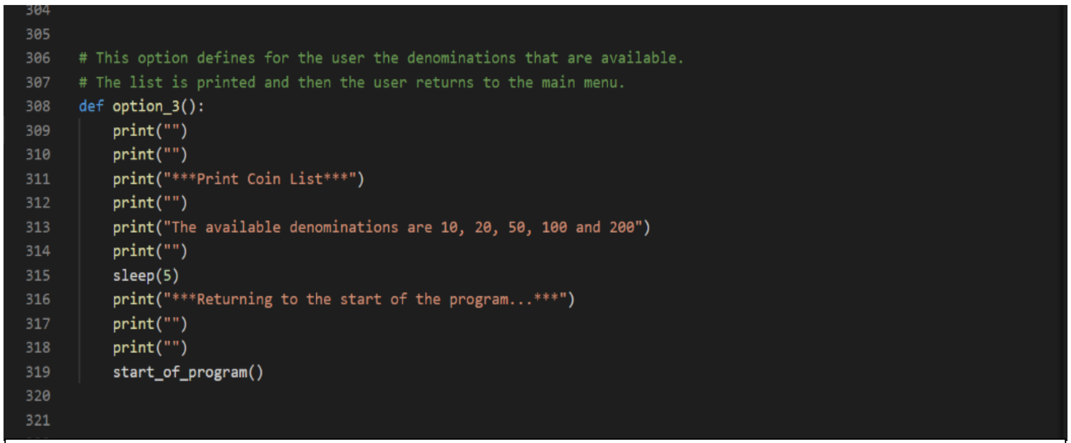
```python
    print("$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€")
    print("")
    print("Examples of currencies,(you also input any other minor currency of your choice): ")
    print("  GBP - British Pound £")
    print("  MGA - Malagasy Ariary Ar")
    print("  USD - US Dollar $")
    print("  EUR - Euro €")
    print("  JPY - Japanese Yen ¥")
    print("")

    # Using Alpha Vantage API
    api_key = "92REN3HWQYJGNQFD"

    #Initialise key variables
    currencyFrom = ""
    currencyTo = ""
    amount = 0

    #Retrieve user's input for currency to convert FROM
    currencyFrom = input("Enter Currency to convert From (e.g. GBP): ").upper()

    # Retrieve user input for currency to convert TO
    currencyTo = input("Enter Currency to convert To (e.g. EUR): ").upper()
    # request amount to convert
    amount = float(input("Enter amount to convert (e.g. 50): "))

    # JSON request to retrieve the required exchange rate

    base_url = 'https://www.alphavantage.co/query?function=CURRENCY_EXCHANGE_RATE'
    main_url = base_url + '&from_currency=' + currencyFrom + '&to_currency=' + currencyTo + '&apikey=' + api_key

    response = urllib.request.urlopen(main_url)
    result = json.loads(response.read())

    #Let's extract the required information
    exchangeRate = result['Realtime Currency Exchange Rate']
    rate = exchangeRate['5. Exchange Rate']

    #Output exchange rate and converted amount
    print('Realtime exchange rate')
    print(f'1 {currencyFrom} : {rate} {currencyTo}')
    print('Converted amount')
    print(f'{amount} {currencyFrom} : {float(rate) * amount} {currencyTo}')
```

Figure 1.2 - An image showing a part of the coding in Part 1, displaying the programming of the currency exchanger.



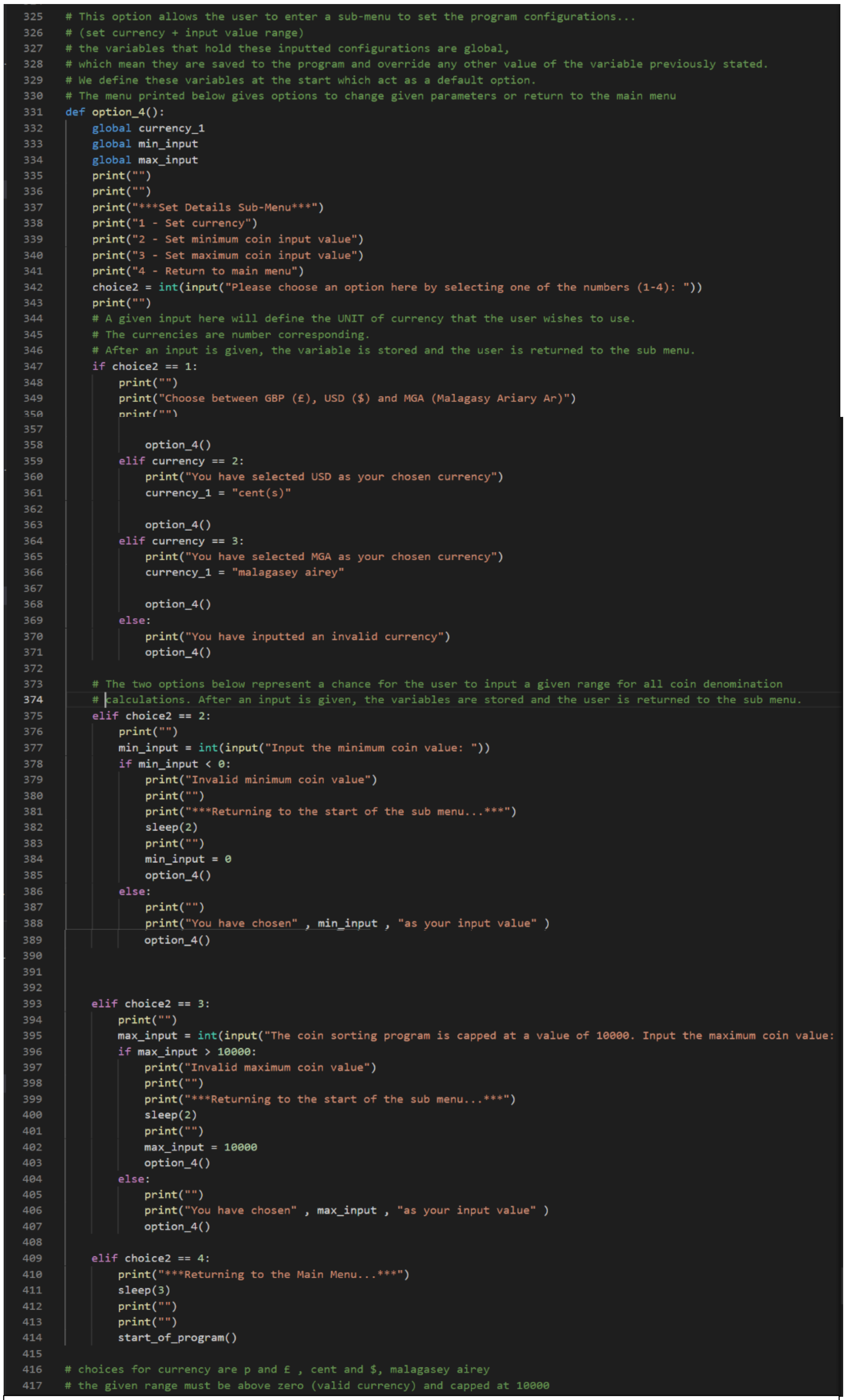
```python

# Below gives the structure to the main menu, this is the first thing to be printed because start_of_program()
# is given at the bottom of the code.
# When a defined number as explained in the print below is chosen, the user will be redirected to that option
# This is done by if/elif statements that lead to the other functions being called upon. If an invalid number is
# given, the user is redirected to the main menu.
def start_of_program():
    print("***Coin Sorter - Main Menu***")
    print("1 - Coin calculator")
    print("2 - Multiple coin calculator")
    print("3 - Print coin list")
    print("4 - Set details")
    print("5 - Display program configurations")
    print("6 - Quit the program")
    choice = int(input("Please choose an option here by selecting one of the numbers (1-6): "))

    if choice == 1:
        option_1()

    elif choice == 2:
        option_2()

    elif choice == 3:
        option_3()

    elif choice == 4:
        option_4()

    elif choice == 5:
        option_5()

    elif choice == 6:
        option_6()
    else:

        print("")
        print("Invalid option")
        print("")
        print("Returning to the main menu...")
        print("")
        print("")
        sleep(2)
        start_of_program()



# Here, we define that the first thing that is called is the start_of_program() function.
# Since this piece of coding is not in a function, this line is the first point of action for the system.
start_of_program()


