Programming Task: Riddles

A riddle is a puzzle in the form of a question. Examples of riddles include:

```
Question:  What has 4 eyes but cannot see?
Answer:    Mississippi

Question:  What will break when you say its name?
Answer:    Silence

Question:  The more you take, the more you leave behind. What are they?
Answer:    Footsteps
```

Your task is to write a program that asks the user a riddle.

Your program should:

- Ask the user a riddle.
- Keep asking the riddle until the user answers correctly
- Keep a count of how many attempts the user has entered the answer.
- If the user has guessed the riddle in 3 guesses or fewer it should congratulate them with the message "Well done, you solved my riddle within three guesses"

Where you are asked to give evidence of your program you should print out or produce a screenshot of your program. You should check your program works at every stage but only need to give proof of testing in section (f). You should ensure you use appropriate commenting in your code.

(a) Make a list what your riddle program needs to be able to do. Plan the program you are going to make. (You can use flowcharts, pseudo code or any other sensible method of showing how your program will work.)

(b) Write a program that asks the user a riddle and allows them to enter their answer.

Test your program works then give evidence of your code.

(c) Add to your program so it keeps asking the user the riddle while the answer the user gives is wrong.

Test your program works then give evidence of your code.

(d) Add to you program so it uses a variable to keep count of the number attempts the user has had to answer the riddle.

Test your program works then give evidence of your code.

(e) Add to your program so once the user has answered correctly it says "Well done you solved my riddle within three guesses" if the user has guessed it in three attempts.

Test your program works then give evidence of your code.

(f) Test your program works. Make notes on how you tested it and what you found out.

(g) Write an evaluation of how well your program works. You should include:

- Any problems you had making it and how you overcame them.
- Any bugs your program still has.
- Any future improvements that could be made to your program.