COMP 1039

Problem Solving and Programming

Programming Assignment 1

UniSA STEM
The University of South Australia
March, 2023

Contents

INTRODUCTION

This document describes the first assignment for Problem Solving and Programming.

The assignment is intended to provide you with the opportunity to put into practice what you have learnt in the course by applying your knowledge and skills to the implementation of a game called **Petals Around the Rose** and **a simple number converter.**

This assignment is an **individual task** that will require an **individual submission**. If you are an **internal student,** you will be required to submit your work via learnonline before **Monday 24 April (week 7), 9am.** Internal students are **not** required to demonstrate their work in person.

This document is a kind of specification of the required end product that will be generated by implementing the assignment. Like many specifications, it is written in English and hence will contain some imperfectly specified parts. Please make sure you seek clarification if you are not clear on any aspect of this assignment.

ASSIGNMENT OVERVIEW

There are **two parts** to this assignment:

Part I: Petals Around the Rose

You are required to write a Python program that allows a user to play a game called **Petals Around the Rose.** The program allows the user to repeatedly guess the answer to the puzzle until the user chooses to stop guessing/playing. Once the user chooses to stop guessing, the program will report the user's and game play statistics to the screen. You may like to play a web version of the game: http://www.borrett.id.au/computing/petals-j.htm.

Part II: Number converter

You are required to write a Python program that performs decimal to binary conversion (and vice versa) on numbers entered by the user. You are only required to deal with whole numbers. If you are interested, you may like to read the following on binary numbers: http://en.wikipedia.org/wiki/Binary_number.

"There are only 10 types of people in the world: those who understand binary, and those who don't." http://en.wikipedia.org/wiki/Mathematical_joke

Please ensure that you read sections titled 'Part I Specification' and 'Part II Specification' below for further details.

GRADUATE QUALITIES

By undertaking this assessment, you will progress in developing the qualities of a University of South Australia graduate.

The Graduate qualities being assessed by this assignment are:

- The ability to demonstrate and apply a body of knowledge (GQ1) gained from the lectures, workshops, practicals and readings. This is demonstrated in your ability to apply problem solving and programming theory to a practical situation.

- The development of skills required for lifelong learning (GQ2), by searching for information and learning to use and understand the resources provided (Python standard library, lectures, workshops, practical exercises, etc); in order to complete a programming exercise.

- The ability to effectively problem solve (GQ3) using Python to complete the programming problem. Effective problem solving is demonstrated by the ability to understand what is required, utilise the relevant information from lectures, workshops and practical work, write Python code, and evaluate the effectiveness of the code by testing it.

- The ability to work autonomously (GQ4) in order to complete the task.

- The use of communication skills (GQ6) by producing code that has been properly formatted; and writing adequate, concise and clear comments.

- The application of international standards (GQ7) by making sure your solution conforms to the standards presented in the Python Style Guide slides (available on the course website).

PART I SPECIFICATION – PETALS AROUND THE ROSE

You are required to write a Python program called *yourEmailId*_rose.py that allows a player to play a game called **Petals Around the Rose**.

Petals Around the Rose

You are required to write a Python program that allows a player to play **a game of Petals Around the Rose**. The program allows the user to repeatedly guess the answer to the puzzle until the user chooses to stop guessing/playing. Once the user chooses to stop guessing, the program will report the game statistics to the screen. You may also like to read Bill Gates and Petals Around the Rose: http://www.borrett.id.au/computing/petals-bg.htm.

We will be adhering to the following 'Petals Around the Rose' rules and game play for the assignment.

Petals Around the Rose Game Play and Rules:

The name of the game is Petals Around the Rose and the name of the game is important. The computer will roll five dice and ask the user to guess the score for the roll. The score will always be zero or an even number. The user's mission is to work out how the computer calculates the score in order to become a *Potentate of the Rose*.

- To begin, the following instructions are displayed to the screen and the user is asked whether they would like to play Petals Around the Rose, i.e.:

```
Petals Around the Rose
----------------------

The name of the game is 'Petals Around the Rose'.  The name of the
game is important.  The computer will roll five dice and ask you to
guess the score for the roll.  The score will always be zero or an
even number.  Your mission, should you choose to accept it, is to
work out how the computer calculates the score.  If you succeed in
working out the secret and guess correctly three times in a row, you
become a Potentate of the Rose.

Would you like to play Petals Around the Rose [y|n]?
```

If the user enters 'n', the following message is displayed to the screen only:

```
No worries... another time perhaps... :)
```

If the user enters 'y', game play continues as normal.

- Game play is as follows:
- The dice face values are then displayed to the screen and the user is asked to guess the score for the roll, for example:

```
     Die 1     Die 2     Die 3     Die 4     Die 5
      [4]       [3]       [2]       [5]       [6]

      * *       *         *         * *       * *
                 *                   *        * *
      * *         *         *       * *       * *


Please enter your guess for the roll:
```

  The numbers in the above output will change depending on the dice values rolled.

- If the user guesses correctly, the following message is displayed:

```
Well done! You guessed it!
```

- If the user does not guess correctly one of two possible messages is displayed to the screen. If the user enters an incorrect non-even number, the following message is displayed to the screen, for example:

```
No sorry, it's 0 not 9. The score is always even.
```

  If the user enters an incorrect even number, the following message is displayed to the screen, for example:

```
No sorry, it's 0 not 6.
```

  The numbers in the above output will change depending on what the user has entered and the score value for the roll.

- The user is then asked whether they would like to play again with the following prompt. Game play continues while the user enters 'y' at the prompt:

```
Roll dice again [y|n]?
```

- If the user has three (3) **incorrect** guesses **in a row** the following message is displayed to the screen:

```
Hint: The name of the game is important... Petals Around the Rose.
```

- If the user has three (3) **correct** guesses **in a row**, the following message is displayed to the screen:

```
Congratulations! You have worked out the secret!
Make sure you don't tell anyone!
```

- Game play continues until the user chooses to quit, i.e. enters 'n' at the following prompt:

```
Roll dice again [y|n]?
```

- The solution or formula for Petals Around the Rose can be found here:

  https://en.wikipedia.org/wiki/Petals_Around_the_Rose

  The key to working out the answer is given by the name of the puzzle, where the “rose” is the centre dot that appears only on the 3 and 5 faces of a die, while the “petals” are the dots which surround the centre dot. Therefore, the result can be calculated by counting 2 for each 3 face and 4 for each 5 face. In the example provided above, there is one 5 face and one 3 face, so the result is four plus two, resulting in a total of six.

