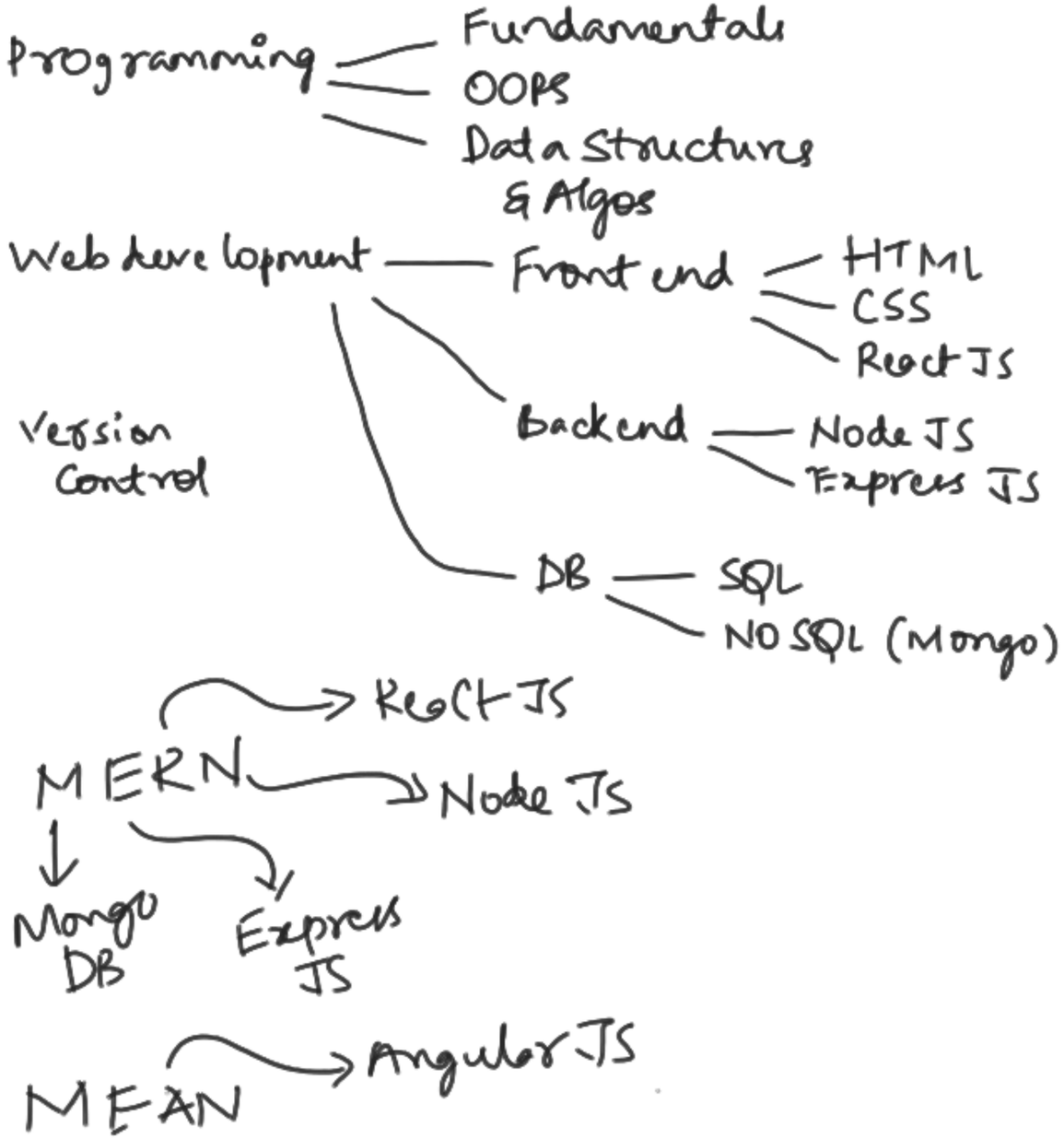
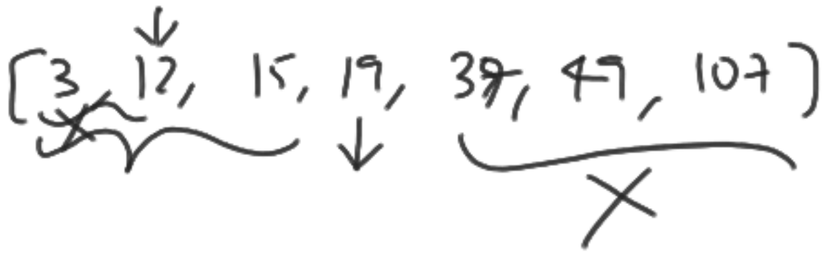
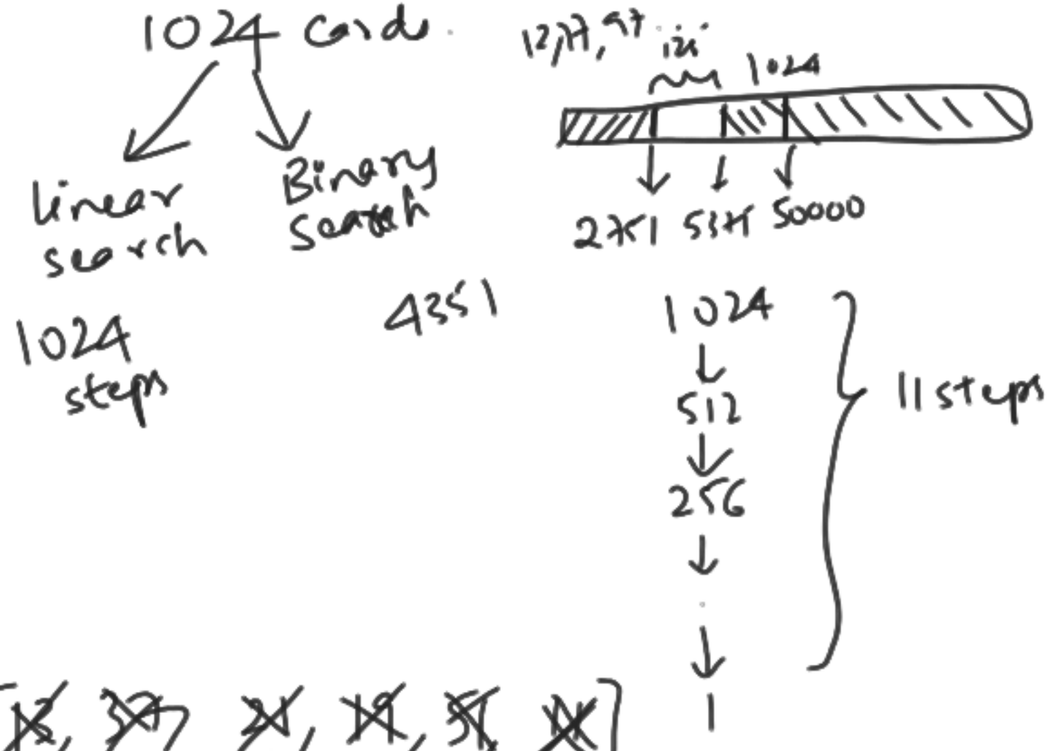