```

Figure 2.1 – An image showing the coding of the main menu. The use of an if statement allows the program to call a given function based on the user's integer input.



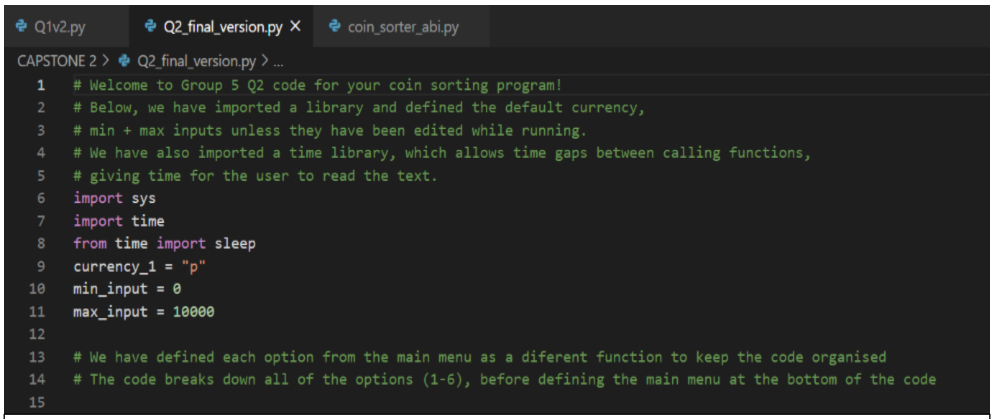
```python

# This option defines for the user the denominations that are available.
# The list is printed and then the user returns to the main menu.
def option_3():
    print("")
    print("")
    print("***Print Coin List***")
    print("")
    print("The available denominations are 10, 20, 50, 100 and 200")
    print("")
    sleep(5)
    print("***Returning to the start of the program...***")
    print("")
    print("")
    start_of_program()


```

Figure 2.2 – An image showing the code to make option 3, which prints the available coin configurations.



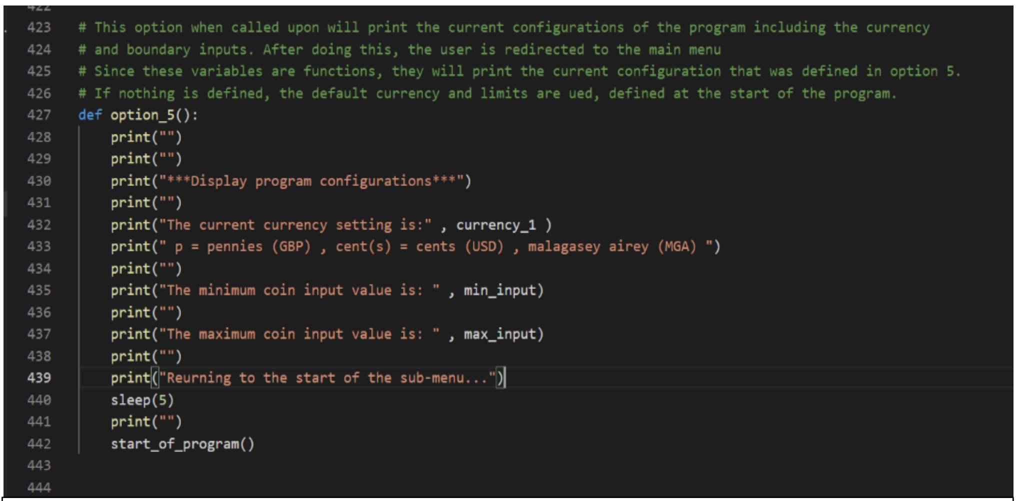
```python
# This option allows the user to enter a sub-menu to set the program configurations...
# (set currency + input value range)
# the variables that hold these inputted configurations are global,
# which mean they are saved to the program and override any other value of the variable previously stated.
# We define these variables at the start which act as a default option.
# The menu printed below gives options to change given parameters or return to the main menu
def option_4():
    global currency_1
    global min_input
    global max_input
    print("")
    print("")
    print("***Set Details Sub-Menu***")
    print("1 - Set currency")
    print("2 - Set minimum coin input value")
    print("3 - Set maximum coin input value")
    print("4 - Return to main menu")
    choice2 = int(input("Please choose an option here by selecting one of the numbers (1-4): "))
    print("")
    # A given input here will define the UNIT of currency that the user wishes to use.
    # The currencies are number corresponding.
    # After an input is given, the variable is stored and the user is returned to the sub menu.
    if choice2 == 1:
        print("")
        print("Choose between GBP (£), USD ($) and MGA (Malagasy Ariary Ar)")
        print("")

            option_4()
        elif currency == 2:
            print("You have selected USD as your chosen currency")
            currency_1 = "cent(s)"

            option_4()
        elif currency == 3:
            print("You have selected MGA as your chosen currency")
            currency_1 = "malagasey airey"

            option_4()
        else:
            print("You have inputted an invalid currency")
            option_4()

    # The two options below represent a chance for the user to input a given range for all coin denomination
    # calculations. After an input is given, the variables are stored and the user is returned to the sub menu.
    elif choice2 == 2:
        print("")
        min_input = int(input("Input the minimum coin value: "))
        if min_input < 0:
            print("Invalid minimum coin value")
            print("")
            print("***Returning to the start of the sub menu...***")
            sleep(2)
            print("")
            min_input = 0
            option_4()
        else:
            print("")
            print("You have chosen" , min_input , "as your input value" )
            option_4()



    elif choice2 == 3:
        print("")
        max_input = int(input("The coin sorting program is capped at a value of 10000. Input the maximum coin value:
        if max_input > 10000:
            print("Invalid maximum coin value")
            print("")
            print("***Returning to the start of the sub menu...***")
            sleep(2)
            print("")
            max_input = 10000
            option_4()
        else:
            print("")
            print("You have chosen" , max_input , "as your input value" )
            option_4()

    elif choice2 == 4:
        print("***Returning to the Main Menu...***")
        sleep(3)
        print("")
        print("")
        start_of_program()

# choices for currency are p and £ , cent and $, malagasey airey
# the given range must be above zero (valid currency) and capped at 10000
```