- Once the user chooses to quit (after having played at least one game), game summary and statistics are displayed to the screen, for example:

```
Game Summary
============

You played 2 games:
  |--> Number of correct guesses:     1
  |--> Number of incorrect guesses:   1


Dice Roll Stats:

Face  Frequency
   1  *
   2  ***
   3  *
   4  *
   5  **
   6  **

Thanks for playing!
```

The numbers in the above output will change depending on the results of the user's games and the dice values rolled. The above should not be displayed if the user has not played any games.

You do not have to write the code that displays the die face values to the screen, a module containing a function that does that for you has been provided. The **`dice.py`** file is a module that contains a function called **`display_dice()`** that displays the face values of the dice to the screen for you. You are required to use this as part of this assignment, however, **please do not modify the `dice.py` file.**

PRACTICAL REQUIREMENTS (PART I)

It is recommended that you develop this part of the assignment in the suggested stages.

It is expected that your solution WILL include the use of:

- Your solution in one file called `yourEmailId_rose.py`.
- Appropriate and well constructed `while` and/or `for` loops (as necessary).
- Appropriate `if, if-else, if-elif-else` statements (as necessary).
- The supplied `dice.py` module (containing the `display_dice` function). This is provided for you – **please DO NOT modify this file**.
- The use of the `random.randint(1,6)` function in order to simulate the roll of a six sided die.
- The use of a **list** in order to keep track of dice roll statistics (how many times each die face value was rolled) and **nested loops** in order to display the dice roll statistics to the screen.
- Output that **strictly** adheres to the assignment specifications. If you are not sure about these details, you should check with the 'Sample Output' provided at the end of this document or post a message to the discussion forum for clarification.
- Good programming practice:
  - Consistent commenting, layout and indentation. You are to provide comments to describe: your details, program description, all variable definitions, and every significant section of code.
  - Meaningful variable names (no single letter identifier names).

Your solution **MAY** make use of the following:

- Any of the Python built-in functions.
- Access the individual elements in a list with an index (one element only). i.e. `list_name[index]`.

Your solutions **MUST NOT** use:

- `break`, or `continue` statements in your solution. **Do not** use the `quit()` or `exit()` functions or the `break` or `return` statements (or any other techniques) as a way to break out of loops. Doing so will result in a significant mark deduction.
- Your own (user-defined) functions.

PLEASE NOTE: You are reminded that you should ensure that all input and output conform to the specifications listed here; if you are not sure about these details you should check with the sample output provided at the end of this document or post a message to the discussion forum in order to seek clarification.

Please ensure that you use Python 3.11.2 or a later version (i.e. the latest version) in order to complete your assignments. Your programs **MUST** run using Python 3.11.2 (or latest version).

STAGES (PART I)

It is recommended that you develop this part of the assignment in the suggested stages. Many problems in later stages are due to errors in early stages. **Make sure you have finished and thoroughly tested each stage before continuing.**

The following stages of development are recommended:

Stage 1

You will need the `dice.py` file for this assignment. This has been provided for you. Please download this file from the course website (Assessments tab) and ensure that it is in the same directory as the *yourEmailId*`_rose.py` file.

Test to ensure that this is working correctly by entering the following in your *yourEmailId*`_rose.py` file:

```
import dice


dice.display_dice(2,3,4,5,6)
```

Run the *yourEmailId*`_rose.py` file. If this is working correctly, you should now see the following output in the Python shell when you run your program:

```
Die 1     Die 2     Die 3     Die 4     Die 5
 [2]       [3]       [4]       [5]       [6]

 *         *         * *       * *       * *
            *                   *        * *
   *         *       * *       * *       * *
```

Note, this is for developmental purposes only, and you will need to modify and correctly position the above code.

Make sure the program runs correctly. Once you have that working, back up your program. *Note: When developing software, you should always have fixed points in your development where you know your software is bug free and runs correctly.*

Stage 2

Add code to simulate the rolling of five dice. Use the `random.randint(1,6)` function to simulate the roll of a die.

Modify your code to now display the randomly generated dice roll to the screen, for example, if your variables (that store the randomly generated die rolls) are called die1, die2, die3, die4, and die5, you would then modify the code from stage one so that the randomly generated values are passed to the function instead, like so:

```
# Place code to randomly generate the roll of five dice here…
:
:
dice.display_dice(die1, die2, die3, die4, die5)
```

Sample output (this will look different given we are generating random values here):

```
     Die 1     Die 2     Die 3     Die 4     Die 5
      [3]       [6]       [4]       [6]       [1]

      *         * *       * *       * *
       *        * *                 * *        *
        *       * *       * *       * *
```

Stage 3

Add code to prompt for and read the user’s guess (i.e. the score guess for the roll) and display an appropriate message to the screen depending on the user’s input. For example

- If the user guesses correctly, the following message is displayed:

  ```
  Well done! You guessed it!
  ```

- If the user enters an incorrect non-even number, the following message is displayed to the screen, for example:

  ```
  No sorry, it's 0 not 9. The score is always even.
  ```

- If the user enters an incorrect even number, the following message is displayed to the screen, for example:

  ```
  No sorry, it's 0 not 6.
  ```

  The numbers in the above output examples will change depending on what the user has entered and the score value for the roll.

  Sample output 1:

  ```
       Die 1     Die 2     Die 3     Die 4     Die 5
        [5]       [4]       [6]       [2]       [4]

        * *       * *       * *       *         * *
         *                  * *
        * *       * *       * *         *       * *


  Please enter your guess for the roll: 4

  Well done! You guessed it!
  ```

  Display the user’s guess, the actual score and appropriate message to the screen as seen below:

  Sample output 2:

  ```
       Die 1     Die 2     Die 3     Die 4     Die 5
        [3]       [6]       [3]       [4]       [1]

        *         * *       *         * *
         *        * *        *                   *
          *       * *         *       * *


  Please enter your guess for the roll: 7

  No sorry, it's 4 not 7. The score is always even.
  ```

Sample output 3:

```
     Die 1     Die 2     Die 3     Die 4     Die 5
      [4]       [5]       [5]       [2]       [6]

      * *       * *       * *       *         * *
                 *         *                  * *
      * *       * *       * *         *       * *



Please enter your guess for the roll: 4

No sorry, it's 8 not 4.
```

Stage 4
Now… it's time to allow the player to play more than one game. Let's add a loop that loops until the user either enters 'n' (to quit the game). Think about where this code should go – what needs to be repeated, etc.

