

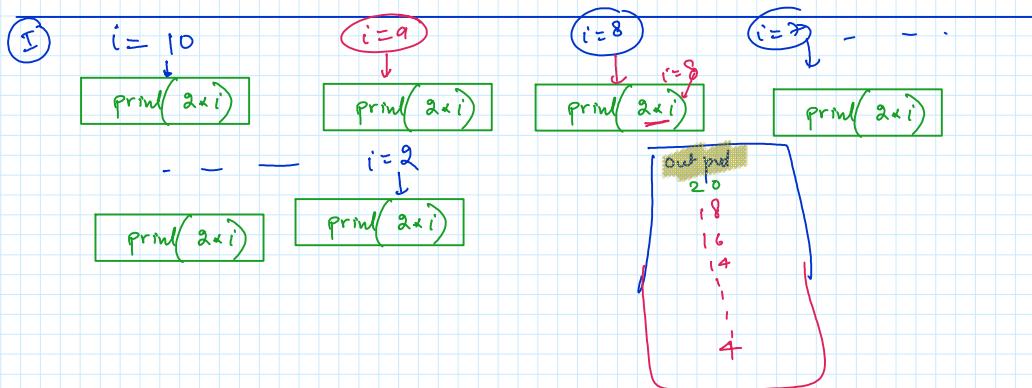
Quick Revision of Loops

For loop

A for loop lets you repeat a block of code for a set number of times.

for i in range(start, stop, step):
 []

for i in range(10, 1, -1):
 print(2*i)



While loop
A while loop runs as long as a condition is True. It's like saying, "Keep doing this until something changes."

Ex

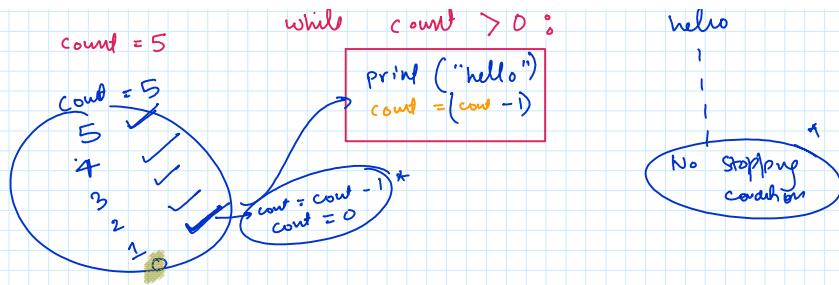
count = 5
 count = 5

while count > 0:
 print("hello")

hello
hello
hello
hello
hello

!

Ex



Question: Write a code to print the following numbers -

`print()`
`print(?)`
`print(?)`
`print(10)`
`print(?)`
`print(?)`
~~if kgd~~

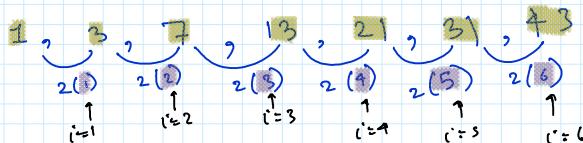
1, 4, 7, 10, 13, 16

$i=1$ increment by 3, till $i=16$, or 17

```
for i in range(1, 17, 3):
    if i == 16:
        print(i, end="")
    else:
        print(i, end=",")
```

Ques

Question: Write a loop to print the following numbers - 1, 3, 7, 13, 21, 31, 43



`num = 1`

`i = 1`

Plain English
 while num < 43
`print(num)`
 increment num by 2
`i = i + 1`

`num = 1`
`i = 1`

while (`num <= 43`):

```
print(num)
num = num + 2(i)
i = i + 1
```

Jump Statements

① Pass

Does nothing.
 It's a placeholder for future code.

After two days

1, 3, 4
 fail fail (even no)

casino (L.N.)

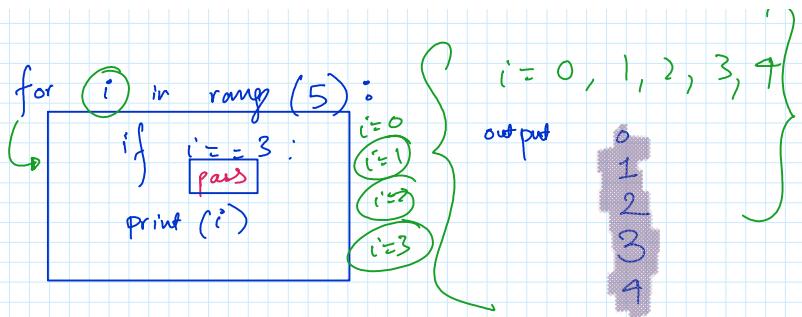
(odd → fail)
 even → Reward
 (Code)

```
if i % 2 != 0:
    print("fail")
else:
    pass
    print("even no.")
```

num2

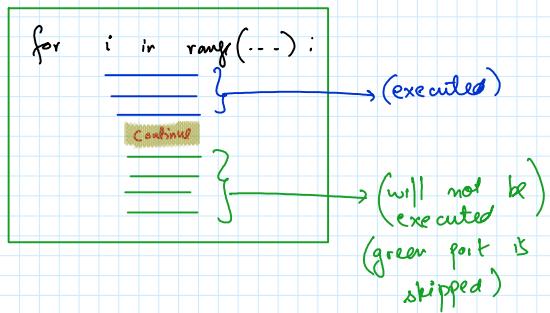
for (i) in range(15): { i = 0, 1, 2, 3, 4 }