Figure 2.3 – An image showing the coding of option 4. A sub-menu is printed after option 4 is selected. An if statement is used to call different parts of the sub menu in response to the user's input.



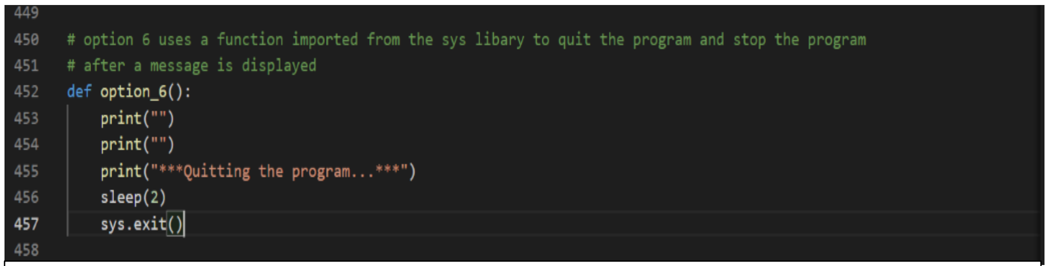
```python
# Welcome to Group 5 Q2 code for your coin sorting program!
# Below, we have imported a library and defined the default currency,
# min + max inputs unless they have been edited while running.
# We have also imported a time library, which allows time gaps between calling functions,
# giving time for the user to read the text.
import sys
import time
from time import sleep
currency_1 = "p"
min_input = 0
max_input = 10000

# We have defined each option from the main menu as a diferent function to keep the code organised
# The code breaks down all of the options (1-6), before defining the main menu at the bottom of the code

```

Figure 2.4 – A part of the coding that represents the default options. The program will use this configuration unless the user inputs a different configuration. The input from the configuration menu will override the default settings as they are global variables.



```python
# This option when called upon will print the current configurations of the program including the currency
# and boundary inputs. After doing this, the user is redirected to the main menu
# Since these variables are functions, they will print the current configuration that was defined in option 5.
# If nothing is defined, the default currency and limits are ued, defined at the start of the program.
def option_5():
    print("")
    print("")
    print("***Display program configurations***")
    print("")
    print("The current currency setting is:" , currency_1 )
    print(" p = pennies (GBP) , cent(s) = cents (USD) , malagasey airey (MGA) ")
    print("")
    print("The minimum coin input value is: " , min_input)
    print("")
    print("The maximum coin input value is: " , max_input)
    print("")
    print("Reurning to the start of the sub-menu...")
    sleep(5)
    print("")
    start_of_program()

```

Figure 2.5 – An image showing the coding of option 5. The variables are used within a string in a print function to display to the user the current configuration.



```python

# option 6 uses a function imported from the sys libary to quit the program and stop the program
# after a message is displayed
def option_6():
    print("")
    print("")
    print("***Quitting the program...***")
    sleep(2)
    sys.exit()

```

Figure 2.6 – An image showing the coding of option 6. The sys.exit command is used to kill the program.

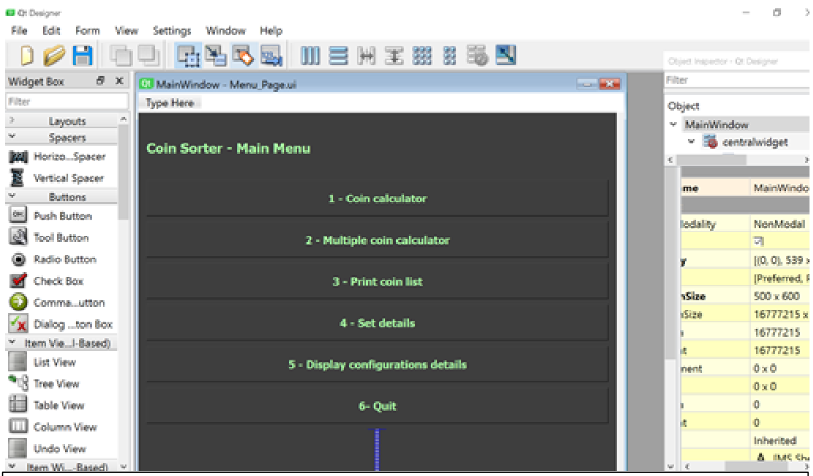
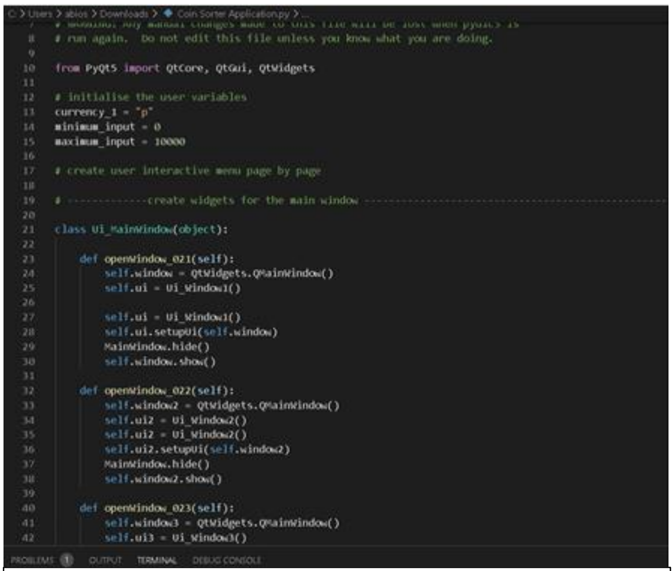