Sample output:

```
Would you like to play Petals Around the Rose [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [3]       [2]       [4]       [1]       [5]

      *         *         * *                 * *
       *                             *         *
        *         *       * *                 * *



Please enter your guess for the roll: 6

Well done! You guessed it!


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [6]       [2]       [4]       [3]       [3]

      * *       *         * *       *         *
      * *                            *         *
      * *         *       * *         *         *



Please enter your guess for the roll: 3

No sorry, it's 4 not 3. The score is always even.


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [2]       [4]       [6]       [6]       [6]

      *         * *       * *       * *       * *
                          * *       * *       * *
        *       * *       * *       * *       * *
```

```
Please enter your guess for the roll: 8

No sorry, it's 0 not 8.


Roll dice again [y|n]? n
```

Stage 5

Add code to keep track of how many games were played, the number of correct guesses and the number of incorrect guesses. Display this to the screen as seen in the sample output.

Stage 6

Add code to keep track of how many correct and incorrect guesses are recorded in a row (consecutively) and display the appropriate messages to the screen (only if **three in a row** are achieved, i.e. must be consecutive) as seen in the sample output. Note, that the appropriate messages must be displayed to the screen every time **three consecutive** correct or incorrect guesses are recorded.

Stage 7

Add code to validate the following user input only:

- Would you like to play Petals Around the Rose [y|n]?

  Sample output:
  ```
  Would you like to play Petals Around the Rose [y|n]? w
  Please enter either 'y' or 'n'.

  Would you like to play Petals Around the Rose [y|n]? y
  ```

- Roll dice again [y|n]?

  Sample output:
  ```
  Roll dice again [y|n]? z
  Please enter either 'y' or 'n'.

  Roll dice again [y|n]? y
  ```

Stage 8

Add code to keep track of dice roll statistics. That is, how many times each die face value was rolled. Display this information to the screen (as seen in the sample output). You **MUST use a list in order to store this information**. You **MUST also use nested loops in order to display this information**.

To define a list in order to store dice roll stats:

`die_count = [0,0,0,0,0,0,0]`

`die_count`

0	0	0	0	0	0	0
0	1	2	3	4	5	6

Given that die face values are 1 – 6 inclusive, we create a list with seven elements but ignore the zero element of the list. This will make it easier to increment the appropriate list element.

To access and update the appropriate list element (using die value as an index):

```
die1 = random.randint(1,6)
die_count[die1] = die_count[die1] + 1
```

For example: If die1 is assigned the value 3.

die_count

0	0	0	1	0	0	0
0	1	2	3	4	5	6

Algorithm to display the frequency to the screen:

```
index = 1
WHILE index < length of die_count list
    display die face value to the screen (value of index)

    FOR number in 0 to die_count[index]
        display one star to the screen

    increment index by 1
```

Stage 9 – THIS IS IMPORTANT!

Finally, check the **sample output (see section titled 'Sample Output – Part I' towards the end of this document) and if necessary, modify your code so that:**

- The output produced by your program **EXACTLY** matches the sample output provided.
- Your program **EXACTLY** behaves as described in these specs **and** the sample output provided.

PART II SPECIFICATION – NUMBER CONVERTER

Write a **menu driven program** called `yourEmailId_converter.py` that will allow the user to enter commands and process these commands until the quit command is entered. Your program will accept entry of a number (in either decimal or binary notation) from the keyboard and convert it to either binary or decimal notation as requested.

Please note:
We will be working with two numeral systems only for this part of the assignment, decimal and binary. Decimal refers to Base 10 and works with ten digits, thus, decimal numbers use digits 0-9. Binary refers to Base 2 and works with two digits, thus, binary numbers use digits 0-1.

The following commands should be allowed:

1. **Convert to binary:**
   Prompt for and read (from the keyboard) a number (decimal whole numbers only) to be converted to binary. Convert the decimal number to the binary representation of the number as a string of 0s and 1s. Display the string (binary number) to the screen.

2. **Convert to decimal:**
   Convert a binary number to a decimal number by prompting for and reading (from the keyboard) a binary number (as a string). Convert the binary number to the decimal representation of the number (integer). Display the decimal number (integer) to the screen.

3. **Binary counting:**
   Prompt for and read a decimal number (whole numbers only). Display all decimal numbers starting from 1 up to and including the decimal number entered along with the binary representation of the numbers to the screen.

4. **Quit:**
   Quits the program displaying a goodbye message to the screen.

PRACTICAL REQUIREMENTS (PART II)

It is recommended that you develop this part of the assignment in the suggested stages.

It is expected that your solution WILL include the use of:

- Your solution in a file called *yourEmailId*_converter.py.
- Appropriate and well constructed while and/or for loops (as necessary).
- Appropriate if, if-else, if-elif-else statements (as necessary).
- The following four functions (refer to stage 6 for the description):
    - display_details()
    - get_menu_choice()
    - convert_to_binary(decimal_number)
    - convert_to_decimal(binary_number)
- You must use a loop(s) in your solution to convert from decimal to binary and vice versa.
- Output that **strictly** adheres to the assignment specifications. If you are not sure about these details, you should check with the 'Sample Output – Part II' provided at the end of this document.
- Good programming practice:
    - Consistent commenting, layout and indentation. You are to provide comments to describe: your details, program description, all variable definitions, all function definitions and every significant section of code.
    - Meaningful variable names.

Your solutions **MAY** make use of the following:

- Built-in functions int(), input(), print(), range(), pow(), len() and str().
- Concatenation (+) operator to create/build new strings.
- Access the individual elements in a string with an index (one element only). i.e. string_name[index] .
- Your own (user-defined) functions (in addition to the necessary four functions listed above).

