

Assignment 3

— Problem Solving —

Algorithmic Thinking

- Title: Really Easy
 - My solution works - Forward(1) Left(1) Forward(4)
 - There are multiple solutions to every problem
-

- Title: Function Calls
 - My solution works - Right(Blue 1) Forward (1) F2 (Left 1 Forward 2 Repeat)
 - There are tons of other answers. I could have flipped the directions.
-

- Title: Combination 2
 - My solution works - Forward(2) Right(1) Forward(1) Left(1) F2(F3(forward) F1)
 - There are many other answers, I didn't need to invoke F3 at all, but decided to for a laugh
-

Problem Solving Steps

- Comments about the problem to aid in understanding it
 - How do we know the list is in order
 - How large are the numbers going to be
 - How do we want the numbers to look on the output

- Do we want only the largest number
- Description of knowledge base
 - I would like to know for loops
 - I would like to know if statements
- What are the various solutions? Weigh pros and cons and pick one.
 - We could use a for loop to go through and just update it if it's the largest, issue is it would go linearly through the dataset, wasting time if the true max is found early.
 - We could also use a Randomized Quickselect (based on Quicksort) which would be much faster on larger datasets as there is intelligent movement to narrow it down.
 - We could just use python's max() function which makes things easy.
- Test cases
 - Have numbers ranging from 0 to 256
 - Have all of the numbers be the same
 - Have all numbers be the
 - Have all numbers be the min
 - Have the numbers in reverse max size (largest first then down)
 - Have the numbers already sorted correctly

How would the problem change?

- The program shouldn't change if it's written well. Using a for loop it wouldn't change. Using quicksort it wouldn't change. It would work fine in certain situations. It would get slower if it's larger in a dataset.

Program

```
1 input_number = input("Please input a number to convert to binary: ")
2 current_number = []
3 print(current_number, input_number, input_number % 2)
4
5 current_number.insert(0,int(input_number % 2))
6 input_number = input_number / 2
7 print(current_number, input_number, input_number % 2)
8
9 current_number.insert(0,int(input_number % 2))
10 input_number = input_number / 2
11 print(current_number, input_number, input_number % 2)
12
13 current_number.insert(0,int(input_number % 2))
14 input_number = input_number / 2
15 print(current_number, input_number, input_number % 2)
16
17 current_number.insert(0,int(input_number % 2))
18 input_number = input_number / 2
19 print(current_number, input_number, input_number % 2)
20
21 current_number.insert(0,int(input_number % 2))
22 input_number = input_number / 2
23 print(current_number, input_number, input_number % 2)
24
25 current_number.insert(0,int(input_number % 2))
26 input_number = input_number / 2
27 print(current_number, input_number, input_number % 2)
28
29 current_number.insert(0,int(input_number % 2))
30 input_number = input_number / 2
31 print(current_number, input_number, input_number % 2)
32
33 current_number.insert(0,int(input_number % 2))
34 input_number = input_number / 2
35 print(current_number, input_number, input_number % 2)
```

Alternate

```
1 | import sys
2 | input_number = int(sys.argv[1])
3 | current_number = []
4 | for i in range(8):
5 |     current_number.insert(0,int(input_number % 2))
6 |     input_number = input_number / 2
7 |
8 | print(current_number.)
```