Figure 3.1 – An image showing the workspace environment in Qt designer, with a designed GUI interface.



```
# run again.  Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

# initialise the user variables
currency_1 = "p"
minimum_input = 0
maximum_input = 10000

# create user interactive menu page by page

# ------------create widgets for the main window ------------------------------------------

class Ui_MainWindow(object):

    def openWindow_021(self):
        self.window = QtWidgets.QMainWindow()
        self.ui = Ui_Window1()

        self.ui = Ui_Window1()
        self.ui.setupUi(self.window)
        MainWindow.hide()
        self.window.show()

    def openWindow_022(self):
        self.window2 = QtWidgets.QMainWindow()
        self.ui2 = Ui_Window2()
        self.ui2 = Ui_Window2()
        self.ui2.setupUi(self.window2)
        MainWindow.hide()
        self.window2.show()

    def openWindow_023(self):
        self.window3 = QtWidgets.QMainWindow()
        self.ui3 = Ui_Window3()
```

Figure 3.2 – An image showing the converted UI file from Qt designer to a python script, as shown above.

References

[1] “Alpha Vantage Support” Aug. 15, 2017. Accessed on: Mar. 8, 2021. [Online]. Available: https://www.alphavantage.co/support/#api-key

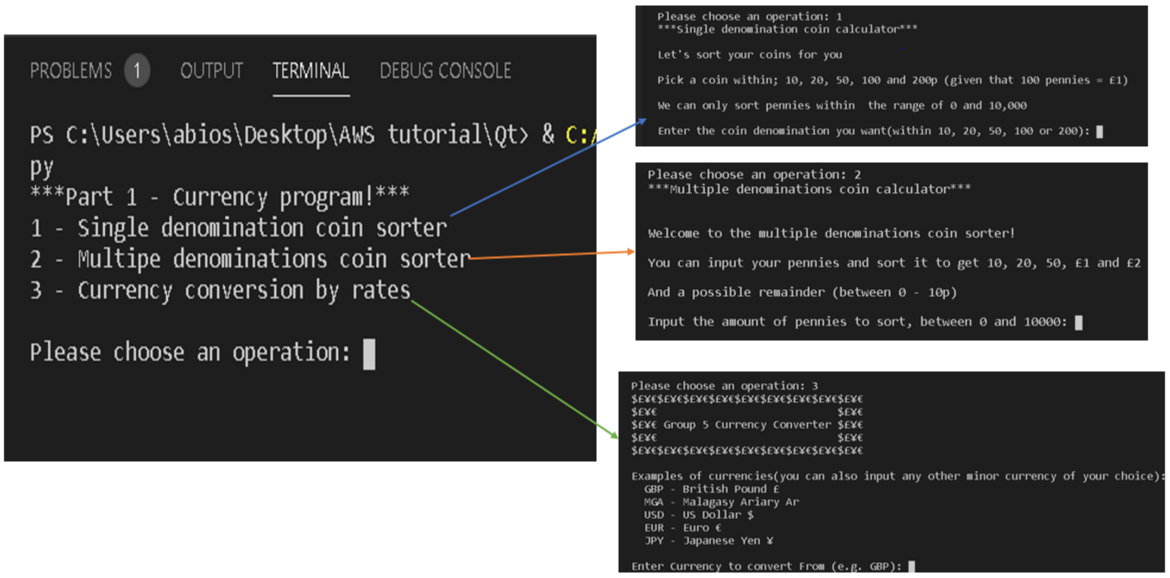
[2] “Why and when to use API keys | Cloud Endpoints with OpenAPI” Jul. 15, 2007. Accessed on: Mar. 6, 2021. [Online]. Available: https://cloud.google.com/endpoints/docs/openapi/when-why-api-key

[3] “PyQT Reference Guide” Feb. 21, 2015. Accessed on: Mar. 4, 2021. [Online]. Available: https://doc.bccnsoft.com/docs/PyQt5/

Appendix

Table 1 – A table showing screenshots of the output for Part 1

PROGRAM START:



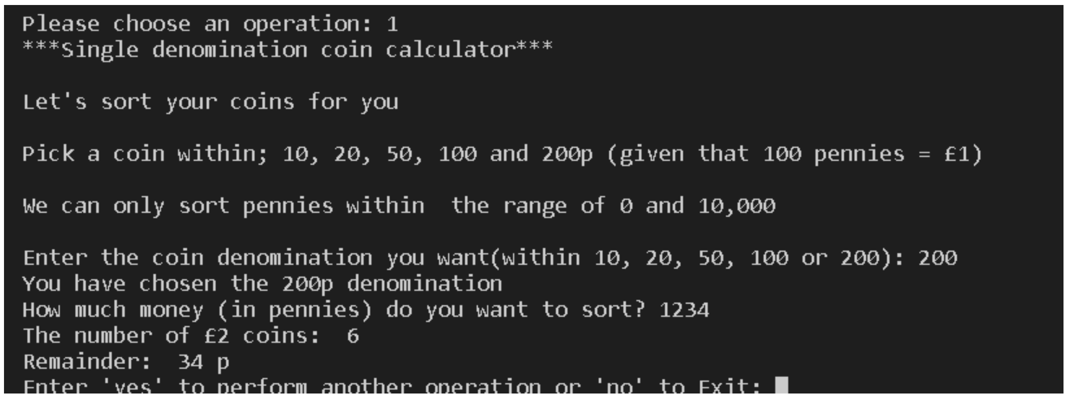
```
PS C:\Users\abios\Desktop\AWS tutorial\Qt> & C:/
py
***Part 1 - Currency program!***
1 - Single denomination coin sorter
2 - Multipe denominations coin sorter
3 - Currency conversion by rates

Please choose an operation:
```

```
Please choose an operation: 1
***Single denomination coin calculator***

Let's sort your coins for you

Pick a coin within; 10, 20, 50, 100 and 200p (given that 100 pennies = £1)

We can only sort pennies within  the range of 0 and 10,000

Enter the coin denomination you want(within 10, 20, 50, 100 or 200):
```

```
Please choose an operation: 2
***Multiple denominations coin calculator***


Welcome to the multiple denominations coin sorter!

You can input your pennies and sort it to get 10, 20, 50, £1 and £2

And a possible remainder (between 0 - 10p)

Input the amount of pennies to sort, between 0 and 10000:
```

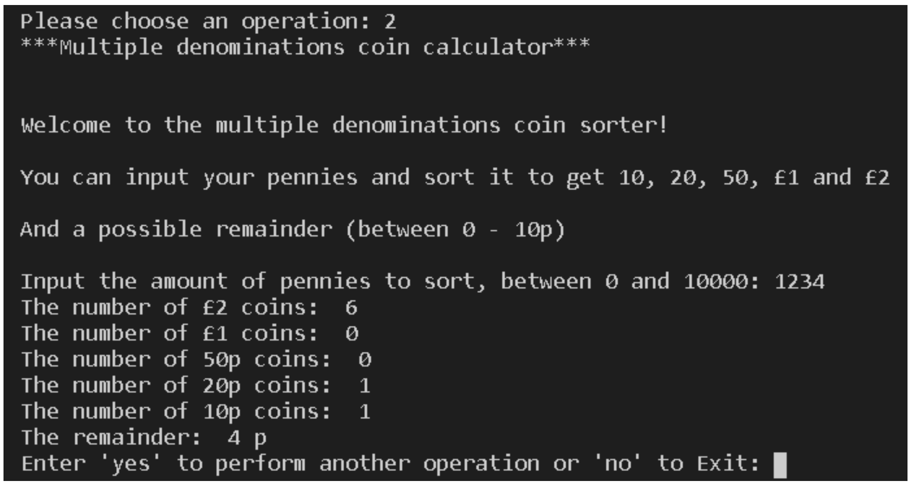
```
Please choose an operation: 3
$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€
$£¥€                                $£¥€
$£¥€ Group 5 Currency Converter $£¥€
$£¥€                                $£¥€
$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€$£¥€

Examples of currencies(you can also input any other minor currency of your choice):
  GBP - British Pound £
  MGA - Malagasy Ariary Ar
  USD - US Dollar $
  EUR - Euro €
  JPY - Japanese Yen ¥

Enter Currency to convert From (e.g. GBP):
```

OPTION 1 (in detail):



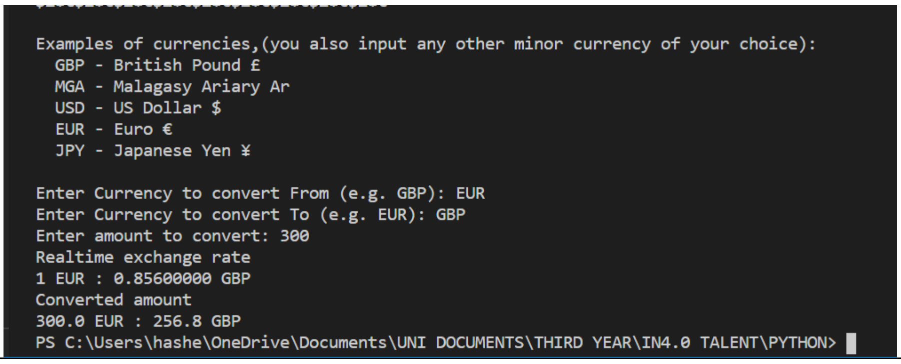
```
Please choose an operation: 1
***Single denomination coin calculator***

Let's sort your coins for you

Pick a coin within; 10, 20, 50, 100 and 200p (given that 100 pennies = £1)

We can only sort pennies within  the range of 0 and 10,000

Enter the coin denomination you want(within 10, 20, 50, 100 or 200): 200
You have chosen the 200p denomination
How much money (in pennies) do you want to sort? 1234
The number of £2 coins:  6
Remainder:  34 p
Enter 'yes' to perform another operation or 'no' to Exit:
```

OPTION 2 (in detail):



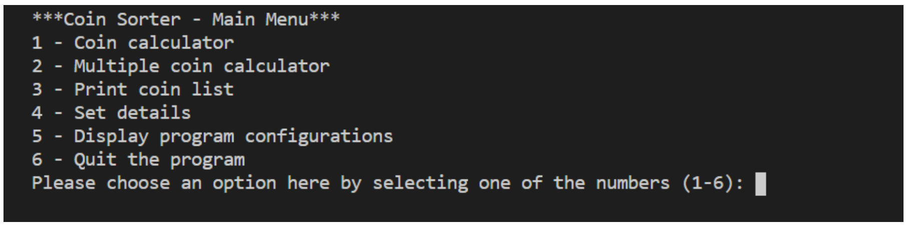
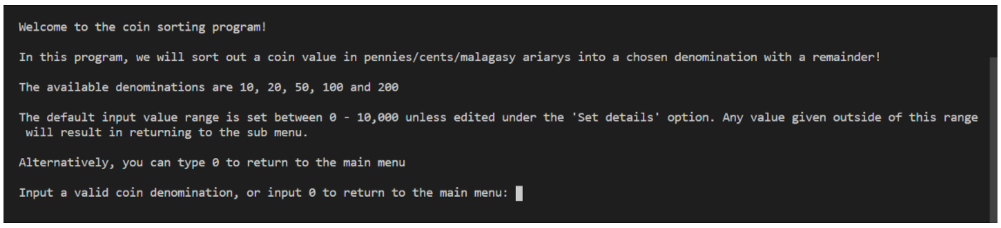
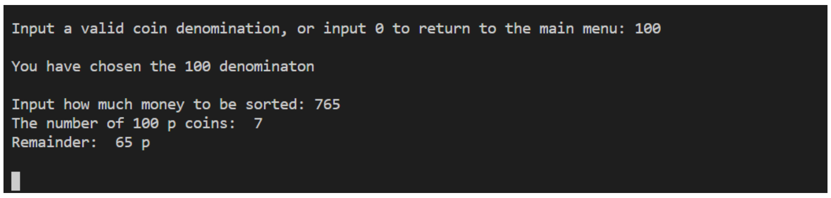
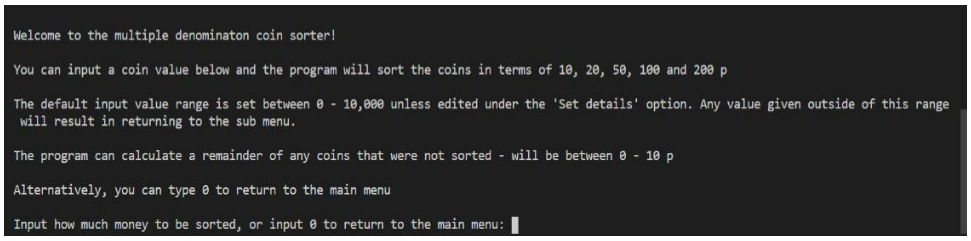
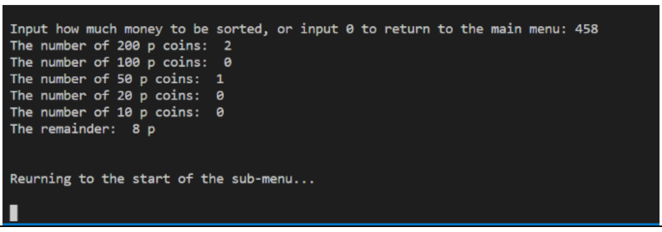
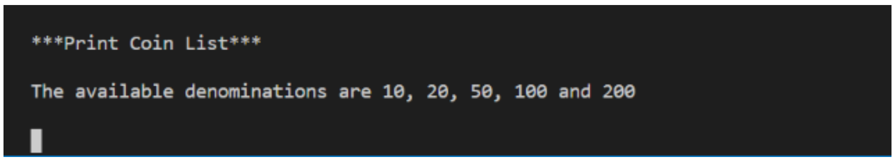
```
Please choose an operation: 2
***Multiple denominations coin calculator***


Welcome to the multiple denominations coin sorter!

You can input your pennies and sort it to get 10, 20, 50, £1 and £2

And a possible remainder (between 0 - 10p)

Input the amount of pennies to sort, between 0 and 10000: 1234
The number of £2 coins:  6
The number of £1 coins:  0
The number of 50p coins:  0
The number of 20p coins:  1
The number of 10p coins:  1
The remainder:  4 p
Enter 'yes' to perform another operation or 'no' to Exit:
```