Your solutions **MUST NOT** use:

- Built-in functions (other than the int(), input(), print(), range(), pow(), len() and str() functions).
- The built-in functions int() and/or bin() in order to convert between number systems, i.e., to perform the conversion from binary to decimal and vice versa.
- Slice expressions to select a range of elements from a string or list. i.e. name[start:end].
- String or list methods (other than those used for input validation, i.e. isdigit() and the append() method. i.e. *list_name*.append(item)).
- break, or continue statements in your solution. **Do not** use the quit() or exit() functions or the break or return statements (or any other techniques) as a way to break out of loops. Doing so will result in a significant mark deduction.

PLEASE NOTE: You are reminded that you should ensure that all input and output conform to the specifications listed here; if you are not sure about these details you should check with the sample output provided at the end of this document or post a message to the discussion forum in order to seek clarification.

Please ensure that you use Python 3.11.2 or a later version (i.e. the latest version) in order to complete your assignments. Your programs **MUST** run using Python 3.11.2 (or latest version).

STAGES (PART II)

It is recommended that you develop this part of the assignment in the suggested stages. Many problems in later stages are due to errors in early stages. **Make sure you have finished and thoroughly tested each stage before continuing.**

The following stages of development are recommended:

Stage 1
Implement the interactive mode (prompt for and read menu commands). Set up a loop to obtain and process commands. Test to ensure that this is working correctly before moving onto stage 2. You need not perform any number conversion at this point, you may simply display an appropriate message to the screen, for example:

Sample output:

```
*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 1
In command 1 - convert to binary

*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 2
In command 2 - convert to decimal

*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 3
In command 3 - binary counting

*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 4

Goodbye.
```

Make sure the program runs correctly. Once you have that working, back up your program. *Note: When developing software, you should always have fixed points in your development where you know your software is bug free and runs correctly.*

Stage 2
Add code to implement command 1. **Convert to binary.**
Prompt for and read a number (decimal notation) to be converted. Convert the decimal number to the binary representation of the number as a string of 0s and 1s. Display the string (binary number) to the screen. You **must** use a loop(s) in your solution.

Hint:
To convert a whole decimal number to binary, divide the decimal number by 2 recording the quotient and the remainder resulting from the division. Continue to divide each new quotient by 2 until you reach a quotient of 0 (remember that the quotient is the result of division). Write the remainders after each step of division (which will either be 1 or 0) off to the side. Since we are dividing by 2, when the dividend is even, the binary remainder will be 0 and when the dividend is odd, the binary remainder will be 1. Finally, write down the remainders in reverse order (with the last remainder written first). This result is the converted binary number.

For example, to convert the number 92_{10} to binary:

```
92 / 2 - quotient = 46 remainder = 0
46 / 2 - quotient = 23 remainder = 0
23 / 2 - quotient = 11 remainder = 1
11 / 2 - quotient = 5  remainder = 1
 5 / 2 - quotient = 2  remainder = 1
 2 / 2 - quotient = 1  remainder = 0
 1 / 2 - quotient = 0  remainder = 1
```

$92_{10} = 1011100_2$.

Stage 3
Add code to implement command 2. **Convert to decimal.**
Convert a binary number to a decimal number by prompting for and reading (from the keyboard) a binary number (as a string). Convert the binary number to the decimal representation of the number (integer). Display the decimal number (integer) to the screen. You **must** use a loop(s) in your solution.

Hint:
To convert a binary number to decimal, first list of powers of 2 from right to left (and their corresponding decimal values), for example:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Write the digits of the binary number below their corresponding powers of two. For example, to convert the number 1011100_2 to decimal:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
	1	0	1	1	1	0	0

Finally, add up all of the decimal values of the non-zero bits, i.e.: $64 + 16 + 8 + 4 = 92_{10}$.

Stage 4
Add code to implement command 3. **Binary counting.**
Prompt for and read a decimal number (whole number). Display all decimal numbers starting from 1 up to and including the decimal number entered along with the binary representation of the numbers to the screen. You **must** use nested loop(s) in your solution to achieve this.

Stage 5
Ensure that you validate all user input with an appropriate message being displayed for incorrect input entered by the user. You may like to make use of appropriate string method(s) in order to do this (this is the only place that you may make use of string methods. Hint, check the Python online docs for help with this: https://docs.python.org/3/library/stdtypes.html#string-methods).

Menu input should also be validated with an appropriate message being displayed if an incorrect menu command is entered by the user.

Add code to validate all user input. Hint: use a while loop to validate input.

Stage 6
Modify your code to include and make use of the following four functions:

- `display_details()`
- `get_menu_choice()`
- `convert_to_binary(decimal_number)`
- `convert_to_decimal(binary_number)`

- Write a function called `display_details()` that will display your details to the screen.
  The function takes no parameters and does not return any values. (Remember that defining the function does not execute the function – you will need to call the function from the appropriate place in the program). Your function should produce the following output (with your details).

  Output:
  ```
  File     : wayby001_converter.py
  Author   : Batman
  Stud ID  : 0123456X
  Email ID : wayby001
  This is my own work as defined by the University's Academic Misconduct Policy.
  ```

- Write a function called `get_menu_choice()` that displays the menu to the screen, prompts for, reads and validates the menu command entered by the user. The function takes no parameters and returns a valid menu command entered by the user.

- Write a function called `convert_to_binary(decimal_number)` that takes a decimal number (integer) as a parameter. The function converts the decimal number to the binary representation of the number as a string of 0s and 1s, i.e. into its binary number (represented as a string). The function returns the binary number (as a string). This function should be called for command 3 - binary counting.

- Write a function called `convert_to_decimal(binary_numbetr)` that takes a binary number (as a string) as a parameter. The function converts the binary number to the decimal representation of the number (integer). The function returns the decimal number (as an integer number).

Stage 7 – THIS IS IMPORTANT!
Finally, check the **sample output (see section titled 'Sample Output – Part II' towards the end of this document) and if necessary, modify your code so that:**

- The output produced by your program **EXACTLY** matches the sample output provided.
- Your program **EXACTLY** behaves as described in these specs **and** the sample output provided.

