Project 3 CS 261

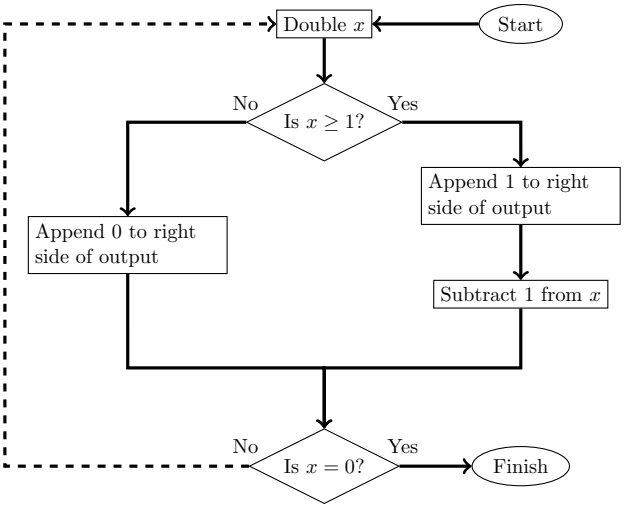
Save your program for this project as `<emailID>_project2.py` *where* `<emailID>` *is the part of your Hampden-Sydney e-mail address before the @ symbol. For example, I would save my program as* `blins_project1.py`. *When you are finished, e-mail your program to* `blins@hsc.edu`. *Your solution is due by noon on Friday, September 20.*

Binary Representation of Decimals

Floating point errors happen because floating point numbers are stored in base-2, which isn't a perfect match for our number system. For example, in Python

```
>>> 0.1 + 0.1 + 0.1
0.30000000000000004
```

In this project we will implement an algorithm to convert any nonnegative floating point number x that is less than 1 into a string of ones and zeros that represents its binary decimal form. A flow chart for the algorithm is shown below.



1. Write a function called `decimal_to_binary(x)` to implement this algorithm.
2. This algorithm will run forever for most inputs. Your function should have some way to prevent this and eventually print out at least the first 10 digits if the binary string doesn't terminate.
3. Test your function on these examples.

   ```
   decimal_to_binary(0.75) # should get 0.11
   decimal_to_binary(0.1)  # should get 0.0001100110
   ```

4. Use a similar algorithm to create a function `decimal_to_ternary(x)` that converts x to a ternary (base-3) string.

   ```
   decimal_to_ternary(0.5)  # should get 0.1111111111
   decimal_to_ternary(0.25) # should get 0.0202020202
   ```

   To adjust the old algorithm, you'll need to triple x at each stage, and then there will be three possibilities: if $x \geq 2$, if $2 > x \geq 1$, and $x < 1$.