

Recursion

✓ 1) Trace the recursion program ✓
dny-run

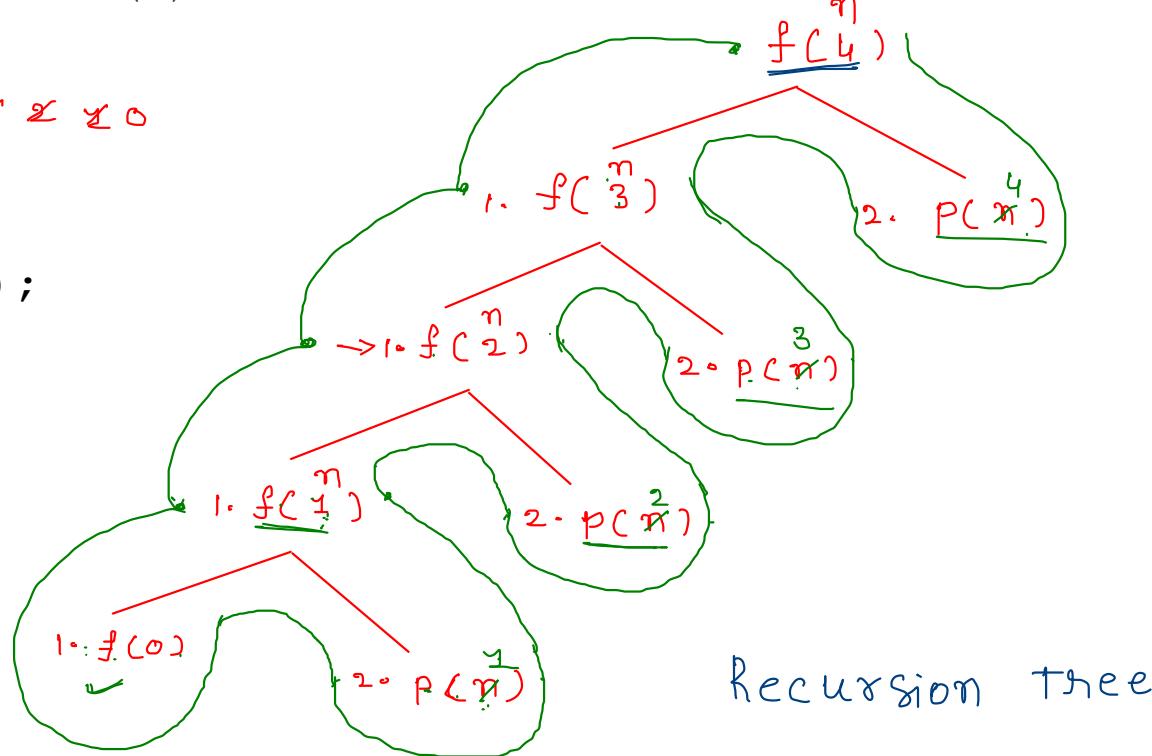
✓ 2) write the program using recursion ⇒

What is the output for $\text{fun}(4)$?

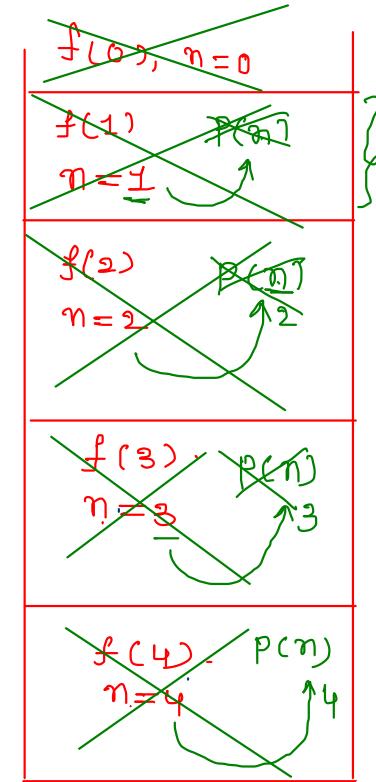
```
function fun(n)
{
    → if(n > 0)
    {
        1. fun(n - 1);
        2. print(n)
    }
}
```

o/p
1 2 3 4 ✓

activation record.