OPTION 3 (in detail):



```
Examples of currencies,(you also input any other minor currency of your choice):
  GBP - British Pound £
  MGA - Malagasy Ariary Ar
  USD - US Dollar $
  EUR - Euro €
  JPY - Japanese Yen ¥

Enter Currency to convert From (e.g. GBP): EUR
Enter Currency to convert To (e.g. EUR): GBP
Enter amount to convert: 300
Realtime exchange rate
1 EUR : 0.85600000 GBP
Converted amount
300.0 EUR : 256.8 GBP
PS C:\Users\hashe\OneDrive\Documents\UNI DOCUMENTS\THIRD YEAR\IN4.0 TALENT\PYTHON>
```

Table 2 – A table showing screenshots of the output for Part 2

	**PROGRAM START** The main menu for the coin sorter. This is the first chance to interact with the program
 	**OPTION 1** Option 1 will lead the user to the singular coin sorter, where a coin denomination and coin value is inputted to be sorted
 	**OPTION 2** Option 2 will lead the user to the multiple coin denomination. A chance to input a coin value to be sorted across all available denominations
	**OPTION 3** Option 3 will print the available denominators

```
 ***Coin Sorter - Main Menu***
 1 - Coin calculator
 2 - Multiple coin calculator
 3 - Print coin list
 4 - Set details
 5 - Display program configurations
 6 - Quit the program
 Please choose an option here by selecting one of the numbers (1-6): 
```

```
Welcome to the coin sorting program!

In this program, we will sort out a coin value in pennies/cents/malagasy ariarys into a chosen denomination with a remainder!

The available denominations are 10, 20, 50, 100 and 200

The default input value range is set between 0 - 10,000 unless edited under the 'Set details' option. Any value given outside of this range
 will result in returning to the sub menu.

Alternatively, you can type 0 to return to the main menu

Input a valid coin denomination, or input 0 to return to the main menu: 
```

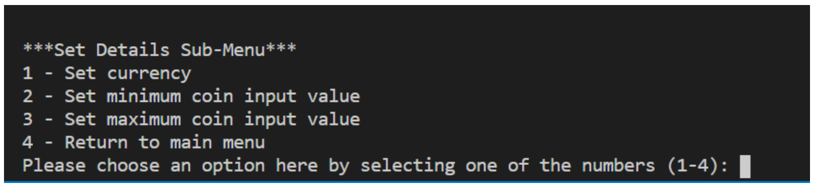
```
Input a valid coin denomination, or input 0 to return to the main menu: 100

You have chosen the 100 denominaton

Input how much money to be sorted: 765
The number of 100 p coins:  7
Remainder:  65 p
```

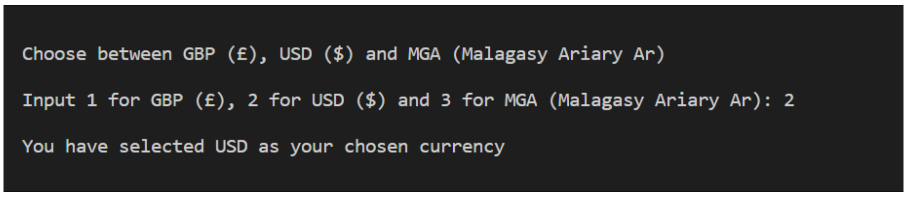
```
Welcome to the multiple denominaton coin sorter!

You can input a coin value below and the program will sort the coins in terms of 10, 20, 50, 100 and 200 p

The default input value range is set between 0 - 10,000 unless edited under the 'Set details' option. Any value given outside of this range
 will result in returning to the sub menu.

The program can calculate a remainder of any coins that were not sorted - will be between 0 - 10 p

Alternatively, you can type 0 to return to the main menu

Input how much money to be sorted, or input 0 to return to the main menu: 
```

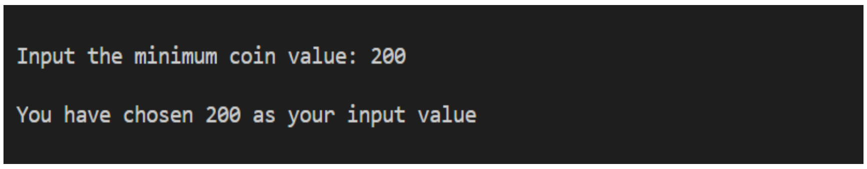
```
Input how much money to be sorted, or input 0 to return to the main menu: 458
The number of 200 p coins:  2
The number of 100 p coins:  0
The number of 50 p coins:  1
The number of 20 p coins:  0
The number of 10 p coins:  0
The remainder:  8 p


Reurning to the start of the sub-menu...
```

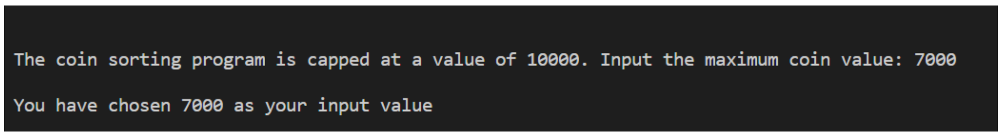
```
 ***Print Coin List***

 The available denominations are 10, 20, 50, 100 and 200
```

