**Task**

**Your task is to implement a documented Python program which allows a user to play a slot machine game. The program will:**

* **Display a set of rules for the game (welcome function)**
* **Enable the user to continue spinning the slot wheels or quit the game (main function)**
* **Output the number of tokens left (play\_slots function)**
* **Output the randomly generated three slot results (play\_slots function)**
* **Output if the slot results win tokens (play\_slots function)**
* **Force the game to terminate if all the tokens are gone (main function)**

**Each time the program is executed, the user is allocated 5 tokens.**

**A spin of the slot machine costs 1 token.**

**Winning slots will result in tokens being added to the tokenCount variable.**

Start by creating a new Python file called **slots\_game.py**

**The User Interface and program logic –**

The **main()** function controls the flow of the program, it starts by initialising the tokenCount variable to 5, then calling the welcome() function. After the welcome() function it enters a loop where it calls the play\_slots() function an undetermined number of times, until either the users quits the program, or the tokenCount causes the program to terminate.

The Welcome screen is displayed, showing the rules of the game. This information should be displayed from a function named **welcome()** as seen in the screenshot below:

Welcome to the FDSE Slots Game

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Each spin of the slot machine costs 1 token

You can quit the game at any time by entering <Q>

The game is automatically over when all tokens have been spent

Winning:

&&& wins 3 tokens

$$$ wins 4 tokens

£££ wins 5 tokens

3 symbols all different in any order wins 1 token

Press <CR> to continue...

After pressing the carriage return <CR> the game starts:

Your code should return to the main() function and then call a **play\_slots()** function from within a repetition construct, passing over the tokenCount value.

You have 4 token(s) left

................

| $ | $ | £ |

................

\*\*\*No Win this Spin\*\*\*

Spin again <CR> or Quit (Q):

As seen from the screen shot above, the play\_slots function will deduct 1 from the tokenCount variable and display the token count on screen.

**The play\_slots function will then generate 3 random numbers (in the range 1 to 3), where a 1 is represented by a “&”, a 2 is represented by a “$”, and a 3 is represented by a “£”.**

Examples of generated values:

**1, 2, 3 is generated then “& $ £” should be displayed**

**2, 1, 1 is generated then “$ & &” should be displayed**

**3, 3, 3 is generated then “£ £ £” should be displayed**

Please note, as identified in the Welcome screen above, only 3 random numbers generated that are all the same or all different, will result in tokens being won and added to the tokenCount.

You have 4 token(s) left

................

| & | £ | $ |

................

\*\*\* Congratulations you won 1 token \*\*\*

Spin again <CR> or Quit (Q): **<CR>**

You have 4 token(s) left

................

| £ | £ | £ |

................

\*\*\* Congratulations you won 4 tokens \*\*\*

Spin again <CR> or Quit (Q): **<CR>**

You have 7 token(s) left

................

| & | & | $ |

................

\*\*\*No Win this Spin\*\*\*

Spin again <CR> or Quit (Q):

The screenshot above shows the user has pressed <CR> twice.

After each spin the play\_slots function **returns** the updated tokenCount back to the main() function.

As seen in the screenshot above, play continues until the user enters a Q or the user runs out of tokens.

Any wins are added to the tokenCount within the play\_slots function.

If the user still has tokens left, they will be given the choice of continuing to spin or quit the game.

If the user has used up all their tokens, the game will automatically end.

\*\*\*No tokens left\*\*\*

Thank you for playing.

**Notes** The play\_slots function receives 1 integer parameter (tokenCount)

The play\_slots function returns 1 value (tokenCount)

Your program should display **exactly** what has been specified (see interface examples).

Ensure your code is commented throughout.

Ensure the program is sufficiently tested, to show tokens being added and tokens being   
deducted. Also show the user exiting from the game using the Q option.

lot