SUBMISSION DETAILS

You are required to do the following in order to submit your work and have it marked:

- Internal students:
  - You are required to submit an electronic copy of your program via learnonline **before Monday 24 April (week 7), 9am**.

All students must follow the submission instructions below:

Ensure that your files are named correctly (as per instructions outlined in this document).

Do NOT zip your files.

Ensure that the following files are included in your submission:

- `yourEmailId_rose.py`
- `yourEmailId_converter.py`

For example (if your name is James Bond, your submission files would be as follows):

- `bonjy007_rose.py`
- `bonjy007_converter.py`

All files that you submit must include the following comments.

```
#
# File: fileName.py
# Author: your name
# Email Id: your email id
# Description: Assignment 1 – place assignment description here…
# This is my own work as defined by the University's
# Academic Misconduct policy.
#
```

Assignments that do not contain these details may not be marked.

You must submit your program **before the online due date**. Work that has not been correctly submitted to learnonline will not be marked.

It is expected that students will make copies of all assignments and be able to provide these if required.

EXTENSIONS AND LATE SUBMISSIONS

There will be **no** extensions/late submissions for this course without one of the following exceptions:

1. A medical certificate is provided that has the timing and duration of the illness and an opinion on how much the student's ability to perform has been compromised by the illness. **Please note** if this information is not provided the medical certificate WILL NOT BE ACCEPTED. Late assessment items will not be accepted unless a medical certificate is presented to the Course Coordinator. The certificate must be produced as soon as possible and must cover the dates during which the assessment was to be attempted. In the case where you have a valid medical certificate, the due date will be extended by the number of days stated on the certificate up to five working days.

2. A Learning and Teaching Unit councillor contacts the Course Coordinator on your behalf requesting an extension. Normally you would use this if you have events outside your control adversely affecting your course work.

3. Unexpected work commitments. In this case, you will need to attach a letter from your work supervisor with your application stating the impact on your ability to complete your assessment.

4. Military obligations with proof.

Applications for extensions must be lodged via learnonline before the due date of the assignment.

Note: Equipment failure, loss of data, 'Heavy work commitments' or late starting of the course are not sufficient grounds for an extension.

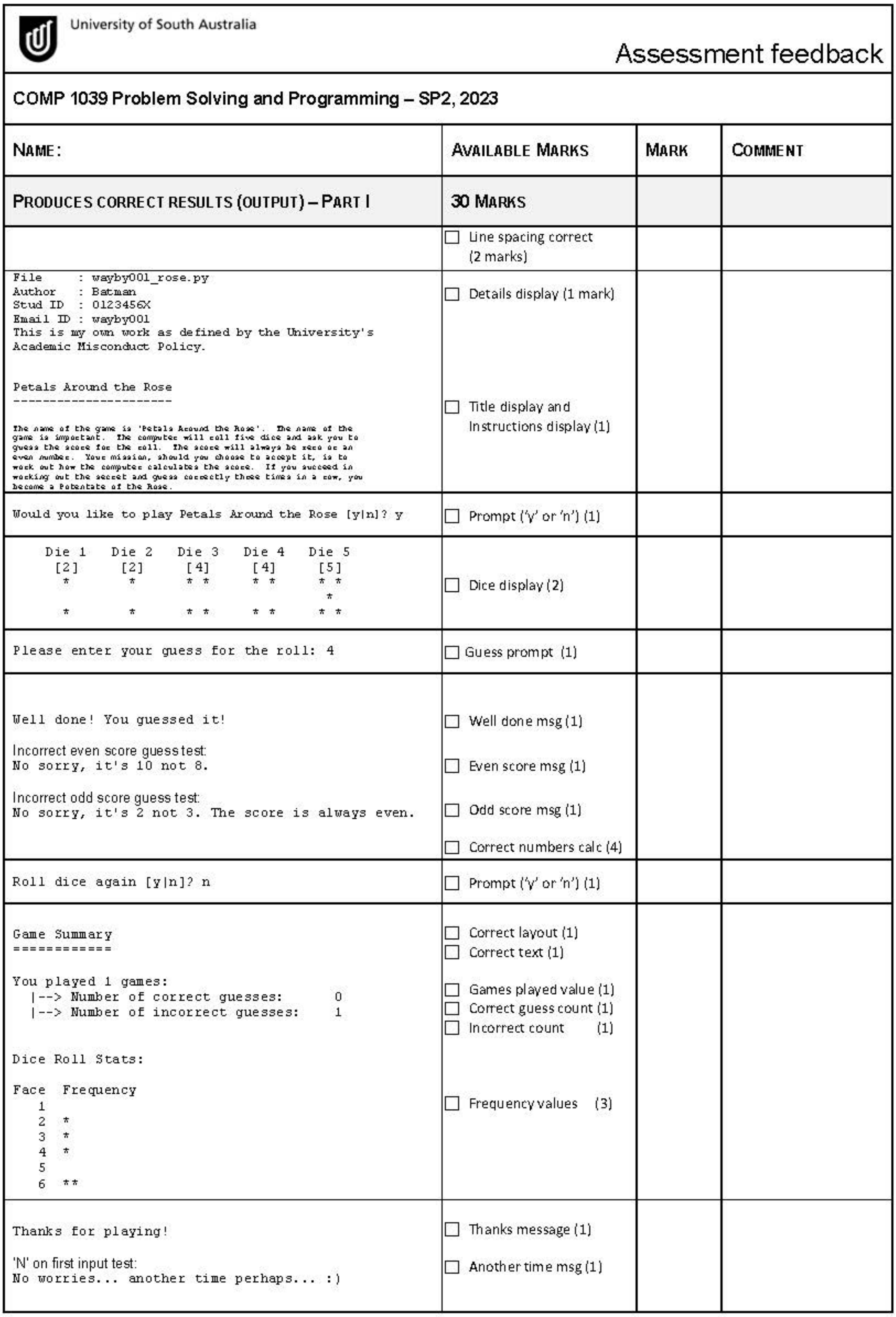
ACADEMIC MISCONDUCT

Students are reminded that they should be aware of what constitutes academic misconduct. Information about academic integrity and what constitutes academic misconduct can be found in the **Academic Integrity Policy and Procedure** (https://i.unisa.edu.au/policies-and-procedures/university-policies/academic/ab-69).

