

# Big Program - Illustrating The Binary Representation of a Floating Point Number

---

Things we will need:

- ① knowledge of int, float, string.
- ② knowledge of functions, both ones that we write, and built-in functions that are part of the int, float, and string classes.
- ③ knowledge of how floating point #'s are stored in Python.
- ④ Loops (for, while)
- ⑤ Branching (if-else)

⑥ Knowledge of how to get input from the user.

⑦ Knowledge of how to output information in a formatted way. (print)

---

## Program Design.

Step 1: Get a floating point # from the user.

```
n = input("Enter your floating point value : \n")
```

→ note: n will be a string !!!

Step 2: Get how many decimal places we want for the

places  
binary "decimal" representation

i.e.  $0.1_{10} = 0.0001100110011 \dots$

How many ???

$P = \text{int}(\text{input}(\text{"Enter the number of decimal places of The result : (n)"}))$

→ note:  $P$  will be an int !!!

Step 3: Write a function which takes  $n, p$  as inputs, and returns the binary "decimal" representation as a string.

What is the algorithm that this function will use?? To figure this out, let's imagine doing it by hand. and the steps involved.

it by example, to 6 decimal places.

Ex.

3.125

to 6 decimal places.

Step 1: Separate the two parts, before and after the decimal.

$$3 + 0.125$$

Step 2: Convert the part in front of the decimal to binary.

$$3 \rightarrow 11$$

(Multiple steps here!!)

Step 3: Convert the part after the decimal to binary.

$$0.125 \rightarrow 0.\overbrace{001000}^{\text{six places.}}00\dots$$

$\uparrow \quad \uparrow \quad \uparrow$   
 $\frac{1}{2} \quad \frac{1}{4} \quad \frac{1}{8}$

(Multiple steps!!)

Step 4: Construct a string out of these results.

$$\text{res} = "11" + "." + "001000"$$

$$11$$



$$3.125_{10} = 11.001000_2 \quad \text{to 6 decimal places.}$$

Step 2, broken down

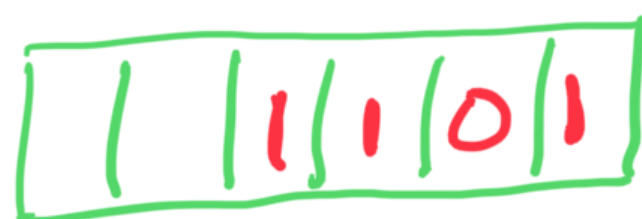
$$3_{10} \rightarrow 11_2$$

Algorithm:

Use integer division + mod function.

Ex.

$$13_{10} = 1101_2$$



$$13 / 2 = 6$$

$$13 \% 2 = 1$$

$$6 / 2 = 3$$

$$6 \% 2 = 0$$



$$3 / 2 = 1$$

$$3 \% 2 = 1$$

$$1 / 2 = \boxed{0} \text{ Done}$$

$$1 \% 2 = 1$$

$$\text{num} = 13$$

$$\text{res} = ""$$

while (num > 0):

$$\text{digit} = \text{num} \% 2$$

$$\text{res} = \text{str}(\text{digit}) + \text{res}$$

$$\text{num} = \text{num} / 2$$

print(res)

---

Now that we know how to do this,  
let's save it as a function:

```
def convert_to_binary(num):  
    {
```

return res

---

Step 3, broken down  $\rightarrow$

$$0.125 \rightarrow \frac{001000}{6 \text{ places.}}$$

Again, let's do it by hand:

$$0.95_{10} \quad \begin{array}{l} \text{if } (num > 0.5) \end{array} \quad \begin{array}{l} \text{Y} \rightarrow \boxed{1} \rightarrow .95 - .5 = .45 \\ \text{N} \rightarrow 0 \rightarrow \text{do nothing.} \end{array}$$

$$0.45_{10} \quad \begin{array}{l} \text{if } (num > 0.25) \end{array} \quad \begin{array}{l} \text{Y} \rightarrow \boxed{1} \rightarrow .45 - .25 = .20 \\ \text{N} \rightarrow 0 \rightarrow \text{do nothing.} \end{array}$$

$$0.20_{10} \quad \begin{array}{l} \text{if } (num > 0.125) \end{array} \quad \begin{array}{l} \text{Y} \rightarrow \boxed{1} \rightarrow .20 - .125 = .075 \\ \text{N} \rightarrow 0 \rightarrow \text{do nothing.} \end{array}$$

$$\boxed{1} \rightarrow .075 - .0625 = .0125$$

0.075<sub>10</sub>

if (nm > 0.0625)   
 Y → 1   
 N → 0 → do nothing.   
 ⋮

---

nm = 0.95

p = 6

count = 0

res = ""

while nm > 0 and count < p;

comparison = 2<sup>count</sup> \* (-1 - count)

if (nm > comparison):

res = res + "1"

nm = nm - comparison

else:

res = res + "0"

count = count + 1

print (res)

---



Again, now that we have this bit working, write it as a function!!

```
def convert-fraction-to-binary(num, p):  
    {  
  
    return res
```

---

Now, put steps 1, 2, 3, 4 together....

See `get-binary-rep.py`

in Week 1 examples.

---

OK, now we have the fractional binary representation of a real floating point decimal number.

Next thing: Get the

2<sup>n</sup> bit floating point binary

representation !!

$$1.\overbrace{0011011}^{\text{mantissa}} \times 2^{\overbrace{101}^{\text{exponent}}}$$

Step 5 : Determine the mantissa and exponent.

---