Ans 2



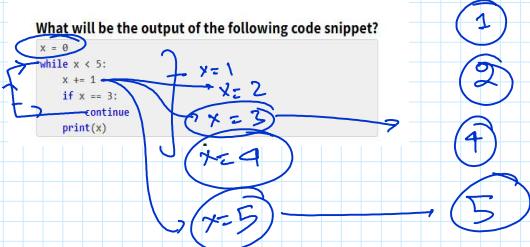
for *i* in range(...):
 _____ → represents empty placeholder work in python.
 |
pass) the loop continues without any change.
 |
 |
 |

continue
 If skip the iteration



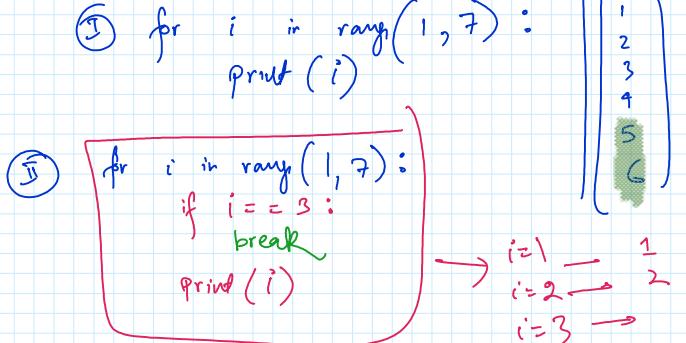
Break

→ 10: 17 pm



Break
 ↓

Exits the loop entirely.



What will be the output of the following code snippet?

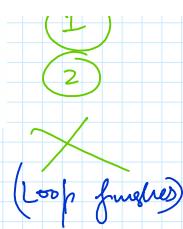
```
x = 0
while x < 5:
    x += 1
    if x == 3:
        break
    print(x)
```

1
2

```

while x < 5:
    x += 1
    if x == 3:
        break
    print(x)
print("Loop finished")

```



What will be the output of the following code?

```

count = 0
while True:
    if count == 5:
        break
    print(count)
    count += 1

```

Which statement is used to skip the current iteration of a loop and move on to the next iteration?

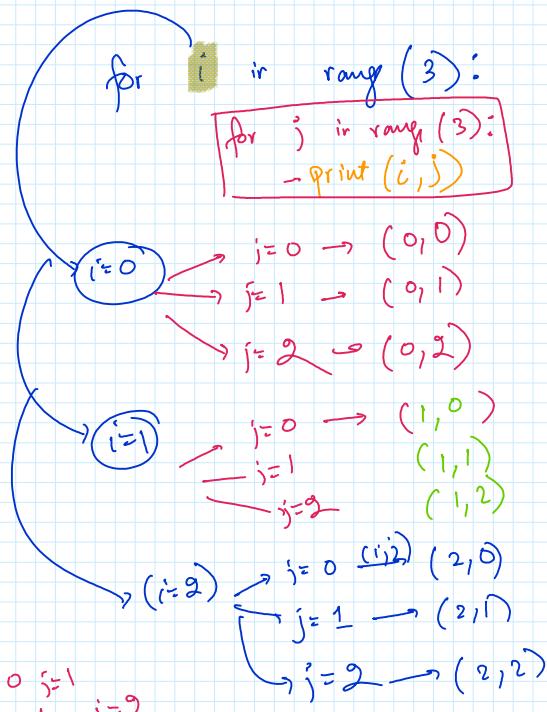
4 options

Active Duration (last opened: 30 seconds)

Appears for 10 Secs

A: break
B: continue
C: pass
D: None of these

Nested loops
(Loop inside a loop)



$N = 3$
 (3×3)

$i=0 * * *$
 $i=1 * * *$
 $i=2 * * *$

$N = 2$
 (2×2)

* *

$N = 4$
 (4×4)

$i=0$	*	*	*
$i=1$	*	*	*
$i=2$	*	*	*
$i=3$	*	*	*

$N = \text{int}(\text{input}())$

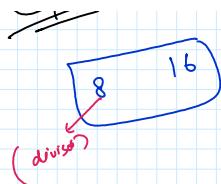
```

for i in range(N):
    for j in range(N):
        print("*")
    
```

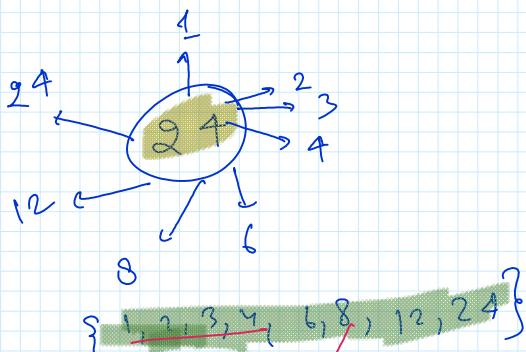
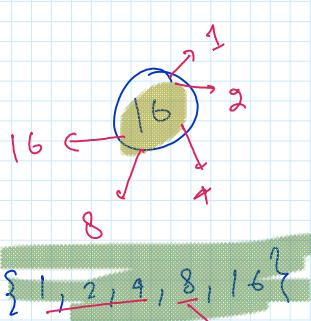
GCD (greatest common divisor)

18 16

$\text{GCD}(a, b) = \text{greatest common}$



$\text{GCD}(a, b) = \text{greatest common divisor}$



max common divisor
 (GCD) $\frac{8}{(16, 24)}$

$$\boxed{\text{GCD} \leq \min(A, B)}$$

$$\left(\frac{*}{2}, 100\right)$$

$\text{GCD} \leq 2$

$\rightarrow 3, 9 \dots$
 $\star (16, 24)$
 100% GCD will be ≤ 16 ?
 \Rightarrow Yes

$$8, 9, 10, 11, 12, 13, 14, 15, 16$$

\star $\frac{A}{16}, \frac{B}{16}$

```
for i in range(16, 0, -1):
    if A % i == 0 and B % i == 0:
        print(i)
        break
```

$$A, B$$

$$X = \underline{\min(A, B)}$$

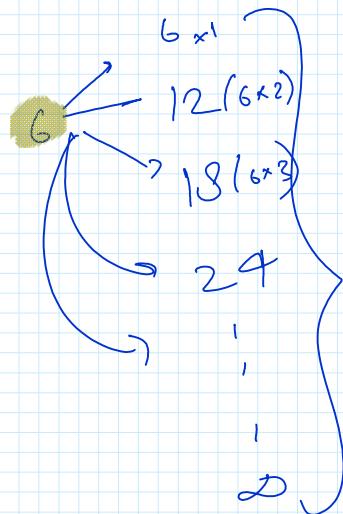
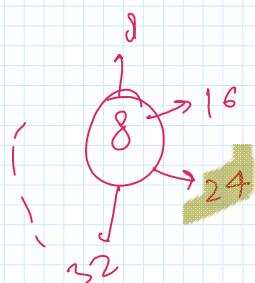
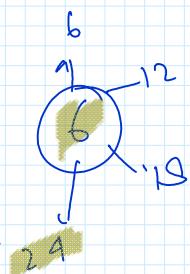
$$A, B$$

$$X = \min(A, B)$$

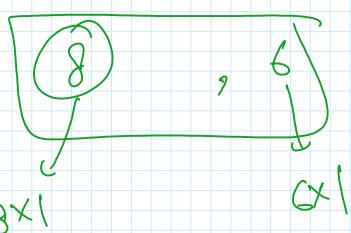
```

for i in range(X, 0, -1):
    if A % i == 0 and B % i == 0:
        print(i)
        break
    
```

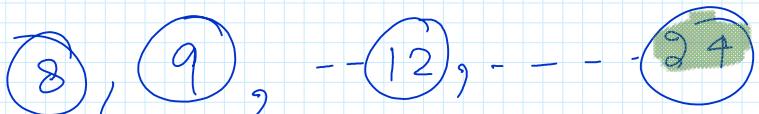
(LCM → Lowest Common Multiple)



24 is the lowest number which is common multiple of 8 and 6



$$\text{LCM}(A, B) \geq \max(A, B)$$



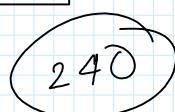
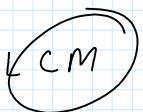
$$X = \max(A, B)$$

while True :

```

if X % A == 0 and X % B == 0:
    print(X)
    break
X = X + 1
    
```

1
2
3
4
5
6
7



can never be multiple of (6 and 7)

8 9 10 12 - - 16 - - 24

LCM
for loop

$$10 \\ 30 \\ \frac{a+b}{a_1 \times b}$$

$x = \max(a, b)$
 $y = a * b$

for i in range(x, y + 1):
 if $x \% A == 0$ and $x \% B == 0$:
 print(x)
 break

48

T = int(input("no of test cases"))

for i in range(T):

N = int(input("Enter the number"))

if N == 0:
 print(1)

else
 print(len(str(N)))

N = 123

str(N)

= "123"

len("123")

3