Deliberate academic misconduct (such as plagiarism, contract cheating, the use of ChatGPT (or similar tool)) is subject to penalties as outlined in the **Academic Integrity Policy and Procedure**.

You are **NOT** permitted to use ChatGPT (or other similar Artificial Intelligence tools) for your assessment items in this course. **Please note**: as stated in the University's Academic Integrity Policy (Policy AB-69), academic misconduct includes (but is not limited to) the use of artificial intelligence (AI) software or paraphrasing tools as a form of contract cheating. Please refer to the Academic Integrity Policy (Policy AB-69) for further information.

MARKING CRITERIA



University of South Australia

Assessment feedback

COMP 1039 Problem Solving and Programming – SP2, 2023

NAME:	AVAILABLE MARKS	MARK	COMMENT
PRODUCES CORRECT RESULTS (OUTPUT) – PART I	30 MARKS		
	□ Line spacing correct (2 marks)		
File : wayby001_rose.py Author : Batman Stud ID : 0123456X Email ID : wayby001 This is my own work as defined by the University's Academic Misconduct Policy. Petals Around the Rose ---------------------- The name of the game is 'Petals Around the Rose'. The name of the game is important. The computer will roll five dice and ask you to guess the score for the roll. The score will always be zero or an even number. Your mission, should you choose to accept it, is to work out how the computer calculates the score. If you succeed in working out the secret and guess correctly three times in a row, you become a Potentate of the Rose.	□ Details display (1 mark) □ Title display and Instructions display (1)		
Would you like to play Petals Around the Rose [y|n]? y	□ Prompt ('y' or 'n') (1)		
Die 1 Die 2 Die 3 Die 4 Die 5 [2] [2] [4] [4] [5] * * * * * * * * * * * * * * * * *	□ Dice display (2)		
Please enter your guess for the roll: 4	□ Guess prompt (1)		
Well done! You guessed it! Incorrect even score guess test: No sorry, it's 10 not 8. Incorrect odd score guess test: No sorry, it's 2 not 3. The score is always even.	□ Well done msg (1) □ Even score msg (1) □ Odd score msg (1) □ Correct numbers calc (4)		
Roll dice again [y|n]? n	□ Prompt ('y' or 'n') (1)		
Game Summary ============ You played 1 games: |--> Number of correct guesses: 0 |--> Number of incorrect guesses: 1 Dice Roll Stats: Face Frequency 1 2 * 3 * 4 * 5 6 **	□ Correct layout (1) □ Correct text (1) □ Games played value (1) □ Correct guess count (1) □ Incorrect count (1) □ Frequency values (3)		
Thanks for playing! 'N' on first input test: No worries... another time perhaps... :)	□ Thanks message (1) □ Another time msg (1)		

Three (3) incorrect guesses in a row test:
`Hint: The name of the game is important... Petals Around the Rose.`
Three (3) correct guesses in a row test:
`Congratulations! You have worked out the secret! Make sure you don't tell anyone!`
	□ 3 in row correct msg (1) and correct behaviour (1)
	□ 3 in row incorrect msg (1) and correct behaviour (1)		

ADHERES TO SPECIFICATIONS (CODE) – PART I		**COMMENT**
Use of `random.randint(1,6)` for die roll simulation		□ -1 No or incorrect use or randint()
While loop for play again (`play_again == 'y' or equivalent boolean expression`)		□ -2 No or incorrect while
Use of provided `dice.py` file and function `display_dice(d1,d2,d3,d4,d5).`		□ -2 No or incorrect use of display dice function
List to store die count information.		□ -2 No or -1 incorrect list for die count
Nested loops to display die count information to screen.		□ -2 No or -1 incorrect nested loops
Validation of user input messages – both (y/n) prompts: `Please enter either 'y' or 'n'.`		□ -2 No validation of user input
No user-defined functions.		□ -2 If using own functions
Good loops (i.e. no break, continue, return, goto, etc statements to exit loops). Must exit loop via Boolean expression.		□ -2 For using break/return/etc statements to exit loops

PRODUCES CORRECT RESULTS (OUTPUT) – PART II	30 MARKS		
	□ Line spacing correct (2 marks)		
File : wayby001_converter.py Author : Batman Stud ID : 0123456X Email ID : wayby001 This is my own work as defined by the University's Academic Misconduct Policy.	□ Details display (2 marks)		
*** Menu *** 1. Convert to binary 2. Convert to decimal 3. Binary counting 4. Quit What would you like to do [1,2,3,4]?	□ Correct menu display and prompt (4) -1 For each output/msg not to specs (up to 2 marks) -2 For incorrectly functioning prompt [1, 2, 3, 4]		
Convert to binary *** Menu *** 1. Convert to binary 2. Convert to decimal 3. Binary counting 4. Quit What would you like to do [1,2,3,4]? 1 Please enter number: 10 Binary number: 1010	□ Number prompt (1) □ Correct display layout (1) □ Correct binary no. (5)		
Convert to decimal *** Menu *** 1. Convert to binary 2. Convert to decimal 3. Binary counting 4. Quit What would you like to do [1,2,3,4]? 2 Please enter binary number: 1010 Decimal number: 10	□ Binary prompt (1) □ Correct display layout (1) □ Correct decimal no. (5)		
Binary counting *** Menu *** 1. Convert to binary 2. Convert to decimal 3. Binary counting 4. Quit What would you like to do [1,2,3,4]? 3 Please enter number: 10 Decimal: 1 = binary: 1 Decimal: 2 = binary: 10 Decimal: 3 = binary: 11 Decimal: 4 = binary: 100 Decimal: 5 = binary: 101 Decimal: 6 = binary: 110 Decimal: 7 = binary: 111 Decimal: 8 = binary: 1000 Decimal: 9 = binary: 1001 Decimal: 10 = binary: 1010	□ Number prompt (1) □ Correct decimal nums (1) □ Correct binary nums. (3) □ Correct text layout (2)		

`What would you like to do [1,2,3,4]? 4`
`Goodbye.`
	□ Goodbye msg (1)		