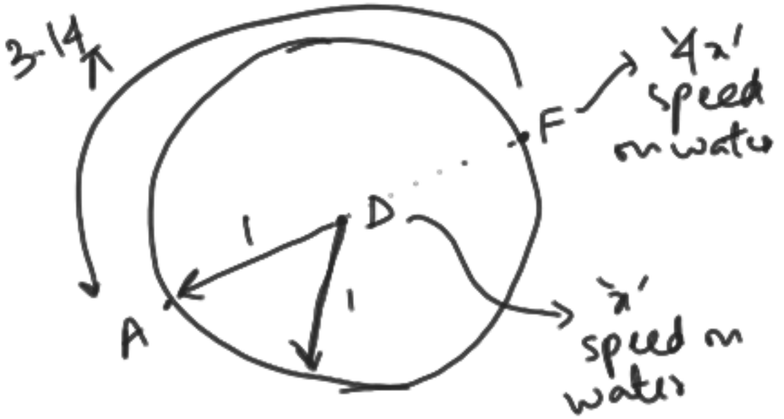
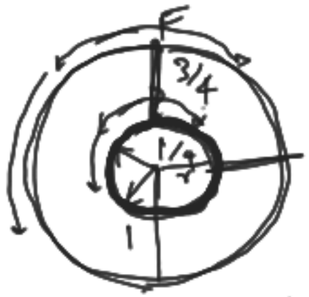

Programming
Fundamentals
OOPS
Data Structures & Algos
Web development
Front end
HTML
CSS
ReactJS
Backend
Node JS
Express JS
Version Control
DB
SQL
NO SQL (Mongo)
MERN
ReactJS
Node JS
Mongo DB
Express JS
MEAN
AngularJS


[3, 12, 15, 19, 39, 79, 107]


1024 cards
linear search
Binary Search
1024 steps
4351
12,17,97
1024
2751
5311
50000
1024
512
256
1
11 steps
[11, 13, 19, 21, 37, 51]


3.14
F
'4x' speed on water
D
1
1
A
'x' speed on water


M
F
Y



$r = 1/4 \quad r = 1$

$s = d/t \quad \boxed{t = d/s}$

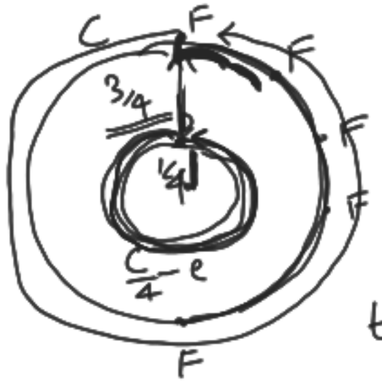
distance by duck $= 2\pi r$
$= 2\pi(\frac{1}{4})$
$= \pi/2$

distance by fox $= 2\pi r$
$= 2\pi$

$t_{duck} = \frac{dist}{speed}$
$= \frac{\pi/2}{x}$
$= \pi/2x$

$t_{fox} = \frac{dist}{speed}$
$= \frac{2\pi}{4x} = \frac{\pi}{2x}$

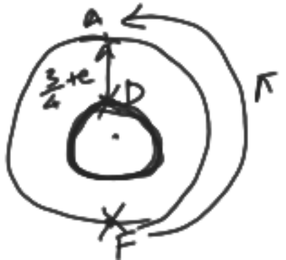
$r \quad 2r$
$\downarrow \quad \downarrow$
$\boxed{2\pi}r \quad 4\pi r$



10 s
9.9 s

$t_d = \frac{\frac{3}{4}+e}{x} = \boxed{\frac{3}{4x}} + \frac{0.001}{x}$

$t_f = \frac{\pi}{4x} = \boxed{\frac{3.14}{4x}}$



$t_{fox} = \frac{\pi}{4x}$
$= \frac{3.14}{4x}$

$t_{duck} = \frac{\frac{3}{4}+e}{x}$
$= \boxed{\frac{3}{4x}}$

if, if-else, if-else if-else

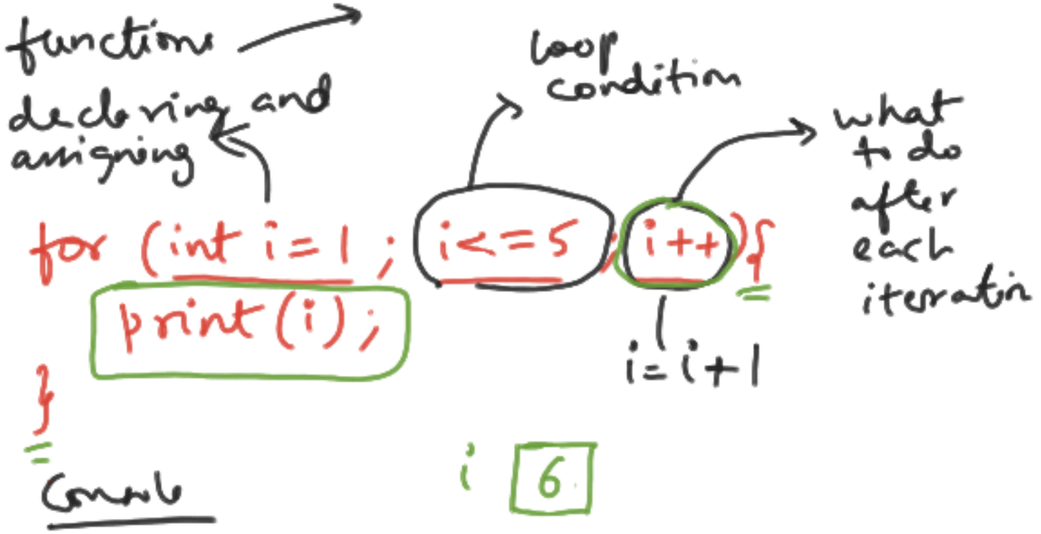
assignment

for

while

functions

declaring and assigning



i 6

Console

1
2
3
4 5

```
int n=3; int p=1;
for (int i=1; i<=n;i=i+1){
    p = p * i;
}
```

n 3

p 6

i 4

i 2

j = 1

#
##
###
####

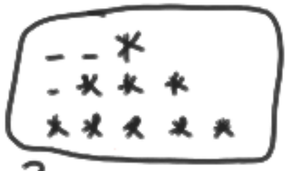
n=5

j=1	4_	1*
j=2	3_	3*
j=3	2_	5*
j=4	1_	7*
j=5	0_	9*

5-j spaces 2j-1 stars

n=3

j=1



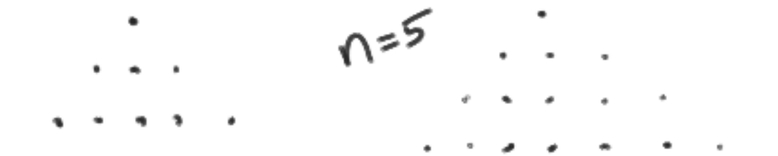
n=5

j=1	0_	3*
j=2	1_	3*
j=3	2_	3*
j=4	3_	3*
j=5	4_	3*

j-1 spaces 3 stars

n=7

j=1	6_	1#
j=2	5_	2#
j=7	0_	7#



n=3

j=1 2_ 1*
j=2 1_ 3*
j=3 0_ 5*

3-j spaces, 2j-1 stars

n=5

j=1 4_ 1*
j=2 3_ 3*
j=3 2_ 5*
j=4 1_ 7*
j=5 0_ 9*

5-j spaces, 2j-1 stars

n=2

j=1 0_ 3*
j=2 1_ 3*

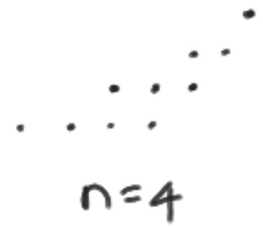
j-1 spaces, 3 stars

n=5

j=1 0_ 3*
j=2 1_ 3*
j=3 2_ 3*
j=4 3_ 3*
j=5 4_ 3*

j-1 spaces, 3 stars

n=4

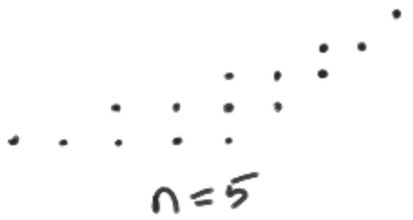


n=4

j=1 6_ 1*
j=2 4_ 2*
j=3 2_ 3*
j=4 0_ 4*

'j' stars

2(4-j) spaces

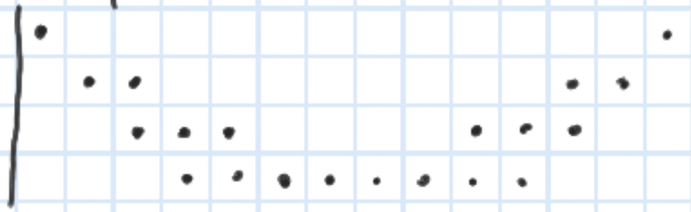


n=5

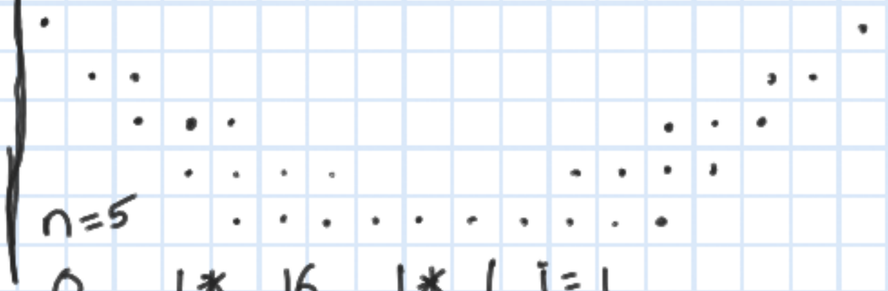
j=1 8_ 1*
j=2 6_ 2*
j=3 4_ 3*
j=4 2_ 4*
j=5 0_ 5*

2(5-j) spaces 'j' stars

n=4



n=5



0_	1*	16_	1*	j=1
1_	2*	12_	2*	j=2
2_	3*	8_	3*	j=3
3_	4*	4_	4*	j=4
4_	5*	0_	5*	j=5

'j-1' spaces 'j' stars 'j' stars

4n-4j
4(n-j) spaces


n=4


n=5


n=4


n=5

n = 3

```
    1
  2 2 2
3 3 3 3 3
```

n = 5

```
      1
    2 2 2
  3 3 3 3 3
4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5
```

n = 3

```
1
1 2 3
1 2 3 4 5
```

n = 5

```
1
1 2 3
1 2 3 4 5
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9
```

n = 3

```
    1
  1 2
1 2 3
```

n = 5

```
        1
      1 2
    1 2 3
  1 2 3 4
1 2 3 4 5
```

```
while (loop condition) {
    // code inside
}
```
(boolean)

```
int i=1;
while (i<=10){
    print("hello");
    i++;
}
```

```
for (int i=1; i<=10; i++){
    print("hello");
}
```

break & continue

```
for (int i=1; i<=15; i++) {
    println("loop: "+ i);
    if (i==3) {
        break;
    }
    println("hello");
} println("end");
```

```
loop: 1
hello
loop: 2
hello
loop: 3
end
```

```
for (int i=1; i<=6; i++){
    println("loop: "+i);
    if (i%2==0){
        continue;
    }
    println("hello");
}
println("end");
```

```
loop: 1
hello
loop: 2
loop: 3
hello
loop: 4
loop: 5
hello
loop: 6
end
```

n = 5

j = 1 — 1, 4 '1's
j = 2 — 1 2, 3 '1's
j = 3 — 1 2 3, 2 '1's
j = 4 — 1 2
j = 5 — 1 2 3 4 5, 0 '1's

print (1, 2, ..., j)
1 .. j , "n - j" 1s

n = 5

j = 1 — 1 to 1 ≡, ~~four~~ five 1s
j = 2 — 1 to 2 ≡, three 1s
j = 3 — 1 to 3 ≡, two 1s
j = 4 — 1 to 4 ≡, one 1
j = 5 — 1 to 5 ≡, zero 1

5 4 3 2 1
5 4 3 2
5 4 3
5 4
5

n = 5

j = 1 — 5 to 1
j = 2 — 5 to 2
j = 3 — 5 to 3
j = 4 — 5 to 4
j = 5 — 5 to 5

'n' to 'j'

543

```
for (i=5; i>=3; i--){
    print(i);
}
```

```
i=5;                    I
while (i >= 3) {        II
    print(i);           code
    i--;                III
}
print(i);
```

i 2

5432

```
i = 5;
while (true) {
    print(i);
    if (i==3) {
        break;
    }
    i--;
}
print(i);
```

i 3

5433

```
int i = 0;
while (i < 15){
    if (i % 2 == 0){
        println("hello");
    }
    println(i);
    if (i % 3 == 0){
        println("world");
        i++;
        continue;
    }
    if (i == 10) {
        break;
    }
    i += 2;
}
println(i);
```

i 10

```
hello
0
world
1
3
world
hello
4
hello
6
world
7
9
world
hello
10
10
```


n 5
j 6
i 1
s 6

6
1
6

```
public class Main {
    public ... main() {
        int x=10;
        println(x);
        if (true) {
            if(x>10) {
                print("hello");
            }
            print("there");
        }
    }
}
```

```
s = -3;
if (s < 0) {
    s = -1 * s;
}
// code
....
if (s<0) {
    print('-');
} else {
    print('+');
}
```

s 3

i 32

j -1

0 some 5 more

0 some 72 more

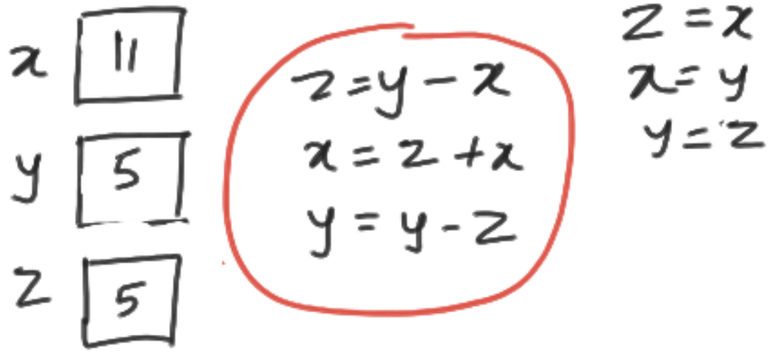
0 10 some 116 more

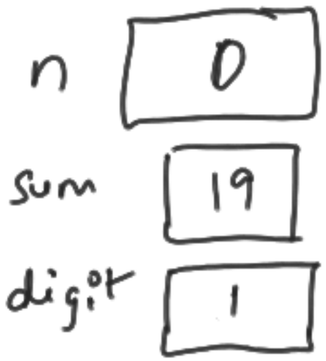
0 10 20 30 some 1914 9 more

0 10 20 30 40 50 60 some 35 30 25 20 more

end

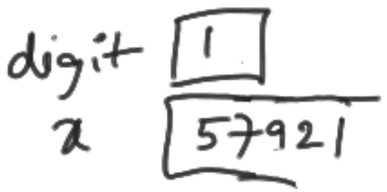
i	0	4
j		





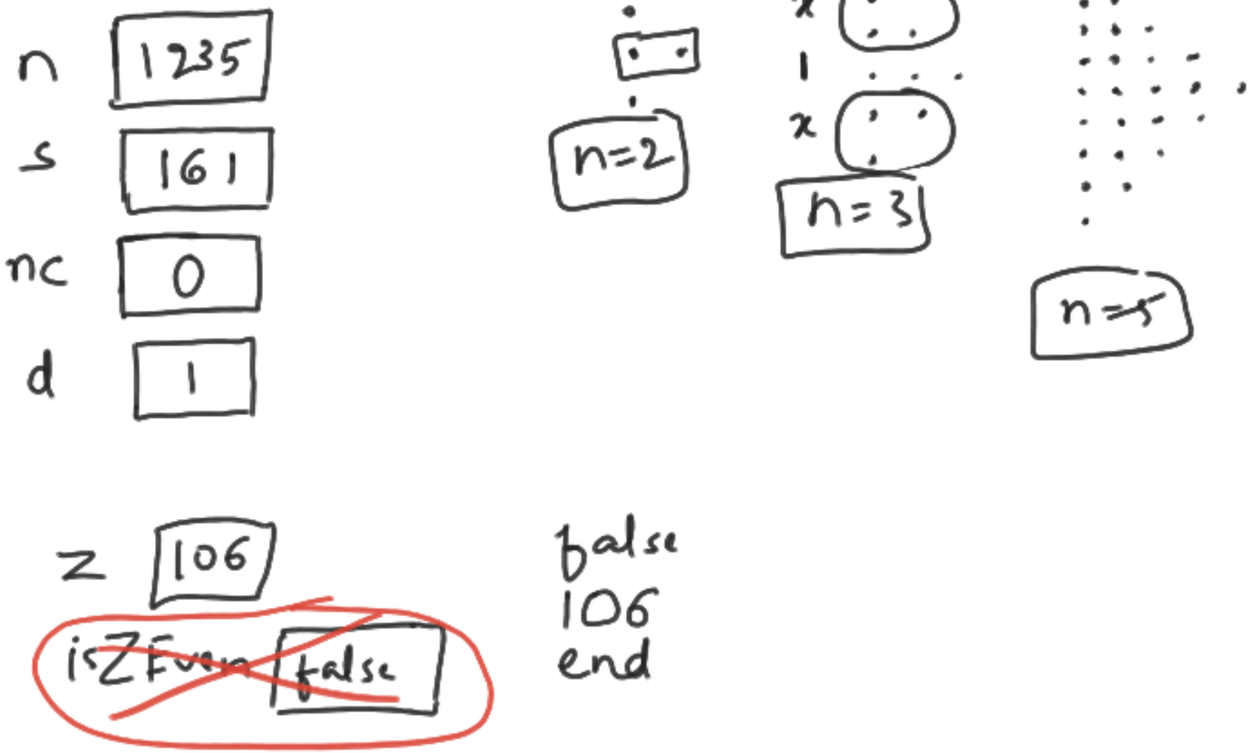
n = 12975

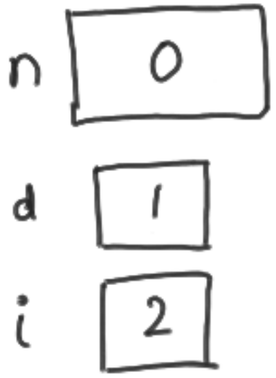
x = 5792



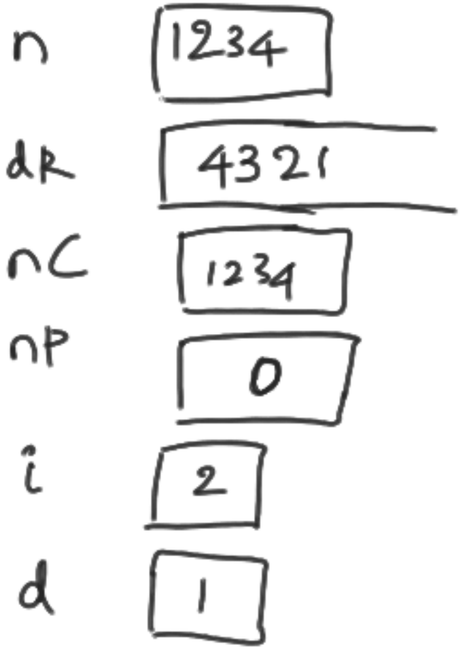
$x = 10x + d$

1. get unit digit of 'n' ($d = n\%10$)
2. Add digit at end of 'x' ($10x + d$)
3. Remove unit digit of 'n' ($n = n/10$)


n
1235
s
161
nc
0
d
1
n=2
x
1
x
n=3
n=5
z
106
isZEven
false
false
106
end



4321



dR = -4

nC − 123

d = −123 %10

= −3

dR = 10 * dR + digit