Centre Name: North Halifax Grammar School Centre Number: 37337

Task completed: Task 2

Date started	Date completed
08/01/2019	26/02/2019

Contents:

Criteria:

Katarina is developing a two-player dice game.

The players roll two 6-sided dice each and get points depending on what they roll. There are 5 rounds in a game. In each round, each player rolls the two dice.

The rules are:

- The points rolled on each player's dice are added to their score.
- If the total is an even number, an additional 10 points are added to their score.
- If the total is an odd number, 5 points are subtracted from their score.
- If they roll a double, they get to roll one extra die and get the number of points rolled added to their score.
- The score of a player cannot go below 0 at any point.
- The person with the highest score at the end of the 5 rounds wins.

Candidate Name: James Sharrock
Candidate Number: 6140
Date: 26/02/2019

• If both players have the same score at the end of the 5 rounds, they each roll 1 die and whoever gets the highest score wins (this repeats until someone wins).

Only authorised players are allowed to play the game. Where appropriate, input from the user should be validated. Design, develop, test and evaluate a program that:

1. Allows two players to enter their details, which are then authenticated to ensure that they are authorised players.

2. Allows each player to roll two 6-sided dice.

3. Calculates and outputs the points for each round and each player's total score.

4. Allows the players to play 5 rounds.

5. If both players have the same score after 5 rounds, allows each player to roll 1 die each until someone wins.

6. Outputs who has won at the end of the 5 rounds.

7. Stores the winner's score, and their name, in an external file.

8. Displays the score and player name of the top 5 winning scores from the external file.

Analysis

Success Criteria:

1. Have a working Username and Password Function checking for authorised players

2. Have a dice rolling system asking the player to roll it

3. Have working algorithms that add points to the score or take away points depending on the above criteria

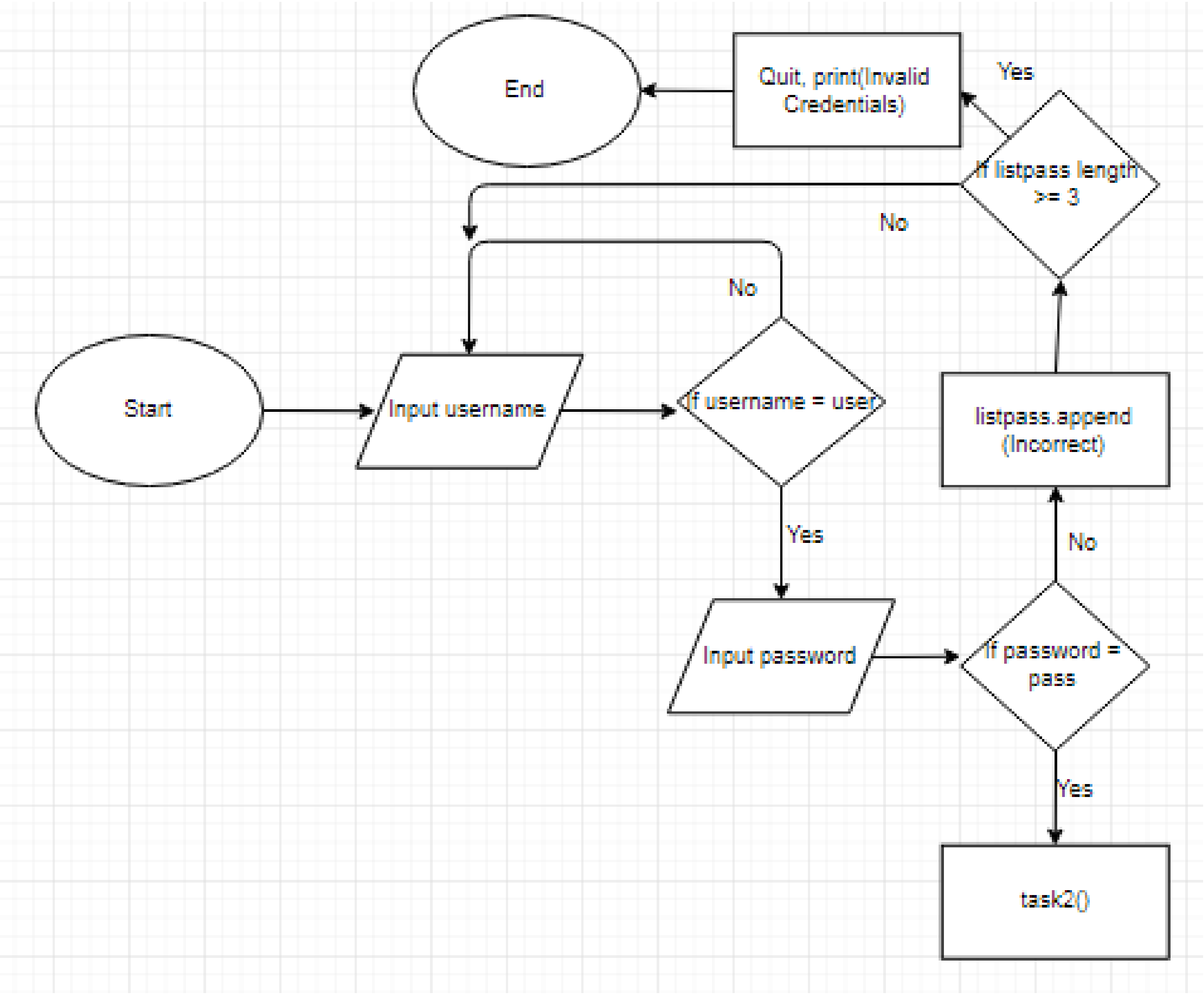
4. Have a decider system

5. Have a working Total System that is stored on a file

Centre Name: Centre Number:

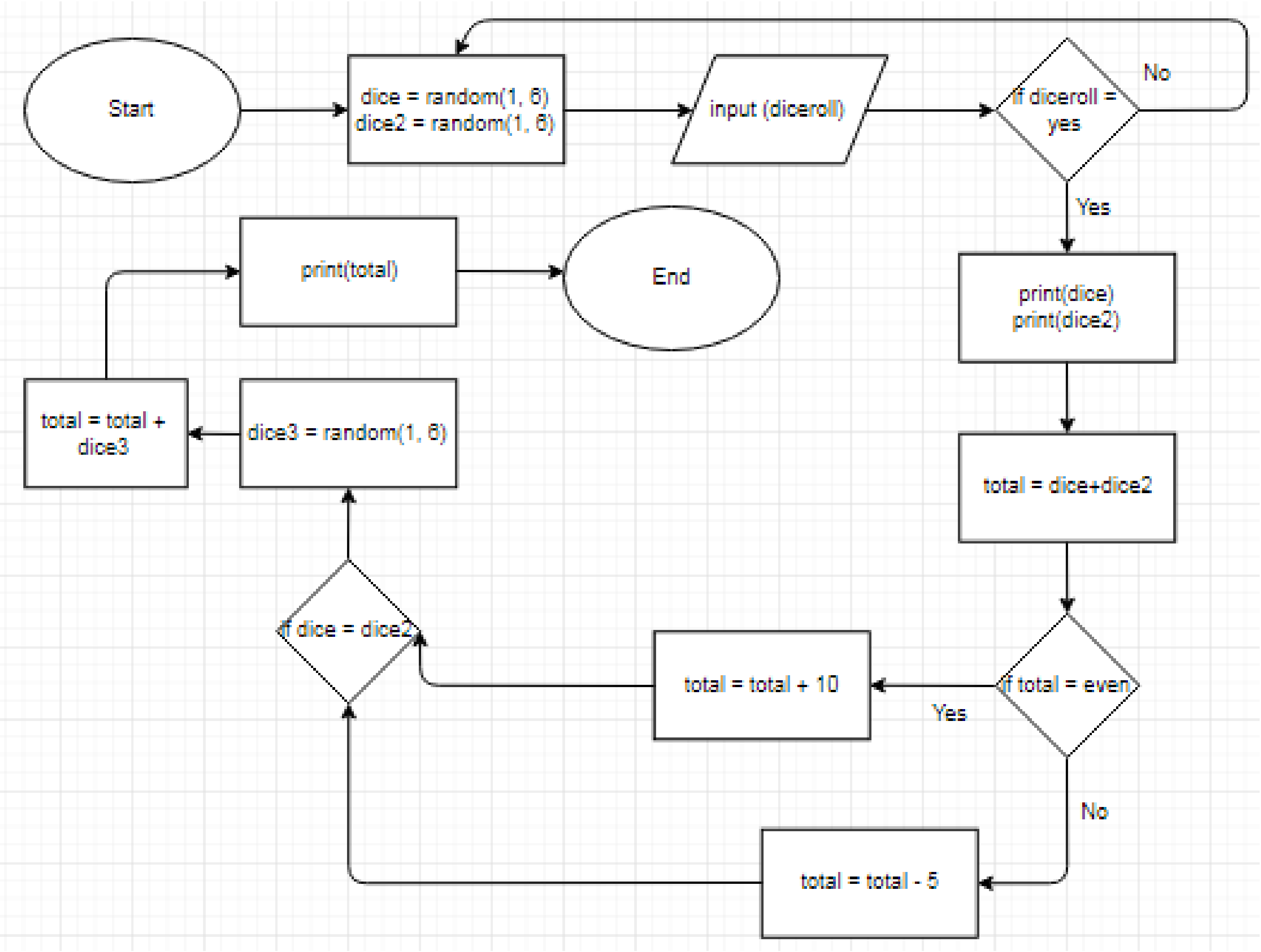
Design

Username and Password Authentication flowchart:



Dice Roll Flowchart:

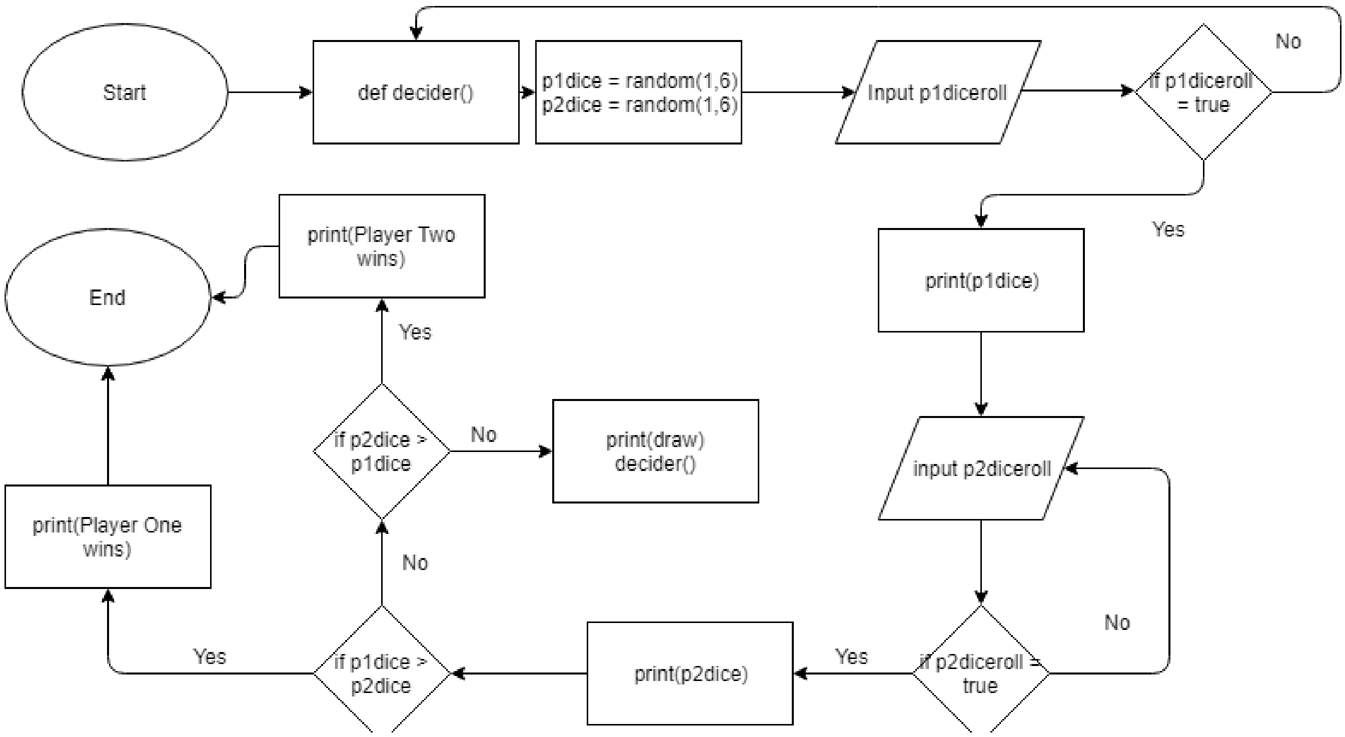
Centre Name: Centre Number:



Decider Flowchart:

Candidate Name: James Sharrock
Candidate Number: 6140
Date: 26/02/2019

Centre Name: Centre Number:



Pseudocode for login:

Candidate Name: James Sharrock
Candidate Number: 6140
Date: 26/02/2019

Centre Name: Centre Number:

```
username = user
password = pass
listpass = []

def username():
  userinput = input("Input your username")
  if userinput == username:
    def password():
      passinput = input("Enter your password")
      if passinput == password:
        print("You are in your account")
        game()
      else:
        listpass.append(incorrect)
        if listpasslength >= 3:
          print("You have been locked out from your account")
          quit()
        else:
          password()
  else:
    username()
```

Test design

My tests:

(Correct Input)

Test	What am I testing?	What data will I use?	Normal/Boundary/Erroneous?	Expected Result
1	Username**(User1 or User2)**	User1	Normal	Asks to Input password
2	Username**(User1 or User2)**	User2	Normal	Asks to Input password
3	Username**(User1 or User2)**	User3	Boundary	Quits Window
4	Username**(User1 or User2)**	*//A8	Erroneous	Quits Window
5	Username**(User1 or User2)**		Erroneous	Quits Window
6	Password**(Pass)**	pass	Normal	Authentication is complete. Game runs
7	Password**(Pass)**	Pass	Boundary	Quits Window
8	Password**(Pass)**	*//A8	Erroneous	Quits Window
9	Password**(Pass)**		Erroneous	Quits Window

Development

My program code:

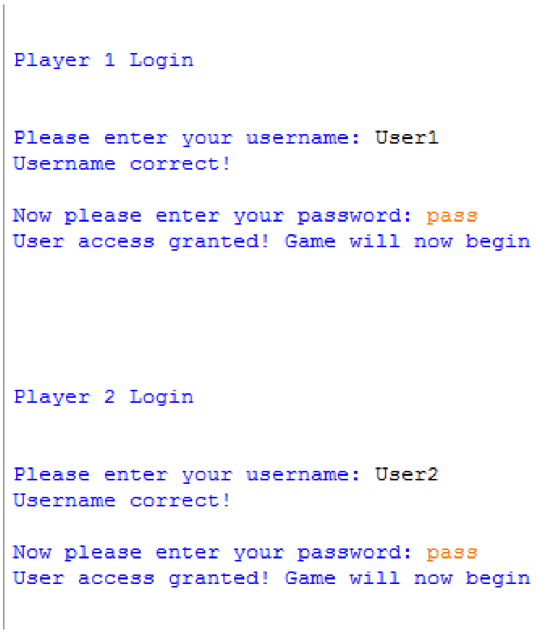
Linked at [My Programming Project(https://repl.it/@James_Sharrock/Python-Programming-Project)](https://repl.it/@James_Sharrock/Python-Programming-Project)

You can see a live Playthrough [here.(https://youtu.be/cgfXiB1DuRQ)](https://youtu.be/cgfXiB1DuRQ)

I have an extended asci art version of the code [here(https://repl.it/@James_Sharrock/Python-Programming-Project-Ascii-version)](https://repl.it/@James_Sharrock/Python-Programming-Project-Ascii-version)

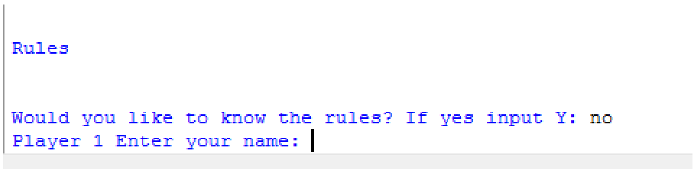
Intro makes the game seem more user friendly -