Adheres to specifications (code) – Part II		**Comment**
While/for loop for convert to binary		□ -2 No or incorrect loop
While/for loop for convert to decimal		□ -2 No or incorrect loop
While/for loop for binary counting (nested)		□ -2 No or incorrect nested loops
While loop for menu/prompt (`choice != 4 or equivalent boolean expression`)		□ -2 No or incorrect while
Should **not use** the following: • Built-in functions (other than int(), input(), print(), range(), pow(), len(), and str()) • int() and/or bin() to convert between number systems • Slice expressions • String or list methods (other than those used for input validation, i.e. isdigit() and the append() method).		□ -2 use of built-in functions not allowed □ -2 use of int() and/or bin() to convert □ -2 use of slicing i.e. [::-1] □ -2 use of string or list methods not allowed
Function `display_details()`		□ -2 No or -1 incorrect display function
Function `get_menu_choice()`		□ -3 No or -1 incorrect menu function
Function `convert_to_decimal(binary_number)`		□ -4 No or -1 incorrect convert function
Function `convert_to_binary(decimal_number)`		□ -4 No or -2 not called for binary counting or -1 incorrect convert to binary funct.

Validation of user input:
`Invalid choice, please enter either 1, 2, 3 or 4.`
`Please make sure your number contains digits 0-9 only.`
`Please make sure your number contains digits 0-1 only.`
`Please make sure your number contains digits 0-9 only.`
		□ -2 No validation of user input – menu
		□ -2 No validation of user input – binary
		□ -2 No validation of user input – dec
		□ -2 No validation of user input – count

Good loops (i.e. no break, continue, return, etc statements to exit loops). Must exit loop via Boolean expression.		□ -2 For using break/return/etc statements to exit loops

STYLE (BOTH PARTS)	5 MARKS	MARK	COMMENT
Comments: ○ your details at top of file ○ program description ○ all variable definitions ○ all function definitions ○ significant sections of code			□ -2 Insufficient comments ○ your details at top of file, ○ program description, ○ all variable definitions, ○ functions, and ○ significant sections of code
Consistent code layout and indentation.			□ -2 Inconsistent indentation and layout
Meaningful variable names (no single letter variable names).			□ -2 Non-descriptive variable names
TOTAL	**65 MARKS**		

The Graduate qualities being assessed by this assignment are indicated by an X:

X	GQ1: operate effectively with and upon a body of knowledge		GQ5: are committed to ethical action and social responsibility
	GQ2: are prepared for lifelong learning	X	GQ6: communicate effectively
	GQ3: are effective problem solvers		GQ7: demonstrate an international perspective
	GQ4:can work both autonomously and collaboratively		

This form meets the 2006 requirements of UniSA's Code of Good Practice: Student Assessment

Other possible deductions:

- *Programming style:* Things to watch for are poor or no commenting, poor variable names, etc.
- *Submitted incorrectly:* -10 marks if assignment is submitted incorrectly (i.e. not adhering to the specs).

SAMPLE OUTPUT (PART I)

Sample output 1:

```
File     : wayby001_rose.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Petals Around the Rose
----------------------

The name of the game is 'Petals Around the Rose'.  The name of the
game is important.  The computer will roll five dice and ask you to
guess the score for the roll.  The score will always be zero or an
even number.  Your mission, should you choose to accept it, is to
work out how the computer calculates the score.  If you succeed in
working out the secret and guess correctly three times in a row, you
become a Potentate of the Rose.

Would you like to play Petals Around the Rose [y|n]? n


No worries... another time perhaps... :)
```

Sample output 2:

```
File     : wayby001_rose.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Petals Around the Rose
----------------------

The name of the game is 'Petals Around the Rose'.  The name of the
game is important.  The computer will roll five dice and ask you to
guess the score for the roll.  The score will always be zero or an
even number.  Your mission, should you choose to accept it, is to
work out how the computer calculates the score.  If you succeed in
working out the secret and guess correctly three times in a row, you
become a Potentate of the Rose.

Would you like to play Petals Around the Rose [y|n]? p
Please enter either 'y' or 'n'.

Would you like to play Petals Around the Rose [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [4]       [3]       [2]       [5]       [5]

      * *       *         *         * *       * *
                 *                   *         *
      * *         *         *       * *       * *



Please enter your guess for the roll: 4

No sorry, it's 10 not 4.


Roll dice again [y|n]? z
Please enter either 'y' or 'n'.

Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [6]       [1]       [1]       [2]       [4]

      * *                           *         * *
      * *        *         *
      * *                             *       * *
```

```
Please enter your guess for the roll: 3

No sorry, it's 0 not 3. The score is always even.


Roll dice again [y|n]? n


Game Summary
============

You played 2 games:
  |--> Number of correct guesses:      0
  |--> Number of incorrect guesses:    2


Dice Roll Stats:

Face  Frequency
   1  **
   2  **
   3  *
   4  **
   5  **
   6  *

Thanks for playing!
```

Sample output 3:

```
File     : wayby001_rose.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Petals Around the Rose
----------------------

The name of the game is 'Petals Around the Rose'.  The name of the
game is important.  The computer will roll five dice and ask you to
guess the score for the roll.  The score will always be zero or an
even number.  Your mission, should you choose to accept it, is to
work out how the computer calculates the score.  If you succeed in
working out the secret and guess correctly three times in a row, you
become a Potentate of the Rose.

Would you like to play Petals Around the Rose [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [3]       [1]       [5]       [1]       [2]

      *                   * *                 *
       *         *         *         *
        *                 * *                   *



Please enter your guess for the roll: 6

Well done! You guessed it!


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [6]       [3]       [5]       [4]       [6]

      * *       *         * *       * *       * *
      * *        *         *                  * *
      * *         *       * *       * *       * *



Please enter your guess for the roll: 2

No sorry, it's 6 not 2.
```

```
Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [5]       [4]       [6]       [4]       [3]

      * *       * *       * *       * *       *
       *                  * *                  *
      * *       * *       * *       * *         *



Please enter your guess for the roll: 6

Well done! You guessed it!


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [1]       [6]       [1]       [1]       [2]

                * *                           *
       *        * *        *         *
                * *                             *



Please enter your guess for the roll: 0

Well done! You guessed it!


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [6]       [3]       [2]       [1]       [1]

      * *       *         *
      * *        *                   *         *
      * *         *         *



Please enter your guess for the roll: 2

Well done! You guessed it!


Congratulations! You have worked out the secret!
Make sure you don't tell anyone!


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [4]       [3]       [1]       [5]       [3]

      * *       *                   * *       *
                 *         *         *         *
      * *         *                 * *         *



Please enter your guess for the roll: 7

No sorry, it's 8 not 7. The score is always even.


Roll dice again [y|n]? n


Game Summary
============

You played 6 games:
  |--> Number of correct guesses:      4
  |--> Number of incorrect guesses:    2


Dice Roll Stats:

Face  Frequency
   1  ********
   2  ***
   3  ******
   4  ****
   5  ****
   6  *****

Thanks for playing!
```

