

[P212.py]

```
name = 'Raj'; roll = 101; per = 70.77
```

```
print ('%s %.d %.f' % (name, roll, per))
```

```
print ('%s %.d %.2f' % (name, roll, per))
```

o/p

Raj 101 70.770000

Raj 101 70.77

[P213.py]

```
name = 'Raj'; roll = 101; per = 70.77
```

```
print ('{} {} {}'.format(name, roll, per))
```

```
print ('{} {} {}'.format(name, roll, per))
```

```
print ('{} {} {}'.format(name, roll, per))
```

```
print ('{} {} {}'.format(n=name, r=roll, p=per))
```

```
print ('{} {} {}'.format(n=name, r=roll, p=per))
```

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

o/p

Raj 101 70.77
 Raj 101 70.77
 70.77 Raj 101
 Raj 101 70.77
 70.77 Raj 101

p214.py

```
def greet():
    print('Hello')
# end: greet
greet()
```

o/p

Hello.

User Defined Functions:-

syntax :

```
def function_name():
```

st1

st2

:

stN

```
# end: Function_name
```

```
function_name()
```

p215.py

```
def india():
```

```
    print('New Delhi')
```

```
# end: india
```

```
def japan():
    print('Tokyo')
# end: japan
```

```
print('Capital of India:', end = '')
India()
print('Capital of Japan:', end = '')
japan()
```

%/p

Capital of India: New Delhi
 Capital of Japan: Tokyo

[pal6.py]

```
def one():
    print('*')
# end: one
def three():
    print('* ***')
# end: three
def five():
    print('* * * * *')
# end: five
def main():
    five()
    three()
    one()
# end: main
```

%/p

* * * * *

* * *

*

M T W T F S S
Page No.:
Date:
YOUVA

M T W T F S S
Page No.:
Date:
YOUVA

main()

P217.py

```
def one () :  
    print ('.. *')  
# end: one  
def three () :  
    one ()  
    print ('.* ***')  
# end: three  
def five () :  
    three ()  
    print ('*****')  
# end: five  
def main () :  
    five ()  
# end: main
```

O/P
.. *
.* ***

main()

P218.py

```
def one () :  
    print ('.. *')  
# end: one  
def three () :  
    print ('.* ***')  
    one ()  
# end: three  
def five () :
```

```

print ('* * * * *')
three ()
# end: five
def main () :
    five ()
# end: main

```

o/p
* * * * *
* * * *
* * *
* * *

main ()

P219.py

```

def square () :
    x = int (input ('enter no. : '))
    s = x ** 2
    print (f 'square : {s}')
# end: square
def main () :
    square ()
# end: main

```

x s
8 64

o/p
enter no: 8
square : 64

main ()

P220.py

'''

```

def greet () :
    print ('Hello')
# end: greet
'''
```

```

def greet () : print ('Hello')
def msg () : print ('Python is easy')
```

def main () :
 greet ()
 msg ()
end : main

main ()

P221.py

def main () :
 greet ()
 msg ()
end : main
def greet () : print ('Hello')
def msg () : print ('Python is easy')

main ()

P222.py

main () # error

def main () :
 greet ()
 msg ()
end : main
def greet () : print ('Hello')
def msg () : print ('Python is easy')

main ()

* Call By Value

Passing parameters / arguments to function (fn)

[P223.py]

```
def display(x): # accept
    print(f' value passed {x} ')
# end: display
def main():
    display(10) # pass
    display(30)
    v = 50
    display(v)
    display(v+20)
    display(90.11)
    display('Raj')
# end: main
```

display

x | x
10

x | x
30

main
v | x
50

x | x
70

main()

o/p

value passed : 10
 value passed : 30
 value passed : 50
 value passed : 70
 value passed : 90.11
 value passed : Raj

x | x
90.11

x | x
'Raj'

P224.PY

```
def square(side):
    area = side * * 2
    print(f'area : {area}')
# end: square

def main():
    s = int(input('enter side : '))
    square(s)
# end: main
```

square

x	side	area
	8	64

main

x	s
	8

main()

* Passing multiple params to fn :-

P225.PY

```
def add(a, b):
    print(f'add is {a+b}')
# end: add

def multiply(c, d):
    print(f'mult is {c*d}')
# end: multiply

def main():
    print('enter 2 no.s : ')
    x, y = int(input()), int(input())
    add(x, y)
    multiply(x, y)
# end: main
```

add

x	a	b
	10	7

multiply

x	c	d
	10	7

main

x	x	y
	10	7

main()

enter 2 no.s:

10 ↴

7 ↴

odd is 17

mult is 70

[P226.PY]

```
def greet(msg, name):
    print(f'{msg} {name}')
# end: greet
def main():
    greet('Hello', 'Raj') # Positional paras/ arguments
    greet(name='Amit', msg='Hi') # keyword
                                # paras / arguments
# end: main
```

main()

greet

O/P

Hello Raj
Hi Amit.

x | msg name
 | 'Hello' 'Raj'

x | msg name
 | 'Hi' 'Amit'

[P227.PY]

```
def greet(msg, prefix, name):
    print(f'{msg} {prefix} {name}')
```

end : greet
 def main () :
 greet ('Hello', 'Mr', 'Raj') # P
 greet (name = 'Amit', msg = 'GM', prefix = 'Dr') # KW
 greet ('Hi', name = 'Ravi', prefix = 'Sir') # P & KW
 #end: main

[main()]

greet

	msg	prefix	name	O/P
x	'Hello'	'Mr'	'Raj'	Hello Mr Raj
x	'GM'	'Dr'	'Amit'	GM Dr Amit
x	'Hi'	'Sir'	'Ravi'	Hi Sir Ravi

[P228.py]

```

def display (* names): # names : tuple
    print ('List :')
    for x in names:
        print (x)
    #end : for
#end: display
def main () :
    display ('Raj', 'Amit', 'Ravi')
    display ('Tom', 'Sam', 'Joe', 'Jim', 'Jack')
  
```

display ('Jill')
end: main

main()

O/P

List :	List :	List :
Raj	Tom	Jim
Amit	Sam	
Ravi	Joe	
	Jim	
	Jack	

- * Variable length arguments / paras.
- * : args operators.

p229.py

```
def display (** record ): # Record: dict
    print ('Record: ')
    for k,v in record.items():
        print (f'{k}: {v}')
    # end: for
# end: display
def main():
    display (company = 'Yamaha', cost = 90000)
    display (name = 'Raj', roll = 101, per = 90.77)
# end: main
```

main()

o/p

Record :

company : Yamaha
cost : 90000

Record :

name : Raj
roll : 101
per 70.77

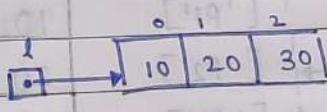
record

['company']	Yamaha
['cost']	90000

* : kw args operators

P230.py

```
def display (p1, p2, p3):  
    print (f'{p1} {p2} {p3}')  
# end : display  
def main ():  
    l = [10, 20, 30]  
    display (l[0], l[1], l[2])  
    display (*l)  
# end : main
```



main()

display

X	P1	P2	P3
	10	20	30

X	P1	P2	P3
	10	20	30

o/p
10 20 30
10 20 30

p231.py

```

def display (p1, p2, p3):
    print (p1, p2, p3)
# end: display

def main ():
    d1 = { 'p1': 10, 'p2': 20, 'p3': 30 }
    display (d1['p1'], d1['p2'], d1['p3'])
    display (**d1)

    d2 = { 'p3': 30, 'p1': 10, 'p2': 20 }
    display (**d2)
    d3 = { 'p1': 10, 'p2': 20 }
    # display (**d3) # error
    d4 = { 'p1': 10, 'p2': 20, 'p4': 30 }
    # display (**d4) # error
# end: main

```

display

main ()

main

d1	'p1'	10	'p2'	20	'p3'	30
→						

x	P1	P2	P3
	10	20	30

x	P1	P2	P3
	10	20	30

y	P1	P2	P3
	10	20	30

O/P

10 20 30

10 20 30

10 20 30

function
main()
Returning value from function
cost
fan 800
tube 43.50

| p232.py |

```
def fan ():  
    return 800  
# end: fan  
def tube ():  
    return 43.50  
# end: tube  
def main ():  
    c1 = fan ()  
    print ('Fan cost: {c1}')  
    c2 = tube ()  
    print ('Tube cost: {c2}')  
# end: main
```

main()

O/P
Fan cost: 800
Tube cost: 43.50

P233.py

```
def square(side):
    area = side * * 2
    return area

# end : square

def main():
    s = int(input('enter side: '))
    a = square(s)
    print(f'area: {a}')

# end : main
```

square

x	side	area
	8	64

main

x	s	a
	8	64

main()

enter side: 8 ↴
area: 64.

P234.py

```
def add(a, b):
    c = a + b
    return c
```

x	a	b	c
	100	70	170

add

end : add

def main():

```
print('enter 2 no.s: ')
x, y = int(input()), int(input())
z = add(x, y)
print(f'add is {z}')
# end: main
```

main

x	x	y	z
	100	70	170

main()

W	T	F	S	S
YOUVA				

M	T	W	T	F	S
Page No.:					
Date:					YOUVA

enter 2 nos.
100 ↴
70 ↴
add is 170.

[p235, p4]

```
def square (x):
    return x * * 2
# end : square
def cube (y):
    return y * * 3
# end : cube
def main ():
    91 = square (10) + cube (5)
    912 = square (cube (2))
    913 = square (cube (2) + 3)
    914 = square (cube (2) + square (3))
    print (91, 912, 913, 914)
# end : main
```

main()

O/p
225 64 121 289

max → 1 line
Single line if-else

gt of 3 nos max → no change
main → changes

[pa36.py]

```

def max(a,b):
    if a > b:
        return a
    else:
        return b
#end: if -else
# end: max
def main():
    print('enter 2 no.s :')
    x,y = int(input()), int(input())
    z = max(x,y)
    print(f'{z} is gt')
# end: main

```

[main()]

(F + enter 2 no.s :

10 ↗

5 ↘

10 is gt.

max

x	a	b
	20	
		80

main

x	y	z
20		
	80	80

enter 2 no.s :

20 ↗

80 ↘

80 is gt.

P237.PY

```

def positive(v):
    return v > 0
# end: positive

def main():
    x = int(input('enter a no.: '))
    if positive(x):
        print(f'{x} is +ve')
    else:
        print(f'{x} is -ve')
# end: main

```

positive

x | v
30

main

x | x
30

main()

positive

x | v
-10enter a no.: 30 ↴
30 is +veenter a no.: -10 ↴
-10 is -ve

main

x | x
-10

P238.PY

```

def square(v):
    return v ** 2
# end: square

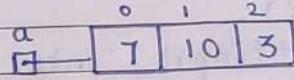
```

```

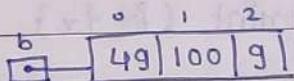
def main():
    a = [7, 10, 3]
    b = [square(x) for x in a]
    print(a, b, sep = '\n')
# end: main

```

main()



%p
[7, 10, 3]
[49, 100, 9]



p239.py

```

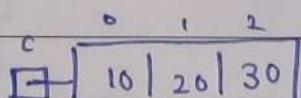
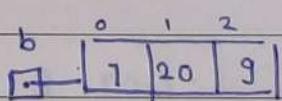
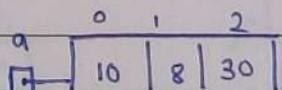
def square(v): return v * v
def main():
    a = [7, 10, 3]
    b = list(map(square, a))
    print(a, b, sep = '\n')
# end: main

```

main()

Ans:

(max)



M T W T F S
Page No.:
Date: YOUNA

M T W T F S
Page No.:
Date: YOUNA

p240.py

```
def add (v1, v2) :  
    return v1 + v2  
# end : square  
def main () :  
    a = [10, 8, 30]  
    b = [7, 20, 9]  
    c = [add (x,y) for x,y in zip(a,b)]  
    # (10,7) (8,20) (30,9)  
    print (a, b, c, sep = '\n')  
# end : main
```

main()

```
a [ 10 | 8 | 20 ]  
b [ 7 | 20 | 9 ]  
c [ 17 | 28 | 39 ]
```

p241.py

```
def add (v1, v2) : return v1 + v2  
def main ():  
    a, b = [10, 8, 30], [7, 20, 9]  
    c = list (map (add, a, b))  
    print (a, b, c, sep = '\n')  
# end : main
```

main()

[p242.py]

```

def positive(v):
    return v > 0
# end: positive

def main():
    a = [10, -5, 70, -30]
    b = [x for x in a if positive(x)]
    # b = [10, 70]
    print(a, b, sep='\n')
# end: main

```

[main()]

%/P

[10, -5, 70, -30]
[10, 70]

[p243.py]

```

def positive(v): return v > 0
def main():
    a = [10, -5, 70, -30]
    b = list(filter(positive, a))
    print(a, b, sep='\n')
# end: main

```

[main()]

P244.PY

M	T	W	T	F	S	S
Page No.:						
Date:						

```

def positive(v1): return v1 > 0
def square(v2): return v2 ** 2
def main():
    a = [10, -5, 70, -30]
    b = [square(x) for x in a if positive(x)]
    print(a, b, sep='\n')
# end: main

```

main()

o/p

[10, -5, 70, -30]
[10, 70]

P245.PY

```

def positive(v1): return v1 > 0
def square(v2): return v2 ** 2
def main():
    a = [10, -5, 70, -30]
    b = list(map(square, filter(positive, a)))
    print(a, b, sep='\n')
# end: main

```

main()

o/p

[10, -5, 70, -30]
[100, 4900]

* factorial

$$n! = 1 \times 2 \times 3 \dots \times n$$

n

4

$$1 \times 2 \times 3 \times 4 : (24)$$



p246.py

Factorial

```
def factorial(n):
```

$$f = i = 1$$

```
while i <= n:
```

$$f = f * i$$

$$i += 1$$

```
# end : while
```

```
return f
```

```
# end : factorial
```

```
def main():
```

```
x = int(input('enter a no. : '))
```

```
y = factorial(x)
```

```
print(f'factorial : {y}')
```

```
# end : main
```

main()

main

x	x	y
4		24

enter a no. : 4 ↴

factorial : 24

Complete Division [c. 10]

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

$$\begin{array}{r} 5 \\ 3 \boxed{1} 5 \\ - 1 5 \\ \hline 0 \end{array}$$

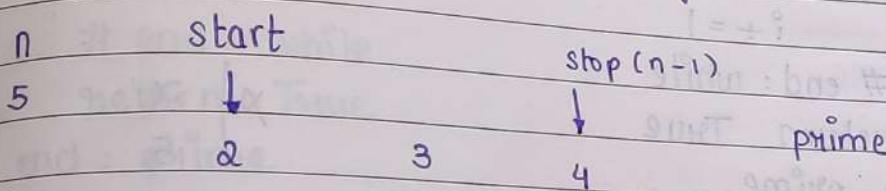
\checkmark c. 10

$$\begin{array}{r} 5 \\ 3 \boxed{1} 6 \\ - 1 5 \\ \hline 1 \end{array}$$

\times c. 10

* Prime numbers :

Number c. 10 by 1 & itself only.



$$\begin{array}{r} 2 \\ 2 \boxed{5} \\ - 4 \\ \hline 1 \end{array}$$

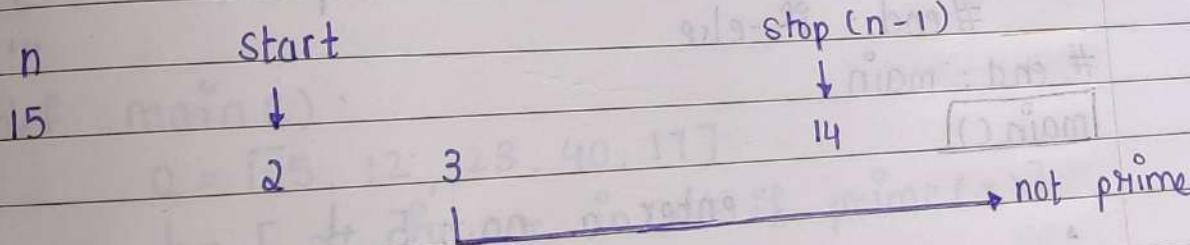
$\boxed{1} \times$ c. 10

$$\begin{array}{r} 3 \boxed{5} \\ - 3 \\ \hline 2 \end{array}$$

$\boxed{2} \times$ c. 10

$$\begin{array}{r} 4 \boxed{5} \\ - 4 \\ \hline 1 \end{array}$$

$\boxed{1} \times$ c. 10



$$\begin{array}{r} 7 \\ 2 \boxed{1} 5 \\ - 1 4 \\ \hline 1 \end{array}$$

\times c. 10

$$\begin{array}{r} 5 \\ 3 \boxed{1} 5 \\ - 1 5 \\ \hline 0 \end{array}$$

\checkmark c. 10

[prac7.py]

Prime

```

def prime(n):
    i = 2
    while i <= n - 1:
        if n % i == 0:
            return False
        # end: if
        i += 1
    # end: while
    return True
# end: prime

def main():
    x = int(input('enter a no.: '))
    if prime(x):
        print(f'{x} is prime')
    else:
        print(f'{x} is not prime')
    # end: if-else
# end: main

```

main

main()

n	i
x	5
	2
	3
	4
	5

x	x
	5

enter a no.: 5 ↴
5 is prime

Prime

main

x	n	i	x	x
	15	2		15
		3		

enter a no.: 15 ↴
15 is not prime.

P248.py

M T W T F S
Page No.: _____
Date: _____ YOUVA

```
def prime(n):
    i = 2
    while i <= n - 1:
        if n % i == 0:
            return False
        # end: if
        i += 1
    # end: while
    return True
# end: prime
```

""

%p

[5, 12, 23, 40, 17]

[5, 23, 17]

[5, 12, 23, 40, 17]

[5, 23, 17]

""

```
def main():
    a = [5, 12, 23, 40, 17]
```

```
b = [x for x in a if prime(x)]
```

```
print(a, b, sep='\n')
```

```
c = list(filter(prime, a))
```

```
print(a, c, sep='\n')
```

```
# end: main
```

main()

[p249.py]

```
def factorial(n):
    f = i = 1
    while i <= n:
        f = f * i
        i += 1
    # end : while
    return f
# end : factorial
def main():
    a = [5, 3, 6]
    print(a)
    b = [factorial(x) for x in a]
    print(b)
    c = list(map(factorial, a))
    print(c)
# end : main
```

[main()]

""

O/P

[5, 3, 6]

[120, 6, 720]

[120, 6, 720]

""

P250.PY

M	T	W	T	F	S	S
Page No.:						
Date:						

YOUVA

```

def factorial(n):
    f = i = 1
    while i <= n:
        f = f * i
        i += 1
    # end: while
    return f
# end: factorial

def prime(n):
    i = 2
    while i <= n - 1:
        if n % i == 0:
            return False
        # end: if
        i += 1
    # end: while
    return True
# end: prime

def main():
    a = [4, 5, 10, 3, 7]
    print(a)
    b = [factorial(x) for x in a if prime(x)]
    print(b)
    c = list(map(factorial, filter(prime, a)))
    print(c)
# end: main

```

main()

P251.py

```
# def greet(): print('Hello')
greet = lambda: print('Hello')
# def square(v): return v ** 2
square = lambda v: v ** 2
# def add(v1, v2): return v1 + v2
add = lambda v1, v2: v1 + v2
def main():
    greet()
    print(square(7))
    print(add(10, 20))
# end: main
```

main()

%
Hello
49
30

P252.py

```
# def square(v): return v ** 2
def main():
    a = [10, 5, 19]
    # b = list(map(square, a))
    b = list(lambda v: v ** 2, a)
    print(a, b, sep='\n')
# end: main
```

main()

P253-PY

```

# def add(v1, v2): return v1 + v2
def main():
    a, b = [10, 20, 30], [7, 8, 9]
    # c = list(map(add, a, b))
    c = list(map(lambda v1, v2: v1 + v2, a, b))
    print(a, b, c, sep='\n')
# end: main

```

main()

P254-PY

```

# def positive(v): return v > 0
def main():
    a = [10, -5, 30, -40]
    # b = list(filter(positive, a))
    b = list(filter(lambda v: v > 0, a))
    print(a, b, sep='\n')
# end: main

```

main()

P255-PY

```

# def square(v1): return v1 ** 2
# def positive(v2): return v2 > 0
def main():
    a = [10, -5, 30, -40]
    # b = list(map(square, filter(positive, a)))

```

```

b = list(map(lambda v1: v1**2, filter(lambda
                                         v2: v2 > 0, a)))
print(a, b, sep='\n')
# end: main

```

main()

* Call By Value:-

Any change made in user defined fn (modify) is NOT sent back to calling fn (main)

int float str bool

p256.py

```

def modify(a, b, c):
    a = 20
    b = 80.88
    c = 'Amit'
# end: modify
def main():
    x, y, z = 10, 70.77, 'Raj'
    print('before call: {} {} {}'.format(x, y, z))
    modify(x, y, z)
    print('after call: {} {} {}'.format(x, y, z))
# end: main

```

modify		
a	b	c
10	70.77	'Raj'
20	80.88	'Amit'

main		
x	y	z
10	70.77	'Raj'

main()

%/p

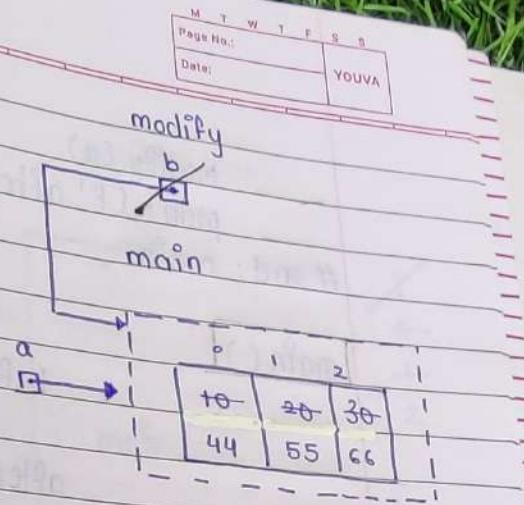
before call: 10 70.77 Raj
 after call: 10 70.77 Raj

P257.PY

```

def modify(b):
    b[0] = 44;
    b[1] = 55;
    b[2] = 66;
# end: modify
def main():
    a = [10, 20, 30]
    print('before call: ', a)
    modify(a)
    print('after call: ', a)
# end: main

```



o/p
before call: [10, 20, 30]
after call: [44, 55, 66]

* Call By Reference:-

Any change made in user defined fn (modify) is sent back to calling fn (main)
list dict set

P258.PY

```

def modify(b):
    b['name'] = 'Amit'
    b['roll'] = 201
    b['per'] = 80.88

```

end: modify

def main():

a = {'name': 'Raj', 'roll': 101, 'per': 70.77}

print('before call: ', a)

modify (a)
print ('f' after call: {a})
end: main

main ()

%/p

before call: { 'name': 'Raj', 'roll': 101, 'per':

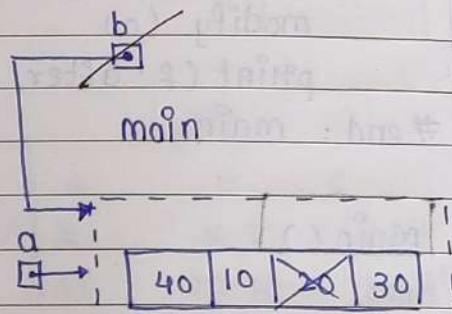
70.772

after call: { 'name': 'Amit', 'roll': 201, 'per':

80.883

P259.py

modify



def modify (b):
 b.remove(20)
 b.append(30)

end: modify
def main ():

a = {10, 20, 40}

print ('f' before call: {a})

modify (a)

print ('f' after call: {a})

end: main

main ()

%/p

before call: {40, 10, 20}

after call: {40, 10, 30}

P260.py

def square (b):

for i in range(len(b)): # 0, 1, 2

b[i] = b[i] ** 2

end: for

end: square

def main():

a = [6, 10, 8]

print(a)

square(a)

print(a)

end: main

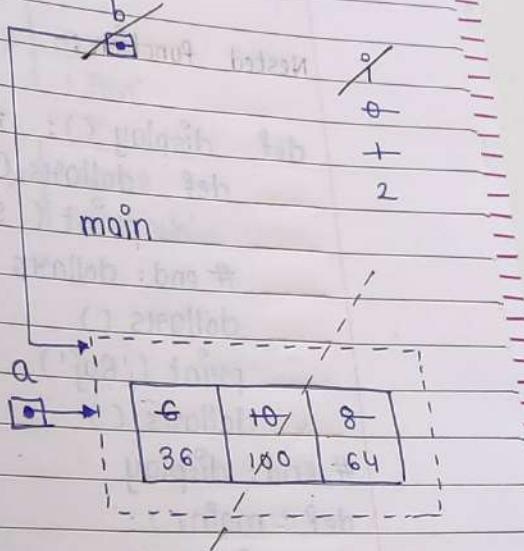
main()

o/p

[6, 10, 8]

[36, 100, 64]

square



* P#1

Requirement length changes

3 times

300+ lines

3

P# 2

def calculate():

100 lines Ø

10 times

1000+ lines

10

end: calculate

100 lines Ø

100 lines Ø

calculate()

Requirement length changes

3 times

103+ lines

1

calculate()

100 lines Ø

10 times

110+ lines

1

calculate()

(890 ↓)

(197 ↓)

P261.py

Nested function:-

```

def display():
    def dollars():
        print('$$$')
    dollars()
    print('Raj')
    dollars()

# end: display
def main():
    display()
# end: main

```

main()

%P
\$\$\$
Raj
\$\$\$

P262.py

```

def display(name):
    def dollars():
        print('$' * len(name))

    dollars()
    print(name)
    dollars()

# end: display
def main():
    display('Jim')
    display('Rakesh')

```

end : main

main()

%/p

\$\$\$

Jim

\$\$\$

\$\$\$\$\$

Rakesh

\$\$\$\$\$

display

x | name
'Jim'

x | name
'Rakesh'

p263.py

l = [10, 20, 30]

print(l)

a, b, c = l # unbox

print(a, b, c)

%/p

[10, 20, 30]

a b c

10 20 30

%/p

10 20 30

[P264.py]

$$l = [\underline{10} , \underline{20} , \underline{30} , \underline{40}]$$

$$a , b , *c = l$$

print (a, b, c)

o/p

10 20 [30, 40]

[P265.py]

$$l = [10, 20, 30, 40]$$

$$a , *b , c = l$$

print (a, b, c)

o/p

10 [20, 30] 40

[P266.py]

$$l = [10, 20, 30, 40, 50]$$

$$a , b , *c , d = l$$

print (a, b, c, d)

o/p

10 20 [30, 40]

| P267.py |

```
def cost():
    return [800, 43.75]
# end: cost
def main():
    fan, tube = cost() # [800, 43.75]
    print(f'Fan cost: {fan}\n'
          f'Tube cost: {tube}')
# end: main
```

| main()

main

x	Fan	tube
	800	43.75

o/p

Fan cost: 800

Tube cost: 43.75

* rectangle

a P



i/p l b (main)

[P268 .PY]

```

def calculate(x):
    return [x * * 2, x * * 3] # [25, 125]
# end: calculate
def main():
    v = int(input('enter a no.: '))
    square, cube = calculate(v) # [25, 125]
    print(f'square : {square}\n' f'cube : {cube}')
# end: main

```

[main ()]

Calculate

x	x
	5

main

v	square	cube
5	25	125

o/p

enter a no.: 5 ↴

square : 25

cube : 125

P269.PY

M T W T F S
Page No.:
Date:
YOUVA

```
def rectangle (l,b):  
    return [l*b, 2*(l+b)] # [80,36]  
# end : calculate  
def main ():  
    len , bre = int (input ('enter len:')), int (input ('enter  
area, peri = rectangle (len, bre) # [80,36]  
print (f' Area: {area} \n'  
      f' Perimeter: {peri}')  
#end : main
```

main()

rectangle

x	l	b
	10	8

o/p

enter len: 10 ↴

Area: 80

Perimeter: 36

main

x	len	bre
	10	8
	area	peri
	80	36

M T W T F S S
Page No.:
Date: YOUVA

P270.py

```
x = 70      # x is created (= : assignment)
print(x)
n = 'Raj'    # n is created (=)
a = [10, 20, 30] # a is created (=)
print(n, a, sep = '\n')
```

x

70

n

'Raj'

