LAB 2 - Loops

Let's write a program to ask the user for an integer, and then print out whether that number is EVEN or ODD. If we divide an even number by 2, what will the remainder be?

```
def main():
  x = input("Input an integer: ")
  if x % 2 == 0:
    print "EVEN"
  else:
    print "ODD"
```

Recall that, in Python, `7 % 3` gives us the remainder when 7 is divided by 3, which is 1.

Now suppose we wish to change our program so that it asks the user to enter 8 integers (one at a time), and then prints out how many of those integers were even numbers. For example, if the user enters the following integers

19, 6, 9, 20, 13, 7, 6, 1

then our program should print out 3, since 3 of those numbers were even.

Clearly, our program should use a loop that runs 8 times. And one thing we know we need to do 8 times is to ask the user to enter an integer. We can write this in Python as:

```
for times in range(8):
  x = input("Input an integer: ")
```

What will our program need to remember as it runs? It does not need to remember all the numbers that have been entered so far. But it will need to remember how many of the numbers entered so far have been even. Let's call this `count`, since we're counting how many values are even. We should initialize count to be 0, because at the beginning of the program, zero even integers have been entered. At the end of the program, `count` should represent the total number of even integers entered.

```
count = 0
for times in range(8):
  x = input("Input an integer: ")
print count, "integers were even"
```

How should the value of `x` (an integer entered by the user) influence the value of `count` (the total number of even integers entered so far)? Suppose `count` is 2, and the user then enters the integer 7. Because 7 is odd, `count` should not change. On the other hand, if the user now enters 6 (an even number), `count` should increment from 2 to 3, representing that now 3 even integers have been entered so far.

We can increment `count` by finding the value of `count` + 1 and storing this result in `count`.

```
count = count + 1
```

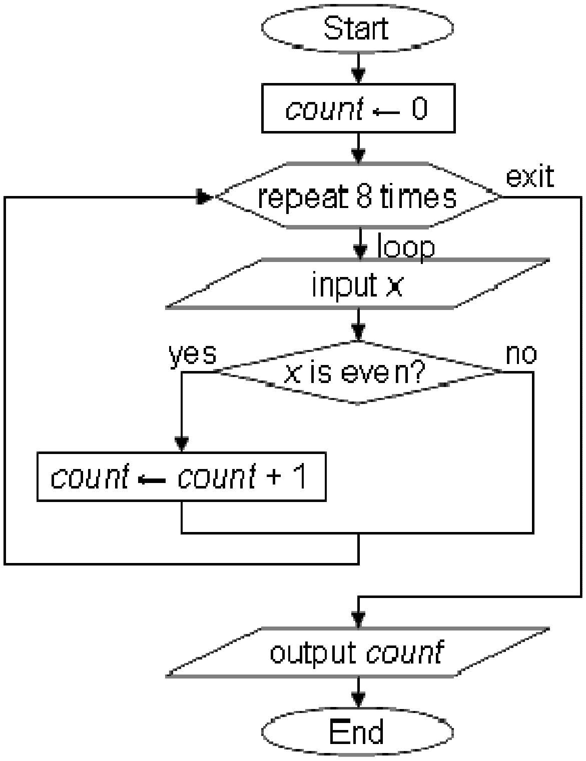
This behavior (incrementing `count`) should only run *if* `x` (the entered integer) is even.

```
if x % 2 == 0:
   count = count + 1
```

And we should perform this test each time the user enters an integer, so our final program reads as follows.

```
def main():
   count = 0
   for times in range(8):
      x = input("Input an integer: ")
      if x % 2 == 0:
         count = count + 1
   print count, "integers were even"
```

Here's a flowchart depicting our algorithm:


Start
count ← 0
repeat 8 times
exit
loop
input x
yes
x is even?
no
count ← count + 1
output count
End