Sample output 4:

```
File     : wayby001_rose.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Petals Around the Rose
----------------------

The name of the game is 'Petals Around the Rose'.  The name of the
game is important.  The computer will roll five dice and ask you to
guess the score for the roll.  The score will always be zero or an
even number.  Your mission, should you choose to accept it, is to
work out how the computer calculates the score.  If you succeed in
working out the secret and guess correctly three times in a row, you
become a Potentate of the Rose.

Would you like to play Petals Around the Rose [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [4]       [2]       [4]       [5]       [4]

      * *       *         * *       * *       * *
                                     *
      * *         *       * *       * *       * *



Please enter your guess for the roll: 4

Well done! You guessed it!


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [3]       [6]       [3]       [1]       [5]

      *         * *       *                   * *
       *        * *        *         *         *
        *       * *         *                 * *



Please enter your guess for the roll: 8

Well done! You guessed it!


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [4]       [5]       [6]       [5]       [2]

      * *       * *       * *       * *       *
                 *        * *        *
      * *       * *       * *       * *         *



Please enter your guess for the roll: 7
```

```
No sorry, it's 8 not 7. The score is always even.


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [2]       [4]       [4]       [2]       [1]

      *         * *       * *       *
                                               *
        *       * *       * *         *



Please enter your guess for the roll: 1

No sorry, it's 0 not 1. The score is always even.


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [2]       [4]       [1]       [3]       [2]

      *         * *                 *         *
                           *         *
        *       * *                   *         *



Please enter your guess for the roll: 8

No sorry, it's 2 not 8.


Hint: The name of the game is important... Petals Around the Rose.


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [2]       [3]       [1]       [5]       [1]

      *         *                   * *
                 *         *         *         *
        *         *                 * *



Please enter your guess for the roll: 6

Well done! You guessed it!


Roll dice again [y|n]? n


Game Summary
============

You played 6 games:
  |--> Number of correct guesses:       3
  |--> Number of incorrect guesses:     3


Dice Roll Stats:

Face  Frequency
   1  *****
   2  *******
   3  ****
   4  *******
   5  *****
   6  **

Thanks for playing!
```

Sample output 5:

```
File     : wayby001_rose.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Petals Around the Rose
----------------------

The name of the game is 'Petals Around the Rose'.  The name of the
game is important.  The computer will roll five dice and ask you to
guess the score for the roll.  The score will always be zero or an
even number.  Your mission, should you choose to accept it, is to
work out how the computer calculates the score.  If you succeed in
working out the secret and guess correctly three times in a row, you
become a Potentate of the Rose.

Would you like to play Petals Around the Rose [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [6]       [3]       [2]       [5]       [3]

      * *       *         *         * *       *
      * *        *                   *         *
      * *         *         *       * *         *



Please enter your guess for the roll: 8

Well done! You guessed it!


Roll dice again [y|n]? y


     Die 1     Die 2     Die 3     Die 4     Die 5
      [6]       [3]       [2]       [5]       [2]

      * *       *         *         * *       *
      * *        *                   *
      * *         *         *       * *         *



Please enter your guess for the roll: 4

No sorry, it's 6 not 4.


Roll dice again [y|n]? n


Game Summary
============

You played 2 games:
  |--> Number of correct guesses:      1
  |--> Number of incorrect guesses:    1


Dice Roll Stats:

Face  Frequency
   1
   2  ***
   3  ***
   4
   5  **
   6  **

Thanks for playing!
```

SAMPLE OUTPUT (PART II)

Sample output 1:

```
File     : wayby001_converter.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 1

Please enter number: 10

Binary number: 1010


*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 2

Please enter binary number: 1010

Decimal number: 10


*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 3

Please enter number: 10

Decimal: 1 = binary: 1
Decimal: 2 = binary: 10
Decimal: 3 = binary: 11
Decimal: 4 = binary: 100
Decimal: 5 = binary: 101
Decimal: 6 = binary: 110
Decimal: 7 = binary: 111
Decimal: 8 = binary: 1000
Decimal: 9 = binary: 1001
Decimal: 10 = binary: 1010


*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 4

Goodbye.
```

Sample output 2:

```
File     : wayby001_converter.py
Author   : Batman
Stud ID  : 0123456X
```

```
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 9
Invalid choice, please enter either 1, 2, 3 or 4.

What would you like to do [1,2,3,4]? 1

Please enter number: seven
Please make sure your number contains digits 0-9 only.

Please enter number: 1t2
Please make sure your number contains digits 0-9 only.

Please enter number: 7

Binary number: 111


*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 2

Please enter binary number: 12110
Please make sure your number contains digits 0-1 only.

Please enter binary number: 1t10
Please make sure your number contains digits 0-1 only.

Please enter binary number: 111

Decimal number: 7


*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 3

Please enter number: 12

Decimal: 1 = binary: 1
Decimal: 2 = binary: 10
Decimal: 3 = binary: 11
Decimal: 4 = binary: 100
Decimal: 5 = binary: 101
Decimal: 6 = binary: 110
Decimal: 7 = binary: 111
Decimal: 8 = binary: 1000
Decimal: 9 = binary: 1001
Decimal: 10 = binary: 1010
Decimal: 11 = binary: 1011
Decimal: 12 = binary: 1100


*** Menu ***

1. Convert to binary
2. Convert to decimal
3. Binary counting
4. Quit

What would you like to do [1,2,3,4]? 4

Goodbye.
```