PYTHON PROGRAMMING
GATE DA/DSA

Agenda: GATE PYQs - III

```
# GATE 2024

def f(x: int, y:int):
  for i in range(y):
    x = x+x+y
  return x
```

Which of the following statements is true about the given function?

a) If the inputs are $x=20$, $y=10$ then the return values is greater than 2^{20}.

b) Inputs: $x=20$, $y=20$ return value is greater than 2^{20}

c) inputs: $x=20$, $y=10$ return value is less than 2^{10}

d) inputs: $x=10$, $y=20$ return value is greater than 2^{20}

---

f(x, y)

for i in range(y):

x = x+x+y

return x

$i = 0$ to $y-1$

$i = y-1$

a)	20	10
b)	20	20
c)	20	10
d)	10	20

---

a) $x=20$, $y=10$

i in range(10) $\rightarrow$ 0 to 9

$i:0$ $\quad x = 2x+y \Rightarrow t_0: 2x+y$

$i=1: \quad t_1 = t_0 + t_0 + y = 2t_0 + y = t_i$

$i=2: \quad t_2 : t_1 + t_1 + y = 2t_1 + y = t_2$

$i:0 \qquad t_0 : \quad 2x+y \quad (2^1 x + (2^1 - 1)y)$

$i:1 \qquad t_1 : \quad 2t_0 + y = 2(2x+y) + y = 4x+3y$

$t_1 : \quad 4x+3y \quad (2^2 x + (2^2 - 1)y)$

$i:2 \qquad t_2 : \quad 2t_1 + y = 2(4x+3y) + y$

$= 8x + 7y$

$\vdots$

$2^3(x) + (2^3 - 1)y$

$i:0 \qquad 2^1 x + (2^1 - 1)y$

$i:1 \qquad 2^2 x + (2^2 - 1)y \ldots$

$$\boxed{i = i : \quad 2^{i+1} x + (2^{i+1} - 1)y}$$

$i=4: \qquad 2^5 x + (2^5 - 1)y$

$32x + 31y$

$i = y - 1 \qquad t_{y-1} : 2^{(y-1)+1} x + (2^{(y-1)+1} - 1)y$

return value.

$$\boxed{t_{y-1} : \quad 2^y x + (2^y - 1)y}$$

a) $x = 20, y = 10$: $\quad t_{y-1} = 2^{10}(20) + (2^{10} - 1)10$

incorrect.

$= 2^{10}(20) + 2^{10}(10) - 10$

$= 30(2^{10}) - 10 \not> 2^{20}$

b) $x = 20, y = 20$

$$t_{y-1} = 2^{20}(20) + (2^{20} - 1)20$$

Correct.

$$= 2^{20}(20) + 2^{20}(20) - 20$$

$$= 40(2^{20}) - 20 > 2^{20} \checkmark$$

c) $x = 20 ; y = 10$.

incorrect

$$t_{y-1} = 30(2^{10}) - 10 \not> 2^{10}$$

d) $x = 10, y = 20$

$$t_{y-1} = 2^{20}(10) + (2^{20} - 1)20$$

Correct

$$= 30(2^{20}) - 20 > 2^{20} \checkmark$$

Option → b & d)

```
# GATE 2018

def fun(n: int):
  i = n
  j = 0
  sum = 0

  while i>1:
    j+=1
    i//=2
  while j>1:
    j//=2
    sum+=1

  return sum
```

$j = 40$

The value returned when we call fun with the input 2^{40} is ______

a) 4   b) 5 ✓

c) 6   d) 40

$\text{fun}(2^{40})$: $n = 2^{40}$

$i = 2^{40}$; $j = 0$; $sum = 0$

$2^{40} > 1$: $j += 1 \Rightarrow 2^{39}$

$j = 1$

$2^{39} > 1$; $j = 2 \Rightarrow 2^{38}$

$2^{1} > 1$; $j = 40 \Rightarrow 1$

$$\boxed{j = 40}$$

sum = 0

$40 > 1$	sum = 1	$j = 20$
$20 > 1$	sum = 2	$j = 10$
$10 > 1$	sum = 3	$j = 5$
$5 > 1$	sum = 4	$j = 2$
$2 > 1$	sum = 5	$j = 1$

× $1 > 1$

Sum = 5

```
while (r>=y):
  r = r-y
  q = q+1
```

GATE 2017

The given code is meant to divide x by y by using repeated subtraction. The variables x, y, q and r are integers.

Which of the following conditions on the variables x, y, q and r before the execution of the above code will ensure that the loop terminates in a state satisfying the condition $\boxed{x == (y * q + r)}$?

a) (q == r) and (r == 0)

b) (x > 0) and (r == x) and (y > 0)

c) (q == 0) and (r == x) and (y > 0)

d) (q == 0) and (y > 0)

x: 45 $\qquad$ y = 7 $\qquad$ x / y

45 − 7 $\qquad$ x == y * q + r

38, 31, 24, 17, 10, 3 < 7

q=1 q=2 q=3, q=4, q=5, q=6

45 / 7 → quotient : 6

7 * 6 = 42, remainder = 3.

45 − 7, 38 − 7, 31 − 7 . . . .

x − y; x1 − y, x2 − y, . . x_k < y

31 $\qquad$ 31 $\qquad$ remainder

r == x, find remainder.

While r >= y:
    q = q+1
    r = r−y

r >= y

45 >= 7

45 − 7 = 38 $\qquad$ r == x

q = 1 $\qquad$ q = 0

y > 0

31 $\qquad$ q = 2 $\qquad$ q = 6

r = 3

10 >= 7

3 >= 7

x

45 == y*q + r

7*6 + 3

45 == 45

GATE 2003

```python
def findFunc(x, y):
  p = 1
  s = 1
  for i in range(1,y):
    p*=(x/i)
    s+=p
  return s
```

Consider the following code.
For large values of y, the return value of the function (findFunc) best approximates ____

a) x^y b) e^x c) $\ln(1+x)$

d) x^x

$p = 1$; $s = 1$ large values of y.

i : 1 to $y-1$

i	p	s
1	$1 \times \frac{x}{1} = x$	$1 + x$ s+=p
2	$x * \frac{x}{2} = \frac{x^2}{2}$	$(1+x) + \frac{x^2}{2}$ s+=p
3	$\frac{x^2}{2} \times \frac{x}{3} = \frac{x^3}{6}$	$1+x+\frac{x^2}{2}+x^3/6$

$$e^x = \boxed{1 + x + \frac{x^2}{2} + \frac{x^3}{6} + x^4/24 + \cdots}$$

$x \times x \times x \times x +$

GATE 2021

```
def inc():
  global i
  i += 1
  return i

count = 0
i = 0
for j in range(-3, 4):
  if ((j>=0) and inc()):
    count += j
count += i
print(count)
```

Which of the following options is correct?

a) Program will throw error.

b) Output : 13

c) Output : 10

d) output : 8

e) None of These.

Count = 0 ; i = 0

j → -3 to 3

if j >= 0

j = -3	if j >= 0 X	i = 0	count = 0
j = -2	j >= 0 X	i = 0	count = 0
j = -1	j >= 0 X	i = 0	count = 0
j = 0	j >= 0 (✓) inc() ✓	i = 1	count += j count = 0
j = 1	j >= 0 ✓ inc() ✓	i = 2	count += j count = 1
j = 2	j >= 0 ✓ inc() ✓	i = 3	count = 1+2 = 3

$j=3$	$j>=0$ inc()	$i=4$	Count=3+3 Count=6

count = 6

$i=4$

count += i

$6+4=10$ ✓

GATE 2016

```
def exp(X: int, Y: int):
  res = 1
  a = X
  b = Y
  while b!=0:
    if b%2 ==0:
      a = a*a
      b = b//2
    else:
      res = res*a
      b = b-1
  return res
```

The following function (exp) computes X^Y for positive integers X and Y. Which of the following condition is TRUE before every iteration of the loop?

a) $X^Y = a^b$

b) $(res*a)^Y = (res*X)^b$

c) $X^Y = res*a^b$

d) $X^Y = (res*a)^b$

res = 6
a = 6
b = 2

res = 6
a = 36
b = 1

$6^3 = 6 \times 36$

$216 = 216$

res = 1, a = X, b = Y

$X=6 ; Y=3 \Rightarrow 216$

Before first Iteration

res = 1 ; a = 6 ; b = 3

a) $6^3 \mid 6^3$ ✓

b) $(1*6)^3 = (1*6)^3$ ✓

c) $6^3 = 1*6^3$ ✓

d) $6^3 = (1*6)^3$ ✓

Enter loop: Iteration 1.

$b = 3$; $b \,!= 0$

$b \,\%\, 2 == 0 \rightarrow 1$

$res = res * a = 1 \times 6$

$res = 6$
$b = 2$

$res = 6$
$a = 6$
$b = 2$

result of iteration 1

Before Iteration 2:

a) $6^3 = 6^2$

b) $(6 \times 6)^3 = (6 \times 6)^2$

$36^3 = 36^2$

c) $6^3 = 6 \times 6^2$

$6^3 = 6^3$

d) $6^3 = (6 \times 6)^2$

$6^3 = (36)^2$

$res =$; $a =$ $b =$

e = "butter"

def f(a):

print(print(a) or e)

f("bitter")

None

None or e

butter.

bitter

butter

Output:

a) Error

b) Butter

c) Bitter

d) Butter
Bitter

e) None of these.

```
# ISRO 2020

name = "satellites"

l = len(name)
s = len(set(name))

val = ord(str(l)[0]) + ord(str(s))

def add(a,b):
  return a+b

def sub(a,b):
  return a-b

list_1 = [[sub, 0], [sub, 6], [add, 13], [sub, 2]]

for i in range(len(list_1)):
  print(chr(list_1[i][0](val, list_1[i][1])), sep="-")
```

name = "satellites"

$\boxed{l = 10}$

s = len(set(name))

s = {s, a, t, e, l, i}

s = len(set(name))

$\boxed{s = 6 \ ; l = 10}$

val = 103

val = ord(

str(l)[0] = '10'

ord('1') + ord('6')

str(l) [0] = '1'

ord('6')

[sub, 0]

0 1

0

$\boxed{val = 103}$

49 54

i : 0 to 3

i : 0

list_1[i] [0]   list [i] [1]

list [0] [0]   lis [0] [1]

sub (103, 0)

sub(103, 0) ⟶ 103 − 0 = 103

ord

chr(103) ⟵ g

i = 1

list_1 [i] [0]   list [i] [1]

1 [0]   1 [1]

sub(103, 6) ⟶ 97   103 − 6

'a'

chr(97) ⟶ a   g | a

i = 2 :   t

i = 3 :   e | gate

print('g', 'a', 't', 'e')

sep = "_"

end = '_'

g
a
t
e

g - a - t - e

g - a - t - e -

lit_1 [i] [0]   li [0] [0]

i = 0 0 0

[[sub, 0]]

li [0] [1]

sub (val, lit[i][1])

sub (val, 0)

sub (val, 0) ← 103 − 0

sub (103, 0)   103

chr(103) — g.

GATE 2014.

```
def myX(E: list, size: int):
  Y = 0
  for i in range(size):
    Y = Y+E[i]

  for i in range(size):
    for j in range(i, size):
      Z = 0
      for k in range(i, j+1):
        Z = Z+E[k]
      if Z>Y:
        Y = Z
  return Y
```

The value returned by function myX is __

a) maximum possible sum of elements in any sublist of E.

b) maximum element in any sublist of E.

c) sum of the maximum elements in all possible sublists of E.

d) sum of all elements in E.

[-1, 2, 3]

[1, 2, 3, 4]

[1], [2], [3], [4]

[1, 2], [1, 2, 3], [2, 3], [2, 3, 4]

[3, 4]

E → [-1, 2, 3] ; size = 3.

$y=0$ | $y + E[i]$ 0 to 2

$y = 0 - 1 = -1$ $i=0$

$y = -1 + 2 + 3$ $i=1$

$y = \boxed{y=4}$ $\boxed{y=4}$

for i in range(size) : 0 to 2

$i=0$ for j in range(i, size):

z = 0

for k in range(i, j+1):

z = z + E[k]

if z > y:

y = z

$i=0$ | $j=0$ to ~~3~~ 2 | z = 0

0 to 1

$j=0$ range(0, 1)

$j=1$ ↓

$K=0$

$z = z + E[0]$

$z = -1$

$z > y$:

$i = 0$

$j = 1$

2

$k : i$ to $j+1$

0 to 2

$k : 0, 1$

$z = 0$

$z = z + E[k]$

$z = E[0] + E[1]$

$z = 1$

$z > y =$

[-1] , [-1, 2] , [-1, 2, 3]

if $z > y$:

$y = z$

$y = 4$

$i = 1$

[2], [2, 3], [3]

$z = 5$

$z > y$:

$y = z$

$y = 5$

$i = 2$,

returned 5.

i = 0	j = 0 to size	k = i to j+1 [-1] [-1, 2] [-1, 2, 3]
1	j : 1 to size	k : i to j+1 1 to j+1 [2], [2, 3]
2	j :	[3]

-1 [-1] 2 [-1, 2] 3 , [-1, 2, 3]

[2] [2, 3] 3

2 [3]

3

-1 + 2 + 3 + 2 + 3 + 3

tr

option (a)

Python → print

Operators, Operator precedence

Data types - list, tuple, set, dictionary, frozen set

Functions, recursive functions

Modules; types of errors