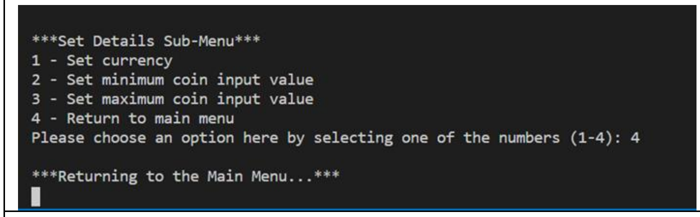
	that the program offers

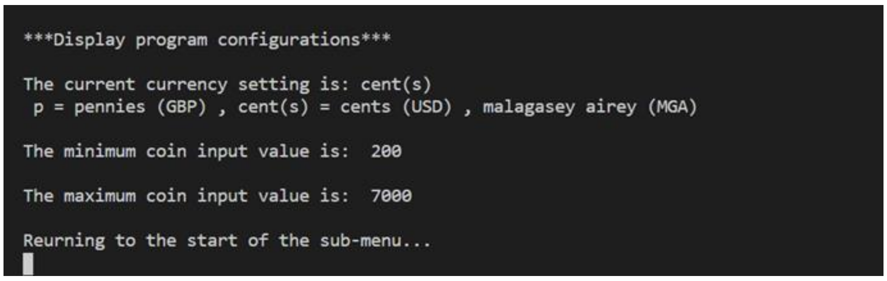

```
***Set Details Sub-Menu***
1 - Set currency
2 - Set minimum coin input value
3 - Set maximum coin input value
4 - Return to main menu
Please choose an option here by selecting one of the numbers (1-4):
```
	OPTION 4
	
	Option 4 will direct the user to a sub-menu, where the configurations of the program can be edited


```
Choose between GBP (£), USD ($) and MGA (Malagasy Ariary Ar)

Input 1 for GBP (£), 2 for USD ($) and 3 for MGA (Malagasy Ariary Ar): 2

You have selected USD as your chosen currency
```
	OPTION 4.1
	
	Option 4 – 1 allows the user to choose an available unit of currency that corresponds to an integer input


```
Input the minimum coin value: 200

You have chosen 200 as your input value
```
	OPTION 4-2
	
	Option 4 – 2 allows the user to set a minimum coin value input. This value must be a positive integer, or an error message is given


```
The coin sorting program is capped at a value of 10000. Input the maximum coin value: 7000

You have chosen 7000 as your input value
```
	OPTION 4.3
	
	Option 4 – 3 allows the user to set a maximum coin value. This value must be capped at 10000.


```
***Set Details Sub-Menu***
1 - Set currency
2 - Set minimum coin input value
3 - Set maximum coin input value
4 - Return to main menu
Please choose an option here by selecting one of the numbers (1-4): 4

***Returning to the Main Menu...***
```
	OPTION 4.4
	
	Option 4 – 4 will return the user to the main menu


```
***Display program configurations***

The current currency setting is: cent(s)
 p = pennies (GBP) , cent(s) = cents (USD) , malagasey airey (MGA)

The minimum coin input value is:  200

The maximum coin input value is:  7000

Reurning to the start of the sub-menu...
```
	OPTION 5
	
	Option 5 will print the current configuration of the program. If the user has not edited the configuration, a default option is defined

