print a —— # {}

b = dict()

print(type(b)) —— # {}

print(len(b)) —— # 0

print(b) —— # {}

RAMA —— Greater
RAJESH —— Smaller

M is after J.

21/7/25

a = {10 : 'Ramesh', 20 : 'Kiran', 15 : 'Amar', 18 : 'Sita'}

print('Keys of dictionary') —— # always 10 20 15 18

print dictionary using print() —— print(a)

Find outputs:

a = {
    print('Hyd'),
    print('Sec'),
    print('Cyb')
}

print(type(a)) —— # class <'set'>

print(a) —— # None//

print(len(a)) —— # 1.

a = 25.7
print(id(a))
print(a) —— # 25.7

a = 7
print(a) - # 7

Unicode values in string comparison

1 < 2 > 3 > 1 - False

10 < 20 < 30 — T

10 >= 20 < 30 → False
F

10 < 20 > 30 — False

1 < 2 < 3 < 4 = True

4 > 3 >= 3 > 2 — True.

Anonymous object:

_ = 25

print(_) —— # Anonymous object 25

print(type(_)) —— # class 'int'>

a, _, c = 10, 20, 30

print(a) —— # 10

print(_) —— # 20

print(c) —— 30

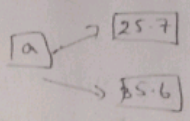
```
for _ in range(5):
    print(_, 'Hello')   ---- # 0 : Hello
                             1 : Hello
                             2 : Hello
                             3 : Hello
                             4 : Hello.
```

Find outputs:

```
a = 25
print(id(a))  — # Address of reference of obj 'a'.
a = 35
print(id(a))  —# Addres of reference of obj 'a' [new object is created].
```

Find outputs:

```
a = 25.7
print(id(a)) — # Address of object 'a'.
print(a)     — # 25.7
a = 35.6
print(id(a)) — # Address of object 'a'
print(a)     — # 35.6
```




```
b = True
print(id(b)) - # Address of object 'b'.
b = false.
print(id(b)) — # Address of object 'b',
c = None
print(id(c)) — # Address of object 'c'.
c = None
print(id(c)) — # Address of object 'c'.
```

Find outputs

```
a = 'Hyd'
print(id(a))  —— # Addres of object 'a'
a[1] = 'e'  —— # Error assignment is not possible
a = 'Sec'
print (id(a))  —— # New address of obj 'a' printed
b = (10, 20, 15, 18)
print (id(b))  —— # Addres of tuple object.
b[2] = 19  —— # Assigning not possible in tuples as they are
               immutable.
b = (30, 40, 35, 32)
print (id(b))  —— # New address of tuple object generated.
c[3] = 10
c = range (5)  — # range does not supp r assignment.
c = (range(5)  — # 0, 1, 2, 3, 4
print (id(c))  —— # Addres of range object c
```

Find outputs: ** operator

```
print (3 ** 4)  —— 3^4 # = 81
print (10 ** -2)  —— 10^(-2) = 1/100 = 0.01
print (4 ** 3 ** 2)  —— 4^3^2 = 4^9 = 24
                        (Right to Left)
print (3+4*5 - 32/8 ** 3)  —— 2^3 = 8
                              32/8 = 4.0
                              4*5 = 20
                              3+20-40 = -17.0
```

/ op / operator program :

VIT Vellore

print(9.0/2) —— # 4.5 (quotient of the following / operator)

print(9/2.0) —— # 4.5

print (9.0/2.0) —— # 4.5

print (9/2) —— # 4.5

print (10.5/2) —— # 5.25

print (10/3) —— # 3.333

print (10/2) —— # 5.0

2) 26 (13) — Quot //
Remainder %

// operator :

print (9//2) # prev integer of (4.5) - 4.

print (9.0//2) # prev integer of (4.5) = 4.0 (Atleast one float output is float.

print (9//2.0) # 4.0

print (9.0// 2.0) # 4.0 as prev integer

print (10.5//2) # 5.25's prev integer - 3.0.

print (10//3) # 3.33 prev integer is 3.0 but has no float so only # 3.

print (10.0 //3) # prev Int of 3.33 - 3.0.

print (8.5//3) # 2.66 prev. Int is 2.0.

print (18//4) # 4.5 prev int 4 no float

print (-18 //4) # -5

% operator:

print (9 % 5) — # 4

print (9.0 % 5) — # 4.0

print (9 % 5.0) — # 4.0

print (105 % 2) — # 0.5

print (8.9 % 3) — # 2.9

Find outps:

print (7/0) — Error occured

print (7//0) — Error occured

print (7 % 0) — Error occured.

22/7/25

Assignment operators

a = 25 — # Assigns ref to obj a 25

print (a) — # 25

b = a — # assign ref b to same list where a

print (b) — # 25

print (a is b) — # True both refer same obj

x = 4 — # Assigns ref to obj 4

y = 5 — # Assigns ref to obj 5

z = x + y * 6 — # Assigns ref result of x + y * 6

print (z) — # 34

25 = a — # Error