



Doubt Clearing Session

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Data Structure: Doubts & Stack

By: Vishvadeep Gothi



DPP

Question 2

Let S be a stack of size $n \geq 1$. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and Pop operations take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation. For $m \geq 1$, define the stack-life of m as the time elapsed from the end of $Push(m)$ to the start of the pop operation that removes m from S . The average stack-life of an element of this stack is

A. $n(X + Y)$

B. $3Y + 2X$

☒ C. $n(X + Y) - X$

D. $Y + 2X$

PUSH or POP $\Rightarrow X$ sec

b/w 2 operat^{ns} $\Rightarrow Y$ sec

$$\text{avg stack life of all elements} = \frac{\text{sum of stack life of all elements}}{\text{total no of elements}}$$

$n = 4$



stack life of element 4 $\Rightarrow y$

— 1 ————— || — 3 $\Rightarrow 3y + 2x$

— || ————— 2 $\Rightarrow 5y + 4x$

— || ————— 1 $\Rightarrow 7y + 6x$

$$\text{avg} = \frac{y + (3y + 2x) + (5y + 4x) + (7y + 6x)}{4}$$

$$= \frac{16y + 12x}{4}$$

$$= 4y + 3x \quad \leftarrow n=4$$

for n

$$\Rightarrow ny + (n-1)x$$

$$\Rightarrow ny + nx - x$$

$$\Rightarrow n(y+x) - x$$

$$Ans = 256$$

$$16^2 = 256$$

Question 3

Let Q denote a queue containing sixteen numbers and S be an empty stack. $Head(Q)$ returns the element at the head of the queue Q without removing it from Q . Similarly $Top(S)$ returns the element at the top of S without removing it from S . Consider the algorithm given below.

```
while Q is not Empty do
  if S is Empty OR Top(S) ≤ Head(Q) then
    x := Dequeue(Q);
    Push(S, x);
  else
    x := Pop(S);
    Enqueue(Q, x);
  end
end
```

$$21 \leq 22$$

True

$$21 > 22 > 23$$

> ...

The maximum possible number of iterations of the **while** loop in the algorithm is _____.

$Q \Rightarrow 16$ numbers

S $21, 22$ Empty

21 , ~~22~~ , $23, \dots, 216, 21$

for max iterations of while loop, Q must contain elements in ^v strictly decreasing order

Ex:- $n = 3$

Q: 3, 2, 1

S

1st:-

Q = 2, 1

S = 3

2nd:-

Q = 2, 1, 3

S

4th:-

Q = 1, 3, 2

S =

6th:-

Q = 2

S = 1, 3 ^{top}

8th:-

Q = 3

S = 1, 2 ^{top}

3rd:-

Q = 1, 3

S = 2

5th:-

Q = 3, 2

S = 1

7th:-

Q = 2, 3

S = 1

9th:-

Q

S = 1, 2, 3

Question 4

Consider an array of size n to implement m number of stacks (numbered 0 to $m-1$).
The empty stack has following specifications:

for $0 \leq i < m$

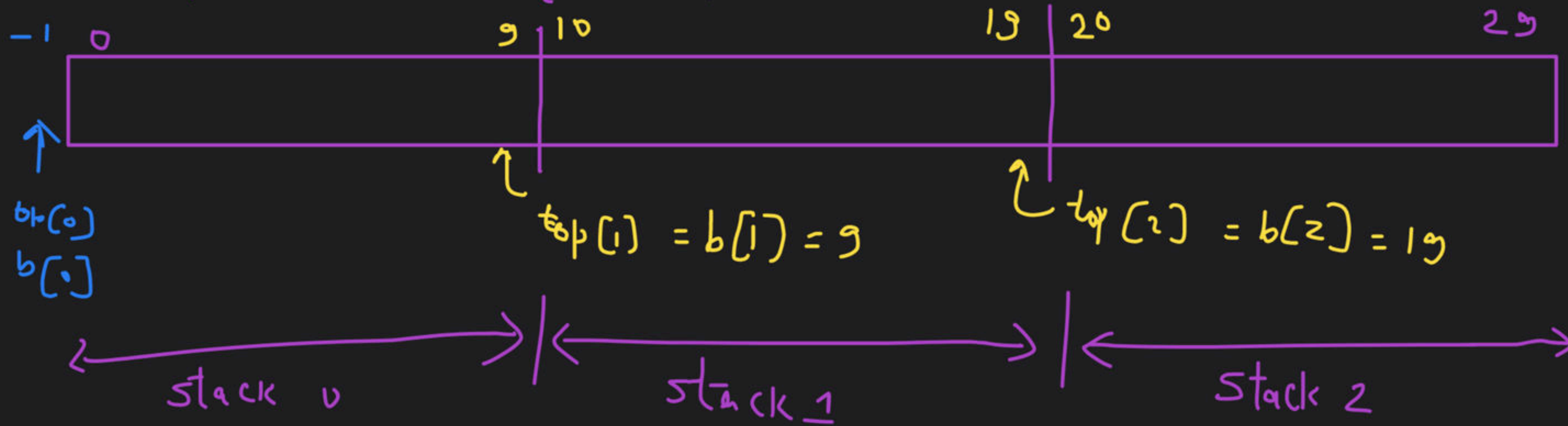
$$\text{top}[i] = \text{bottom}[i] = i * \left\lfloor \frac{n}{m} \right\rfloor - 1$$

for $i=m$

$$\text{bottom}[i] = n-1$$

Where bottom is constant index and top is variable (moving) index.

$n = 30 \mid m = 3$ stacks $(0, 1, 2)$



$b[0]$
 $b[0]$

$$top[1] = b[1] = 9$$

$$top[2] = b[2] = 19$$

for $i = 0$

$$top[0] = b[0] = 0 * \left\lfloor \frac{30}{3} \right\rfloor - \underline{1}$$

$$= \underline{-1}$$

for $i = 1$

$$top[1] = b[1] = 1 * \left\lfloor \frac{30}{3} \right\rfloor - \underline{1} = 9$$

Question 4 continue

Fill in the blank:

1.

void push (int i, char item)

```
{  
  if(_____)  
  {  
    printf("Overflow");  
  }  
  else  
  {  
    stack[++top [i]] = item;  
  }  
}
```


top[i] == bottom[i+1]

Question 4 continue

Fill in the blank:

2.

```
void pop(int i)
```

```
{  
    if(_____)  top[i] == bottom[i]  
    {  
        printf("Underflow");  
    }  
    else  
    {  
        stack[top[i]--];  
    }  
}
```


② Pre:- $- + a * bc \wedge d \wedge ef$
Post:- $a bc * + d ef \wedge \wedge -$

Question 1

⑤ Pre:- $+ + A / * B + CD F * DE$
Post:- $A B CD + * F / + D E * +$

Convert following infix notations into prefix and postfix:

1. $(A + B) * (C - D) / F - X * Y / Z$

2. $a + b * c - d \wedge e \wedge f$

3. $A + B * (C + D) / F + D * E$

4. $3 * \log(x + 1) - a / 2$

5. $a = \underline{(b)} * c \uparrow d \uparrow e + \underline{f * g / h} - i * j$

① Pre:- $- * 3 \log + x 1 / a 2$
Post:- $3 x 1 + \log * a 2 / -$

③ Pre:- $\Rightarrow a - + * - b \uparrow c \uparrow d e / * f g h * i j$
Post:- $a b - c d e \uparrow \uparrow * f g h / + i j * - =$

① Pre:-

Post:-

$- / * + A B - C D F / * X Y Z$

$A B + C D - * F / X Y * Z / -$

Question 2

Convert following notations into Infix:

1. $A B C D - * + E /$ ① $(A + B * (C - D)) / E$

2. $A B C * D / + E -$

②

$$A + B * C / D - E$$

3. $A B * C - D +$

③

$$A * B - C + D$$

4. $A B C + * D -$

④

$$A * (B + C) - D$$

Question 3

Evaluate Following Expressions using stack:

1. $+ 6 * - 3 6 7$

2. $/ 3 * 9 + 4 + 5 3$

3. $4 5 6 + - 2 5 * -$

4. $4 3 * 2 / 1 8 9 \uparrow \uparrow + \underset{1}{2} - 3 +$

5. $++ - + 2 3 1 \uparrow 1 \uparrow 3 6 / * 6 3 2$

① $- 's$

② $3 / 108$

③ $- 17$

④ 8

⑤ 14

▲ 1 • Asked by Tirthankar...

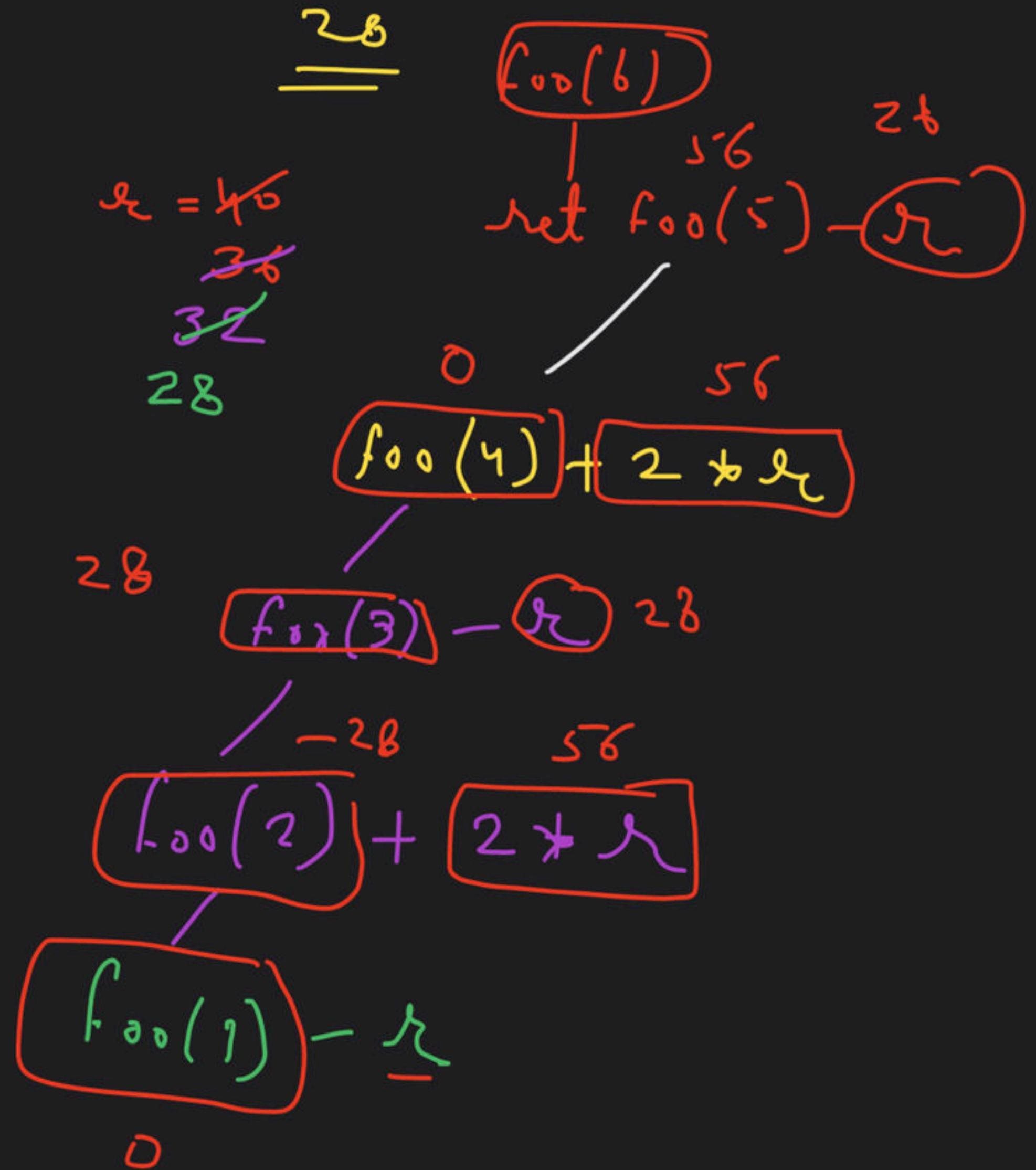
Please help me with this doubt

Q. 26

Consider the following code:

```
#include <stdio.h>
int foo(int n)
{
    static int r = 40;
    if (n == 0 || n == 1) return 0;
    if (n % 2 == 0)
    {
        r = r - 4;
        return foo(n - 1) - r;
    }
    else return foo(n - 1) + 2 * r;
}
int main( )
{
    printf("%d", foo(6));
    return 0;
}
```

The output of the above program is _____.



▲ 2 • Asked by Tirthankar...

Please help me with this doubt

Q. 12

Consider the following C code:

```
#include <stdio.h>
int main( )
{
    char str1[ ] = "Madeeasy";
    char str2[ ] = {'M', 'a', 'd', 'e', 'e', 'a', 's', 'y'};
    int m = sizeof(str1)/sizeof(str1[0]);
    int n = sizeof(str2)/sizeof(str2[0]);
    printf("m = %d, n = %d", m, n);
    return 0;
}
```

What will the values printed by the above code?

$\text{printf}("%d", \text{sizeof}(\text{str1}))$
 $\text{printf}("%d", \text{sizeof}(\text{str2}))$
 $\text{printf}("%d", \text{sizeof}(\text{str1}[0]))$
 $\text{printf}("%d", \text{sizeof}(\text{str2}[0]))$