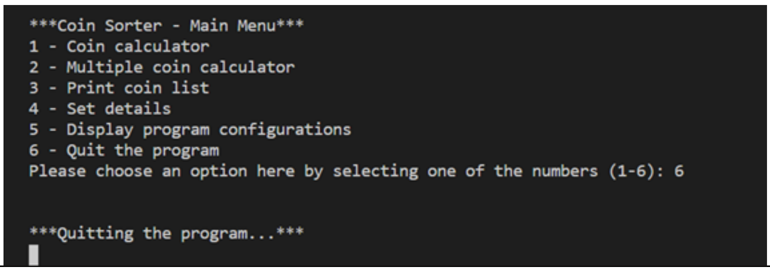
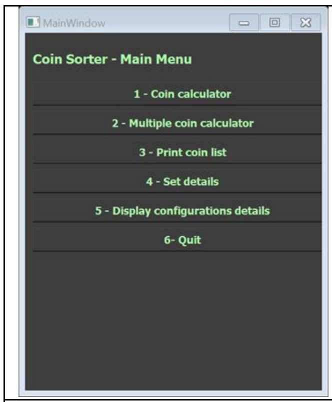
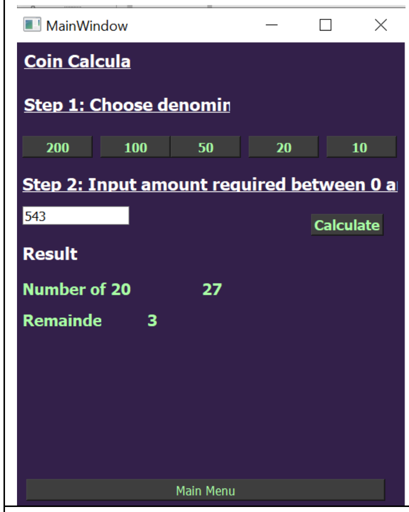
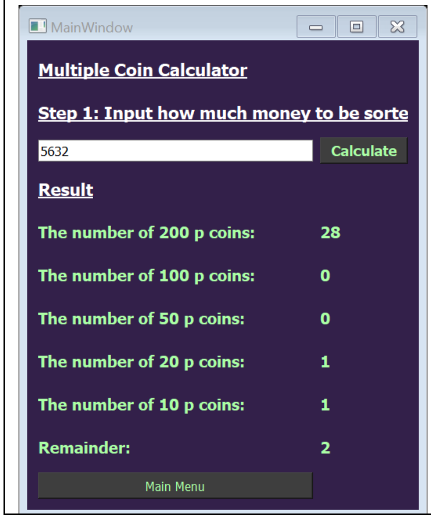
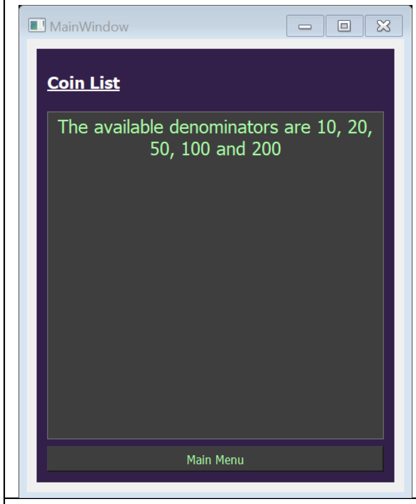
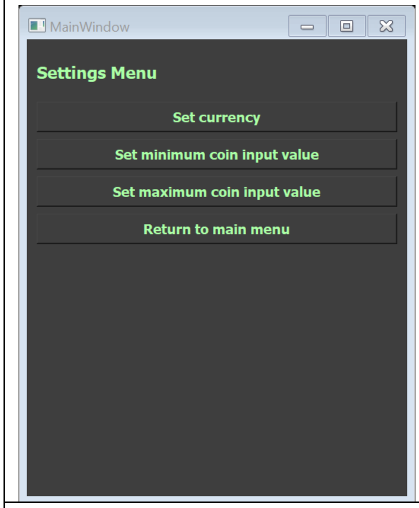
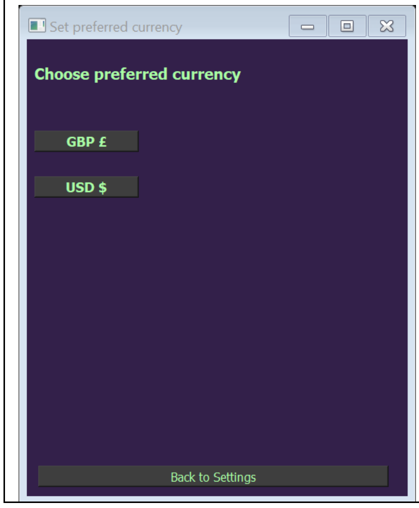
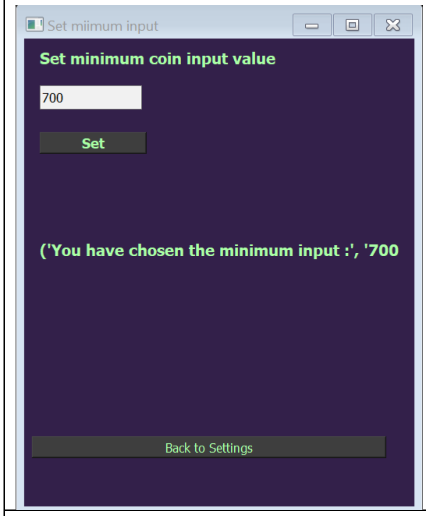
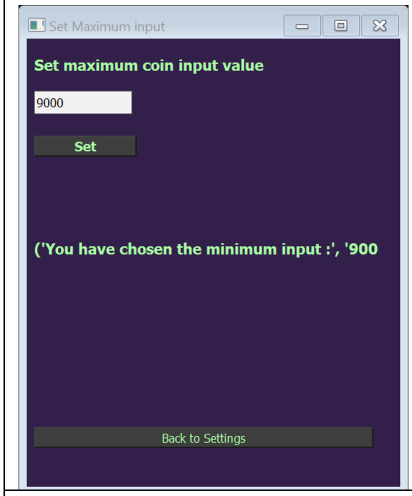
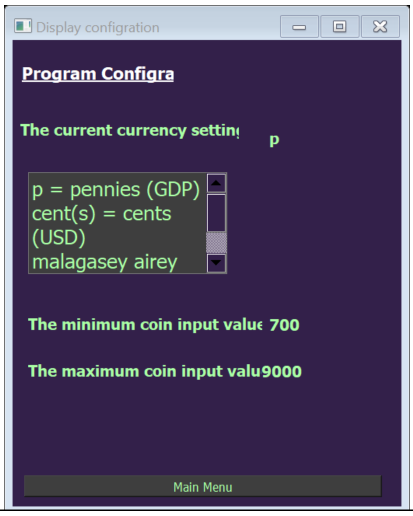

```
***Coin Sorter - Main Menu***
1 - Coin calculator
2 - Multiple coin calculator
3 - Print coin list
4 - Set details
5 - Display program configurations
6 - Quit the program
Please choose an option here by selecting one of the numbers (1-6): 6


***Quitting the program...***
```
	OPTION 6
	
	Option 6 terminates the program

Table 3 – A table showing screenshots of the output for Part 3

 MainWindow Coin Sorter - Main Menu 1 - Coin calculator 2 - Multiple coin calculator 3 - Print coin list 4 - Set details 5 - Display configurations details 6- Quit	**PROGRAM START** The main menu for the coin sorter. This is the first chance to interact with the program
 MainWindow Coin Calcula Step 1: Choose denomin 200 100 50 20 10 Step 2: Input amount required between 0 a Calculate Main Menu 543 Result Number of 20 27 Remainde 3	**OPTION 1** Option 1 will lead the user to the singular coin sorter, where a coin denomination is selected via a push button and coin value is inputted to be sorted
 MainWindow Multiple Coin Calculator Step 1: Input how much money to be sorte Calculate Main Menu 5632 Result The number of 200 p coins: 28 The number of 100 p coins: 0 The number of 50 p coins: 0 The number of 20 p coins: 1 The number of 10 p coins: 1 Remainder: 2	**OPTION 2** Option 2 will lead the user to the multiple coin sorter, where a coin denomination is inputted into an input bar and a "calculate" push button performs the calculation

 The available denominators are 10, 20, 50, 100 and 200	**OPTION 3** Option 3 prints the available coin denominations
 MainWindow Settings Menu Set currency Set minimum coin input value Set maximum coin input value Return to main menu	**OPTION 4** Option 4 will direct the user to a sub-menu, where the configurations of the program can be edited
 Set preferred currency Choose preferred currency GBP £ USD $ Back to Settings	**OPTION 4.1** This option leads the user to a page where a push button operator will store the currency for the running program.

 Set miimum input Set minimum coin input value 700 Set ('You have chosen the minimum input :', '700 Back to Settings	**OPTION 4.2** This option leads the user to a page where an input bar allows the user to set a minimum value. Once set, a message pops up below to confirm the selection.
 Set Maximum input Set maximum coin input value 9000 Set ('You have chosen the minimum input :', '900 Back to Settings	**OPTION 4.3** This option leads the user to a page where an input bar allows the user to set a maximum value. Once set, a message pops up below to confirm the selection.
 Display configration Program Configra The current currency settin p p = pennies (GDP) cent(s) = cents (USD) malagasey airey The minimum coin input value 700 The maximum coin input valu9000 Main Menu	**OPTION 5** Option 5 will print the current configuration of the program. If the user has not edited the configuration, a default option is defined. Here, we can see that the program has stored and displayed the inputted configurations successfully