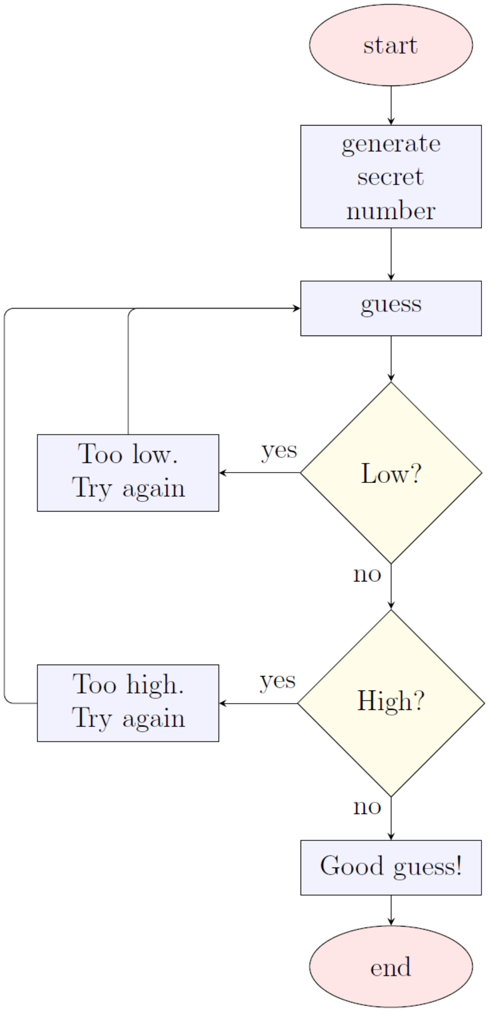
day4: a number guessing program
Due: 7th: Wednesday 8/23/23

Today's task is to make a number guessing program. The program should have a secret number, then ask the user to guess the number, over and over, ending the loop when the user enters the correct number.

Let's talk about how to do this. The following is a simple flow chart that describes how you might structure your program:



A flowchart can be helpful in terms of getting yourself organized. It is not a step by step command by command version of your program, but instead a big-picture view of the main things that are going to need to be done.

To make this program a reality, you'll need to do some things in each section as noted above. Your code won't literally be called "Start", or "End", these are just concepts in the diagram.

Go to the next page to start working on today's assignment.

Create a new repl named day4. Start by storing a secret number in a variable (you pick the number), then you'll do a while True: loop.

Inside the loop you will need to ask the user to guess the secret number. You need to use the input() function to do this, and also the int() function to turn the user entry into an integer.

After you have the guess stored in a variable you need to compare the guess with the secret number. If they are the same, congratulate the user and exit the while loop using a break command. If the user guessed too high, tell them. If they guessed too low, tell them. Then the loop starts over.

That's one way to write this program. You certainly can do it another way if you prefer. Jump in and give it a try.

Name today's program day4 number guess and turn in a Repl Invite link on the Google Classroom.

Extensions (do these if you finish before the period is over):

1. Make the secret number be a random number. To do this, add this line up at the top of your program:

```
import random
```

Then, on the line where you were storing a specific number into your secretNumber variable, do this:

```
secret = random.randint(1,100)
```

2. Count how many guesses it took the user to get the right answer

Create a count variable near the top of the program (before the while True: loop) and put zero into it. Each time the user makes a guess, before you check if the guess is correct, or too low or too high, add 1 to the count variable. Then, when the game is over (below the while loop) print how many guesses the user needed to get it right like this:

```
print("You took", count, "guesses to win.")
```

3. Add commentary that congratulates the user if they win in 5 or fewer guesses and makes fun of the user if they win in over 5 guesses. Use two if statements to do this.

4. Tell the user how far away they are from the right answer if they guess too low or too high. Of course, if you told them exactly how far away they were, it would make the game too easy. So be vague, like this:

```
if secret - guess > 5 or secret - guess < -5:
   print("You are more than 5 away from the secret number.")
```

The above code prints that message if the guess is more than 5 away either direction (too high or too low).

No matter what, check your program to make sure it works. Run it more than once, make sure the prompts are correct (that when your guess is too low, it tells you that it was too low, etc.) Make sure that when you get the right answer, it tells you that you won and tells you how many guesses it took.