▲ 1 • Asked by Tirthankar...

Please help me with this doubt

10/20/20

Consider the following threads T_1 , T_2 , T_3 executed on a uniprocessor system with 3 binary semaphore variables A , B and C respectively.

T_1	T_2	T_3
<code>while(TRUE)</code>	<code>while(TRUE)</code>	<code>while(TRUE)</code>
<code>{</code>	<code>{</code>	<code>{</code>
<code> P(C);</code>	<code> P(A);</code>	<code> P(B);</code>
<code> print("2");</code>	<code> print("1");</code>	<code> print("0");</code>
<code> V(B);</code>	<code> V(C);</code>	<code> V(A);</code>
<code>}</code>	<code>}</code>	<code>}</code>

If A , B , C are all initialized to 1, which of the following sequence(s) is/are possible?

$A = + \cancel{0} \cancel{1} \cancel{0} \underline{1}$

$B = \cancel{1} \cancel{0} \underline{1}$

$C = \cancel{1} \cancel{0} \underline{1}$

▲ 1 • Asked by Tirthankar...

Please help me with this doubt

Consider the following threads T_1 , T_2 , T_3 executed on a uniprocessor system with 3 binary semaphore variables A , B and C respectively.

T_1	T_2	T_3
<code>while(TRUE)</code>	<code>while(TRUE)</code>	<code>while(TRUE)</code>
<code>{</code>	<code>{</code>	<code>{</code>
<code> P(C);</code>	<code> P(A);</code>	<code> P(B);</code>
<code> print("2");</code>	<code> print("1");</code>	<code> print("0");</code>
<code> V(B);</code>	<code> V(C);</code>	<code> V(A);</code>
<code>}</code>	<code>}</code>	<code>}</code>

If A , B , C are all initialized to 1, which of the following sequence(s) is/are possible?

☒ A 012012012.....

☐ B 120120120.....

☐ C 10120120.....

☐ D 21212121.....

Happy Learning