Both of the users input their logins for correct authentication –

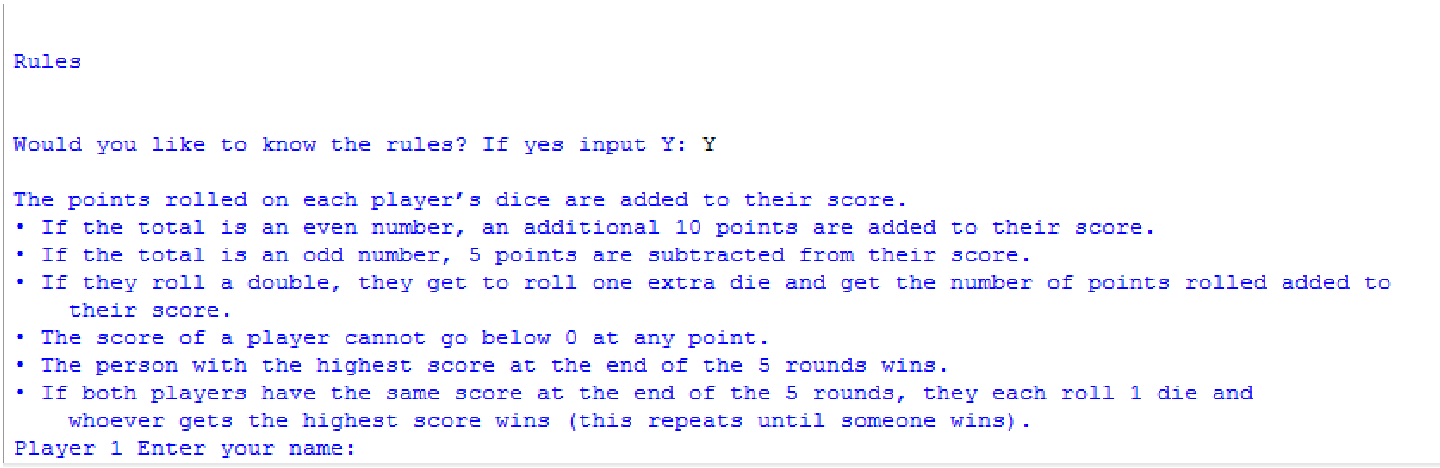


```
Player 1 Login


Please enter your username: User1
Username correct!

Now please enter your password: pass
User access granted! Game will now begin




Player 2 Login


Please enter your username: User2
Username correct!

Now please enter your password: pass
User access granted! Game will now begin
```

The players have a choice of whether they want to see the rules for the game or not -



```
Rules


Would you like to know the rules? If yes input Y: no
Player 1 Enter your name: 
```



```
Rules


Would you like to know the rules? If yes input Y: Y

The points rolled on each player’s dice are added to their score.
• If the total is an even number, an additional 10 points are added to their score.
• If the total is an odd number, 5 points are subtracted from their score.
• If they roll a double, they get to roll one extra die and get the number of points rolled added to
    their score.
• The score of a player cannot go below 0 at any point.
• The person with the highest score at the end of the 5 rounds wins.
• If both players have the same score at the end of the 5 rounds, they each roll 1 die and
    whoever gets the highest score wins (this repeats until someone wins).
Player 1 Enter your name:
```

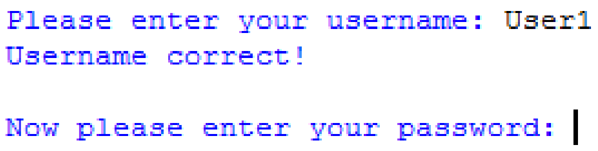
Testing

My tests:

Test	What am I testing?	Expected result	Pass/Fail	Do I need to change my program? If so, how?
1	Username Input	Asks for password	Pass	N/A
2	Username Input	Asks for password	Pass	N/A
3	Username Input	Quits Window	Pass	N/A
4	Username Input	Quits Window	Pass	N/A
5	Username Input	Quits Window	Pass	N/A
6	Password Input	Authentication Completes	Pass	N/A
7	Password Input	Asks for Username	Pass	N/A
8	Password Input	Asks for Username	Pass	N/A
9	Password Input	Asks for Username	Pass	N/A

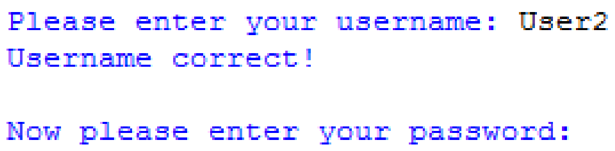
My test screenshots:

Test 1:



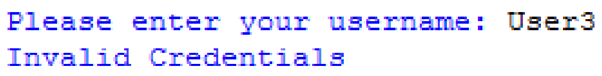
```
Please enter your username: User1
Username correct!

Now please enter your password:
```

Test 2:



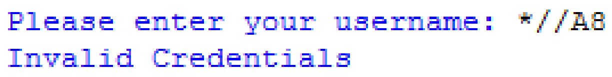
```
Please enter your username: User2
Username correct!

Now please enter your password:
```

Test 3:



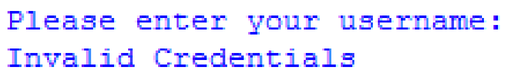
```
Please enter your username: User3
Invalid Credentials
```

Test 4:



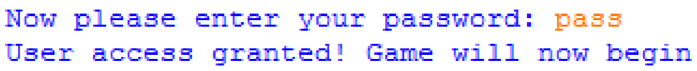
```
Please enter your username: *//A8
Invalid Credentials
```

Test 5:



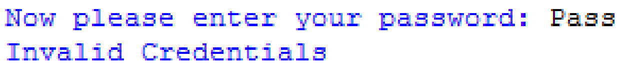
```
Please enter your username:
Invalid Credentials
```

Test 6:



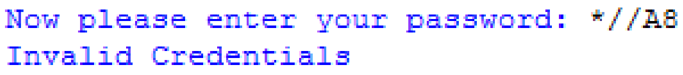
```
Now please enter your password: pass
User access granted! Game will now begin
```

Test 7:



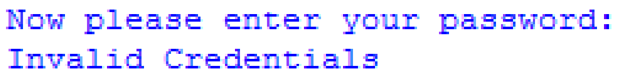
```
Now please enter your password: Pass
Invalid Credentials
```

Test 8:



```
Now please enter your password: *//A8
Invalid Credentials
```

Test 9:



```
Now please enter your password:
Invalid Credentials
```

Evaluation

How successful was my program?

My program was on the whole, a great success. I achieved all the objectives except, 'Displays the score and player name of the top 5 winning scores from the external file.' In my first programming session I created the username and password function for login. I later came back to this and created my global variable 'listpass' allowing me to create the 3 wrong password lockout system. I then set about creating a round for the game. I made all of the 6 dice needed as 2 extra dice would be used for if the first two dice were doubles and a third one would be rolled. I made the game, print out the players scores onto the screen and then set about the logic of the scoring system and checking if the total score was odd or even and adding or removing the appropriate amount of points to the player's score. I decided on using modulus to find the remainder of the score when divided by 2. If it divided by 2 with no remainder it was even and if not, it was odd. I then checked if dice1 was equal to dice2 and if this was true then a third dice would be rolled and the player's score would be appended to hold this new scoring dice. Therefore, I had created a working round of the game. My initial reaction to creating the five round game was to create a loop using for i in range(5). However, this did not work as it meant that the score was always being changed at the beginning of the new round and therefore the total of the 5 rounds was equal to the 5th total. This means if a player scored a total of 110 and got 24 in the last round, the logic would believe they scored a total of 40. After a few hours of tinkering and thinking, I decided to create a list in which the total of each round could be appended to it and withdrawn later to create a total. This worked and allowed me to having a working program albeit without a decider function and without working out the winner. I withdrew the scores from the program and compared them to each other using if and else statements. If they were equal it would run decider function which I created later. If not it compared again and printed out the winner. I created my decider function to have 1 dice for each player which would be rolled and then the dice would be compared. Again if they were the same it would rerun the decider function. If not, it would print the winner. I then created an external file called Programming_Statistics in which the winners score would be appended. I was unable to make it so only the top 5 scores were put in the external file. I understood the logic that I need to split up the User's names and scores which had been withdrawn from the external file and the new score, order the scores and then splice the name and score back together again in the correct order. I then had to write to the file to delete all of the old information and then print out each line of the file. In reality I managed to make sure only the winner's score was appended ad then print each line of the file. In Conclusion, I think my program was in majority a success.

Centre Name: Centre Number:

What new skills have I developed?

Problem solving and algorithmic thinking as well as abstraction and decomposition when I broke down my program into workable snippets of codes inserted into functions.

Candidate Name:
Candidate Number:
Date: