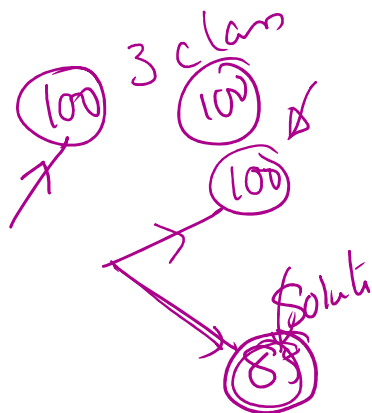




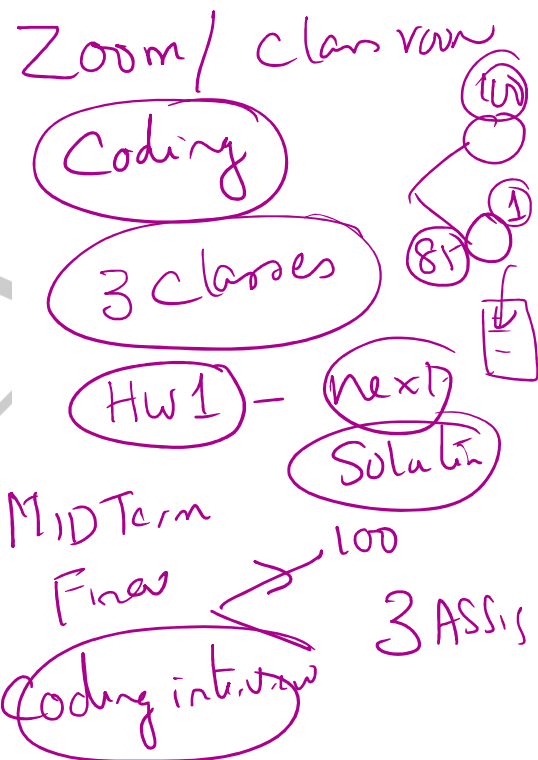
2f 1 TA
↑ Observer

2/3 hr →
3 weeks



Letter	Total Marks
A	95.0 - 100
A-	94.9 - 90
B+	89.9 - 85
??	< 85

100



OOP PYTHON

C, C++, JAVA

32 bit / 64 bits

Memory

8 bit

A handwritten diagram illustrating memory storage. A large purple oval contains the text "int a = 75". An arrow points from the variable "a" to a smaller purple oval below it, which contains the value "75".

unt- a = 75

$$\frac{n}{1}$$
$$\frac{\text{number}}{2}$$

0	1	2	3

10

vol-4

0 to 15

MSB 3 2 1 0 LSB

	3	2	1	0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	0	0	0
13	1	0	0	1
14	1	1	1	1
15	1	1	1	0

4 bit

15

$$2^4 - 1 = 15$$

Alg

$$\text{uml-} \textcircled{d=75}$$

0100

010000

0000001101

$\rightarrow a = 13$

D2B (mit n)

$$\begin{array}{r}
 2 \overline{) 13} \begin{array}{l} \swarrow 6 \\ \searrow 7 \end{array} \\
 \underline{12} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 2 \overline{) 6} \begin{array}{l} \swarrow 3 \\ \searrow 3 \end{array} \\
 \underline{6} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 2 \overline{) 3} \begin{array}{l} \swarrow 1 \\ \searrow 1 \end{array} \\
 \underline{2} \\
 1
 \end{array}$$

OK

1 1 0 1

$$\begin{array}{r} 2) 3(1 \\ \underline{2} \\ 1 \\ 2) 1(0 \\ \underline{0} \\ 1 \end{array}$$

(C++)

int- a = 13 C, java, C++

a = 13

a = 1

4 bytes 1 32
0000101
a

4 bytes
1
a

13

(C) / C++
class int {

?

7

$$\begin{cases} \text{unt-} a = 15 \\ a = 13: \end{cases}$$

1000!

$a = 15$
 $a = 13$

STACK

heat object -
Charge
col

Python module

32
2 - 1

$$\text{unt- } a = \underline{\hspace{2cm}}$$
$$\begin{array}{r} 1 \overline{) 232} \\ \underline{2} \\ 32 \\ \underline{32} \\ 0 \end{array}$$

$a = 593189512195$

class IntA

PYTHON

Point:

Class object

Class int'l

7

06

$$\{ \text{int-} a = \frac{2^5 - 1}{3} \}$$

Start()

WS 11

Objekt

PRIVS CAR

Q = Priv(1)
& stat[1-1]

~~6~~ = Prim()
 b. Start()

$C = \frac{1}{P_{\text{min}}}$

Clan

Private()

3

a = first object

b

C

July

A handwritten diagram illustrating the relationship between different components. At the top left is the word "Class". To its right is a large curly brace. Below "Class" is the word "Public". To the right of "Public" is another curly brace. To the right of this second brace are the words "Start()" and "Stop" stacked vertically. A line descends from the "Public" section, passing between the "Start()" and "Stop" sections, and ending at a large oval at the bottom containing the word "Private".

int

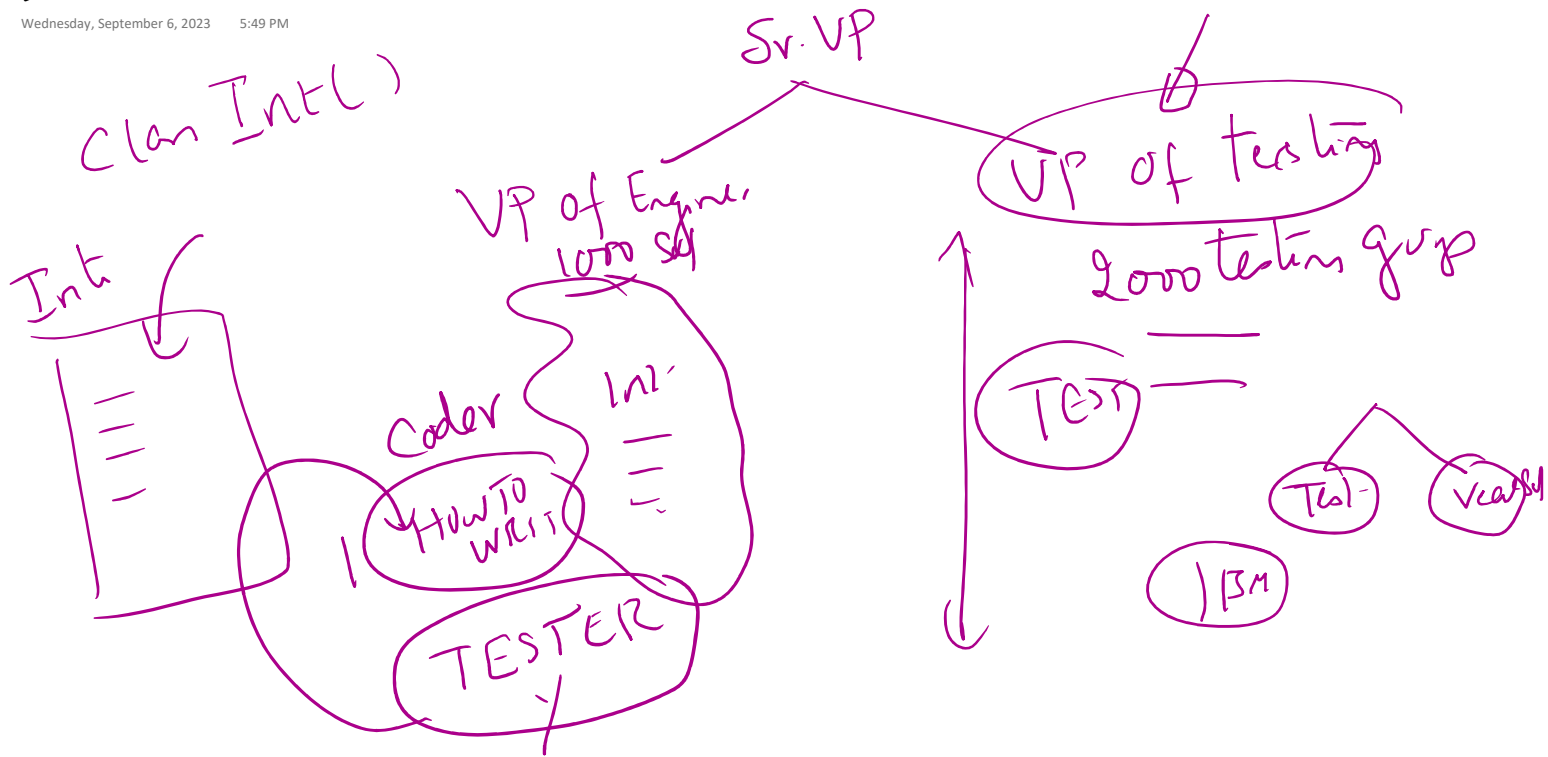
Class & Public
Private:

Class Object
Form

Ag:
DOP.

```
a = From()
a.name =
a.Dob =
```

$$b = f_{\text{min}}$$



✓

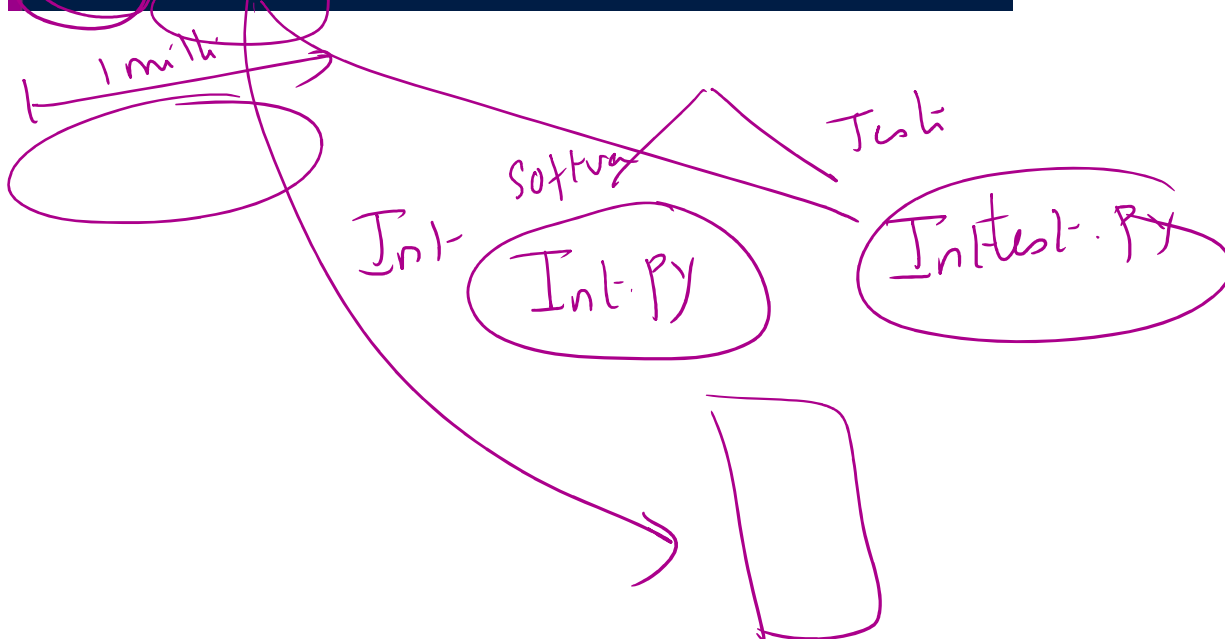
$$a > b$$
$$a > b$$

unt

① class

② TESTER

```
C:\Users\jag\OneDrive\vasu\work\py3\objects\Int\dir>ls
Int.py  Inttest.py  __pycache__  dir.pyproj  dir.sln
```



```

print(sys.version)
t = Inttest()
init("D")

```

Construct

__init__

DS + Alg + Code

```

class Inttest:
    def __init__(self):
        self._why_are_we_building()
        self._data_structure()
        self._access()
        self._arithmetic()
        self._relational()

```

Alg

Exp

a = 75

a
STACK

75
Heap

t

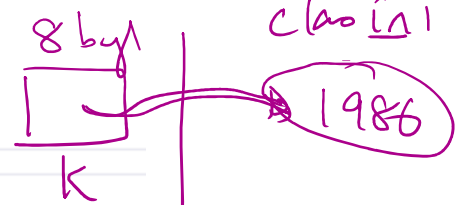
Public

Private ~~Private~~

```

def _why_are_we_building(self):
    print("-----Testing idea-----")
    k = 1986
    print("type of k", type(k), "Value of k =", k, "id =", id(k))
    k = 19861234567890123456789013456789134678
    print("type of k", type(k), "Value of k =", k, "id =", id(k))
    print(
        "You cannot do in c or java: int k = 9861234567890123456789013456789134678"
    )
    # print(k[3])
    print("You cannot get 3 digit which is 6 using k[3]")
    print("----- idea Passed -----")

```



Int

```

-----Testing idea-----
type of k <class 'int'> Value of k = 1986 id = 2233512931792

```

64bit-La

223512

class int

1986

int c =



0 1 2 3
a = 1 9 8 6
a[2] 8

get(unl-x²)

0	—	1
1	—	9
2	—	8
3	—	6

```
class Int:
    def __init__(self, n: "Python int" = 0):
        # ONLY DATA STRUCTURE ALLOWED
        # self._positive
        # self._a
        self._positive = True
        if n < 0:
            self._positive = False
        self._a = self.build(n)
```

build(n)
-a =

Class
PYTHON
obj
a
b
c
d

TYPE

class Int

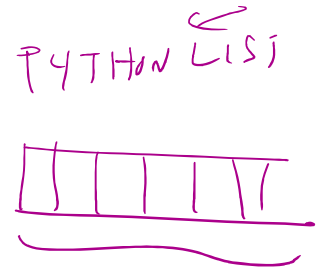
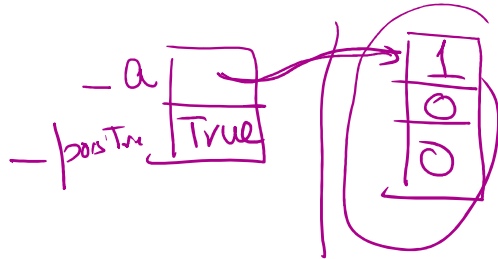
Int

```
a = Int()
print(a)
a = Int(100)
print(a)
a = Int(-100)
print(a)
n = -1009876545678654234222
a = Int(n)
```

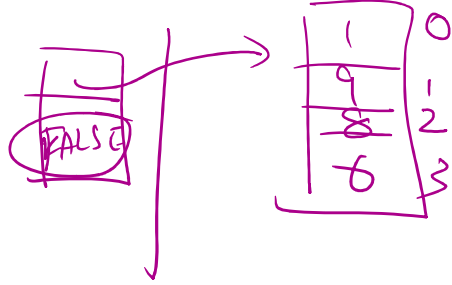
Obj - class
→ a = Int(100) a = Int(23)

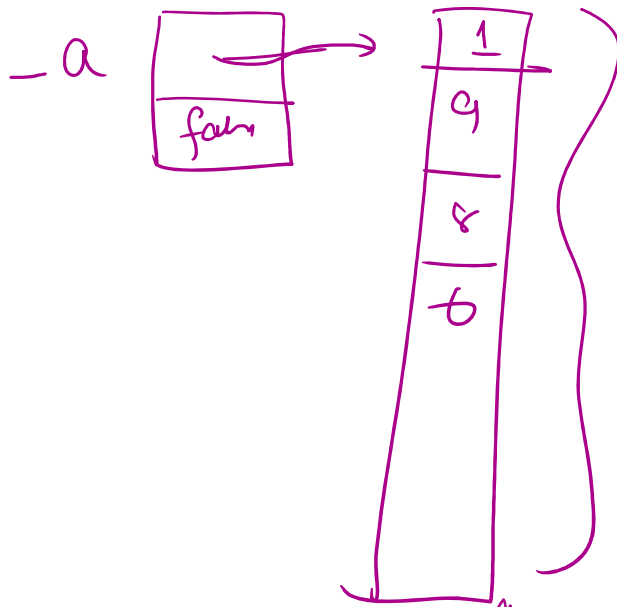


Dynamic LIST
class Int
ARRAY



a = Int(100)
a = Int(-1986)





1986

1986
~~1~~
~~9~~
~~8~~
~~6~~
 0..9

Dynamic Array
 LIST

1986

n = 1986

10) 1986 (198
 1980

10) 198 (19
 1070
 8

10) 19 (1
 10
 9

10) 1 (0
 0
 1

6

l = []

0 1
 1 9
 2 8
 3 6

```

def build(self, n: "Python int") -> "list of int":
    if n < 0:
        n = -n
    l = []
    if n < 10:
        l.append(n)
    else:
        while n != 0:
            l.append(n % 10)
            n = n // 10
        self._reverse(l)
    return l

```

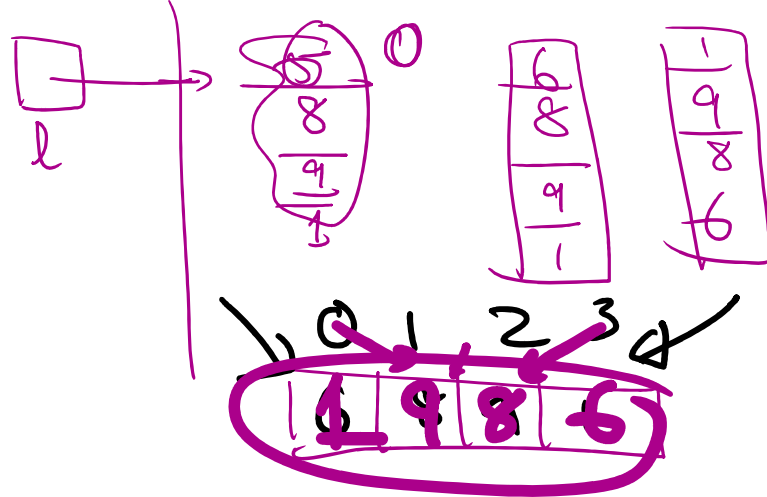
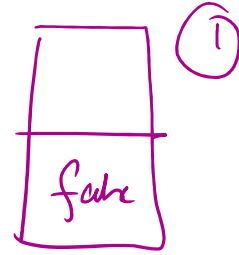
1986

198

10) 1986

6

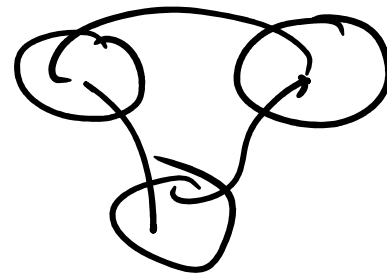
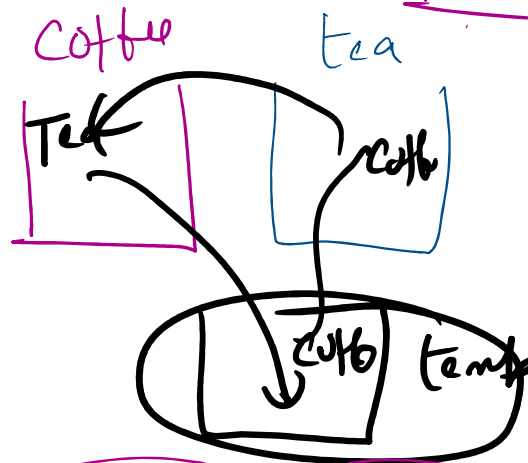
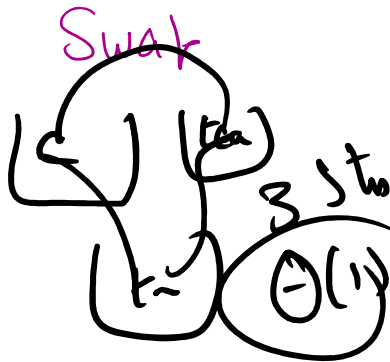
0..9



```

#####
def reverse(self, l: "list of int") -> "None":
    i = 0
    j = len(l) - 1
    while i < j:
        t = l[i]
        l[i] = l[j]
        l[j] = t
        i = i + 1
        j = j - 1

```



Magic function

Wednesday, September 6, 2023 6:47 PM

① --init--

② --str--

③ --getitem--

④ --len--

⑤ --setitem--

⑥ --add--
--sub--
--mul--

$a = \text{Int}(200)$
 $\text{print}(a)$ 25395680

$a[0]$ $a[1]$ $a[2]$
 $a[3]$

$a = \text{Int}(1986)$
 $\text{len}(a)$ 4 digit

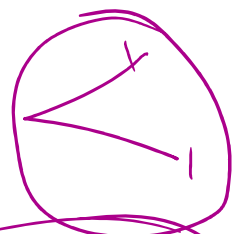
$a[3] = 420$

$a = \text{Int}(20)$
 $b = \text{Int}(40)$
 $s = a + b$
 $s = 60$

-- 11 --

1

4 roots



~~$a < b$~~
 $a > b$
 $a \leq b$
 $a \geq b$

$a < b$

$a < b$

$a > b = b < a$

$a \leq b = \neg(b < a)$
 $a \geq b = \neg(a < b)$

~~$20 == 20$~~
 $(a == b)$

$= \neg(a < b) \text{ OR } (b < a)$
 $20 \leq 20$ (False)
 $20 \leq 20$ (True)
 Equa

(DS)

(ml)

1
9
8
6

(1986)

false

PYTHON LIST - 1986

(int)

= [1, 9, 8, 6]

0	1	2	3
1	9	8	6

(a)

S = "1986"

ml = xx(PYTHON LIST)

198

S ~~198~~

S = 0

S = 0 + 1 = 1

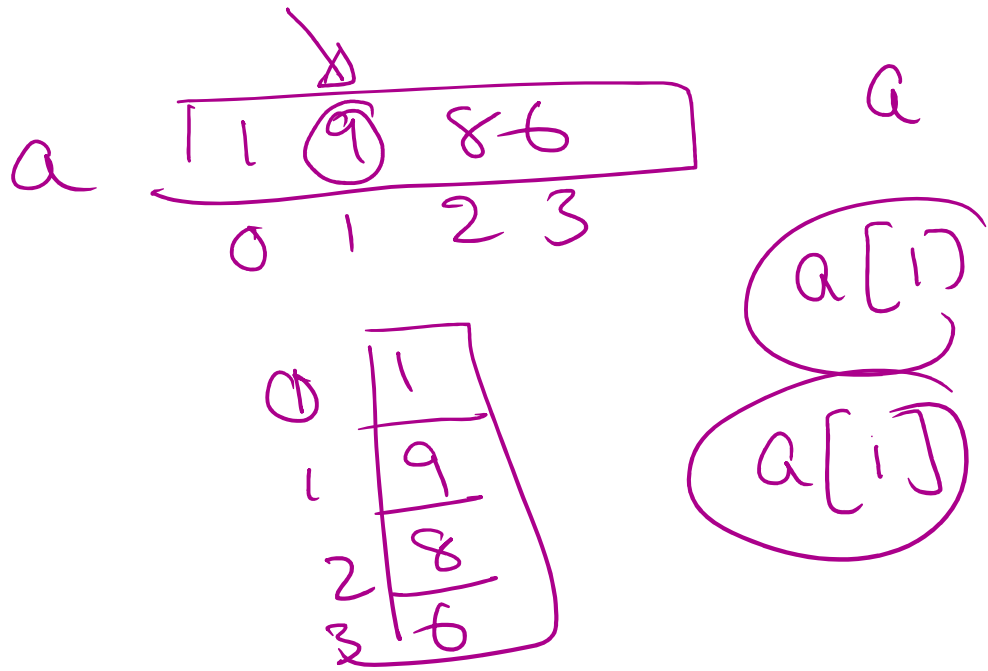
S = 0

S = S * 10 + a[i]

1980 + 6 = 1986

10 + 9 = 19

190 + 8

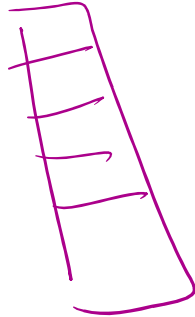


```

testing add_structure
0 +[0]
100 +[1, 0, 0]
-100 -[1, 0, 0]
-10098765456786542342222 -[1, 0, 0, 9, 8, 7, 6, 5, 4, 5, 6, 7, 8, 6, 5, 4, 2, 3, 4, 2, 2, 2, 2]

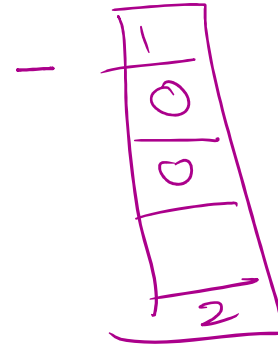
```

Sign



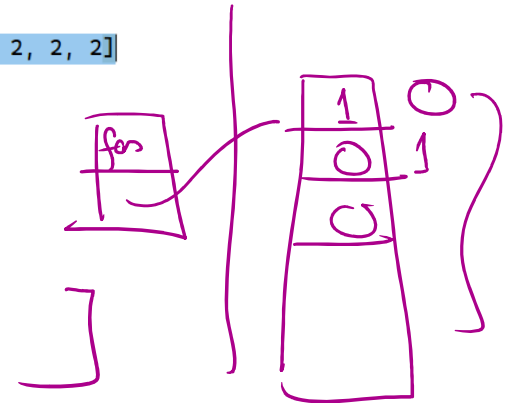
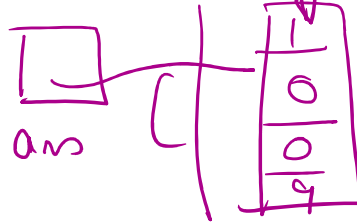
$$+ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

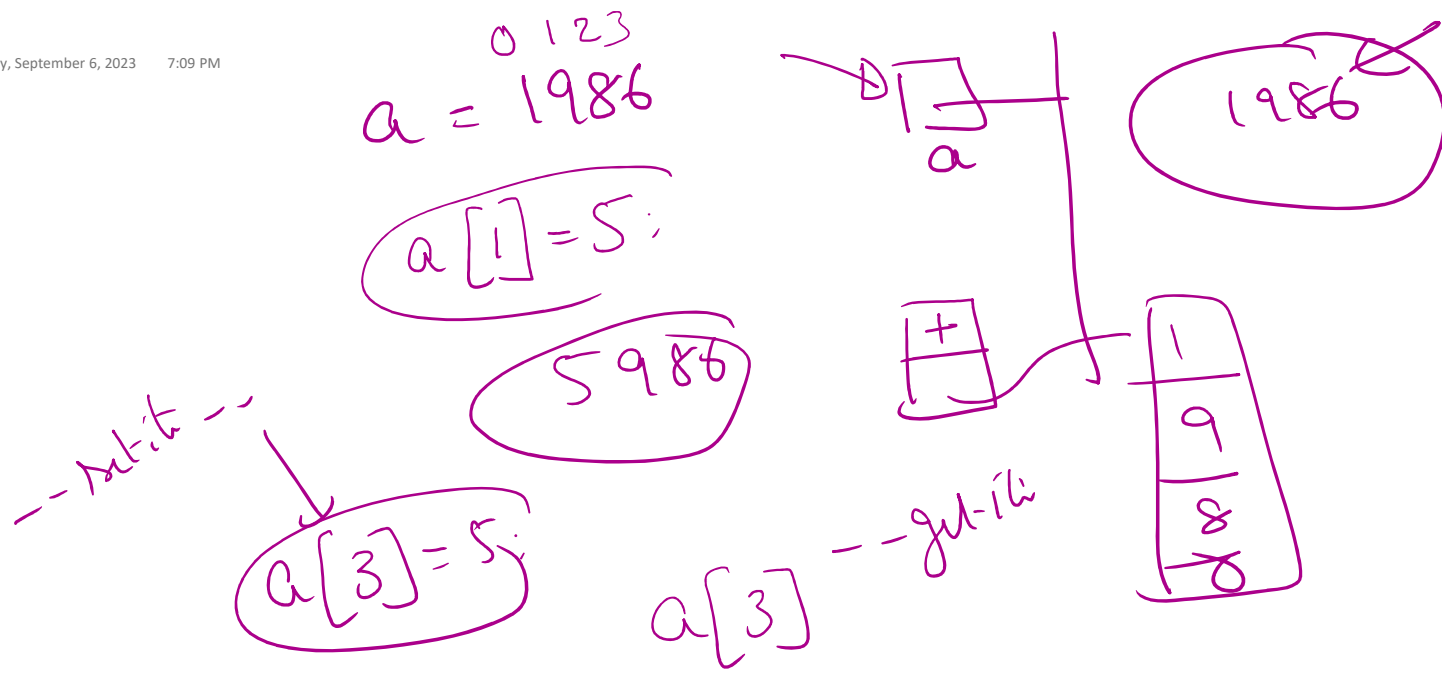
$$-100 = -[1, 0, 0] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$




```
def _access(self):  
    print("-----Testing access-----")  
    n = -10098765456786542342222  
    a = Int(n)  
    print(a)  
    ans = [1, 0, 0, 9, 8, 7, 6, 5, 4, 5, 6, 7, 8, 6, 5, 4, 2, 3, 4, 2, 2, 2, 2]  
    d = len(a)  
    d1 = len(ans)  
    if d != d1:  
        print("Number of digits is ", d1, " Your answer is", d)  
    assert d == d1  
    for i in range(d):
```

--len--





1.1.1 Naming convention

1 **Class Name: Must start with Capital**
EX: class Bank();

2 **Functions and Variable Names:**
Function names should be lowercase, with words separated by underscores as necessary to improve readability
Example: num_work_done

3 **single leading underscore**
EXAMPLE: work_done
This convention is used for declaring private variables, functions, methods and classes in a module. Anything with this convention are ignored in from module import *.
However, of course, Python does not supports truly private, so we can not force somethings private ones and also can call it directly from other modules.
So sometimes we say it "weak internal use indicator".

4 **Double Trailing Underscore __init__:**
Indicates special methods defined by Python language.

Graph()

build_getp

__set__
class w.l.

__magic__

matrix

+ - *

0 = 0 + 0

25

Wednesday, September 6, 2023

7:43 PM

Solution

too