Lab 4 – Flow Control

Part 1: Write a number guessing game.

You know this game; you'll enter a number between 0 and 100 that has to equal a number in the program (also between 0 and 100). If you guess too high, program writes "Your guess of <whatever> is too high; if too low, "Your guess of <whatever> is too low." You have 5 tries; after each guess the program tells you how many you have left.

If your guess is outside the range (0-100, in this case), program tells you "Your guess of <whatever> is invalid" and re-prompts for the number.

If you use up your guesses, the program writes "You have no more guesses left" - NOT you have 0 guesses left.

When done (guessed or not guessed), program asks if you wanna play again.

Set an integer = 45 as the *computer's guess* (we didn't talk about generating random numbers yet)

Below is a sample run:

```
Wanna play the number guessing game? (y or n) ==> y
Enter a number between 1 and 100 ==> 50
50 is too high
You have 4 guesses left.

Enter a number between 1 and 100 ==> 85
85 is too high
You have 3 guesses left.

Enter a number between 1 and 100 ==> 45
You found the number 45!
And it took you 3 guesses to find it

Wanna play the number guessing game? (y or n) ==> n
```

Wording need not be exact.

Part 2: Code a simple menu

Write a program that displays a menu for C (create), R (read), U(update), D(delete) or Q(quit). The output should resemble:

```
Enter:
    C to create,
    R to read,
    U to update,
    D to delete or
    Q to quit ==> c
Calling CREATE routine..…
Enter:
    C to create,
    R to read,
    U to update,
    D to delete or
    Q to quit ==> t
Your entry T is invalid - try again
Enter:
    C to create,
    R to read,
    U to update,
    D to delete or
    Q to quit ==> r
Calling READ routine..…
Enter:
    C to create,
    R to read,
    U to update,
    D to delete or
    Q to quit ==> q
Exiting program
```