Recursion tree



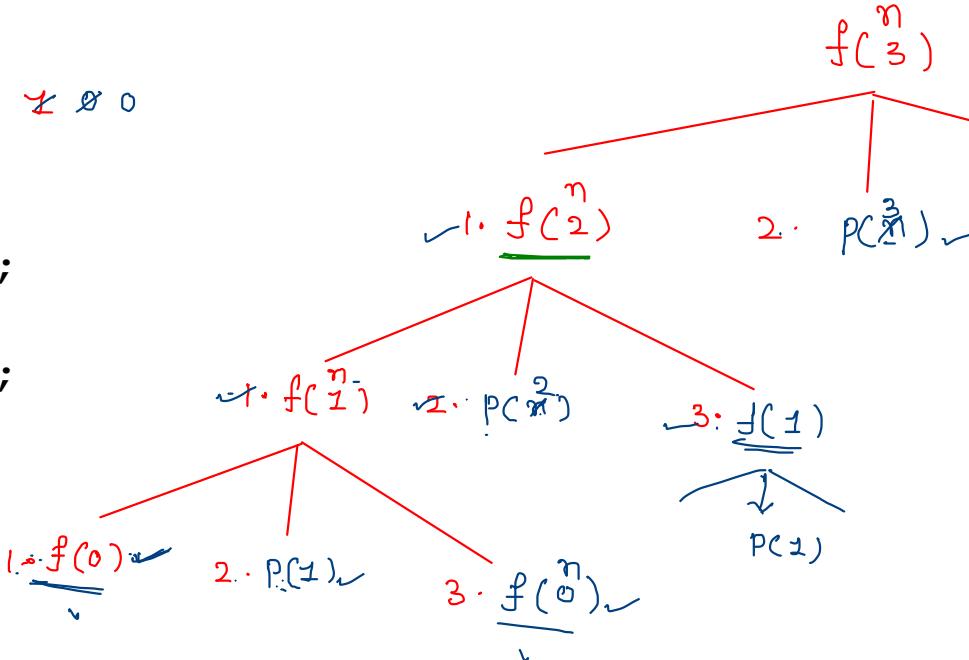
stack

top → bottom
↓↓

left → right

What is the output for $\text{fun}(3)$?

```
function fun(n)
{
    ↳ 8 2 4 8 0
    → if (n > 0)
    {
        1. fun(n - 1);
        2. print(n)
        3. fun(n - 1);
    }
}
```



Op: 1, 2, 1, 3, 1, 2, 1
↳ 1, 2, 1
↓
Op: 1, 2, 1

```

function fun(n,r)
{
    if(n>0)
        return (n%r)+fun(n/r, r) ✓
    else
        return 0 ✓
}

```

what is the return value of the function fun when it is called as fun(345,10)

Note:-

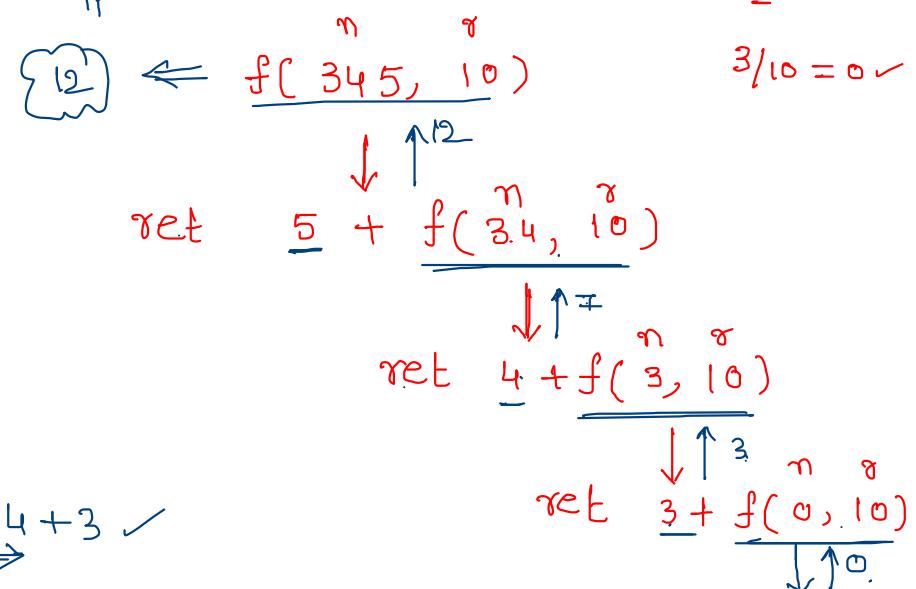
Assume n/r : return's always floor of it.

$$345 \% 10 = 5$$

$$345 / 10 = 34.5 \Rightarrow 34 ✓$$

$$\frac{34}{10} = 3$$

$$3 / 10 = 0 ✓$$



$$\xrightarrow{5+4+3} ✓$$

$\text{fun}(3, 4) = 12 \checkmark$

```

✓ function fun(a, b)
{
    if (b == 0) b ✓
        return 0;
    if (b % 2 == 0) b
        return fun(a + a, Math.floor(b/2));
    return fun(a + a, Math.floor(b/2)) + a; ✓✓
}

```

$12 = f(4, 3)$

$$\begin{array}{c} \downarrow \uparrow 12 \\ \text{ret } f(8, 1) + \cancel{a} \end{array}$$

$$\begin{array}{c} \downarrow \uparrow 12 \\ \text{ret } f(16, 0) + \cancel{a} \end{array}$$

10:45

$\begin{array}{c} a \quad b \\ \text{?} \quad 12 \\ \Leftrightarrow f(3, 4) \\ \downarrow \uparrow 12 \\ \text{ret } f(6, 2) \\ \downarrow \uparrow 12 \\ \text{ret } f(12, 1) \end{array}$

$\begin{array}{c} \downarrow \uparrow 12 \\ \text{ret } f(24, 0) + \cancel{12} \\ \downarrow \uparrow 0 \end{array}$

$f(6, 2) \checkmark$
 $f(7, 1) \checkmark$

$\text{fun}(5, 3) \checkmark$

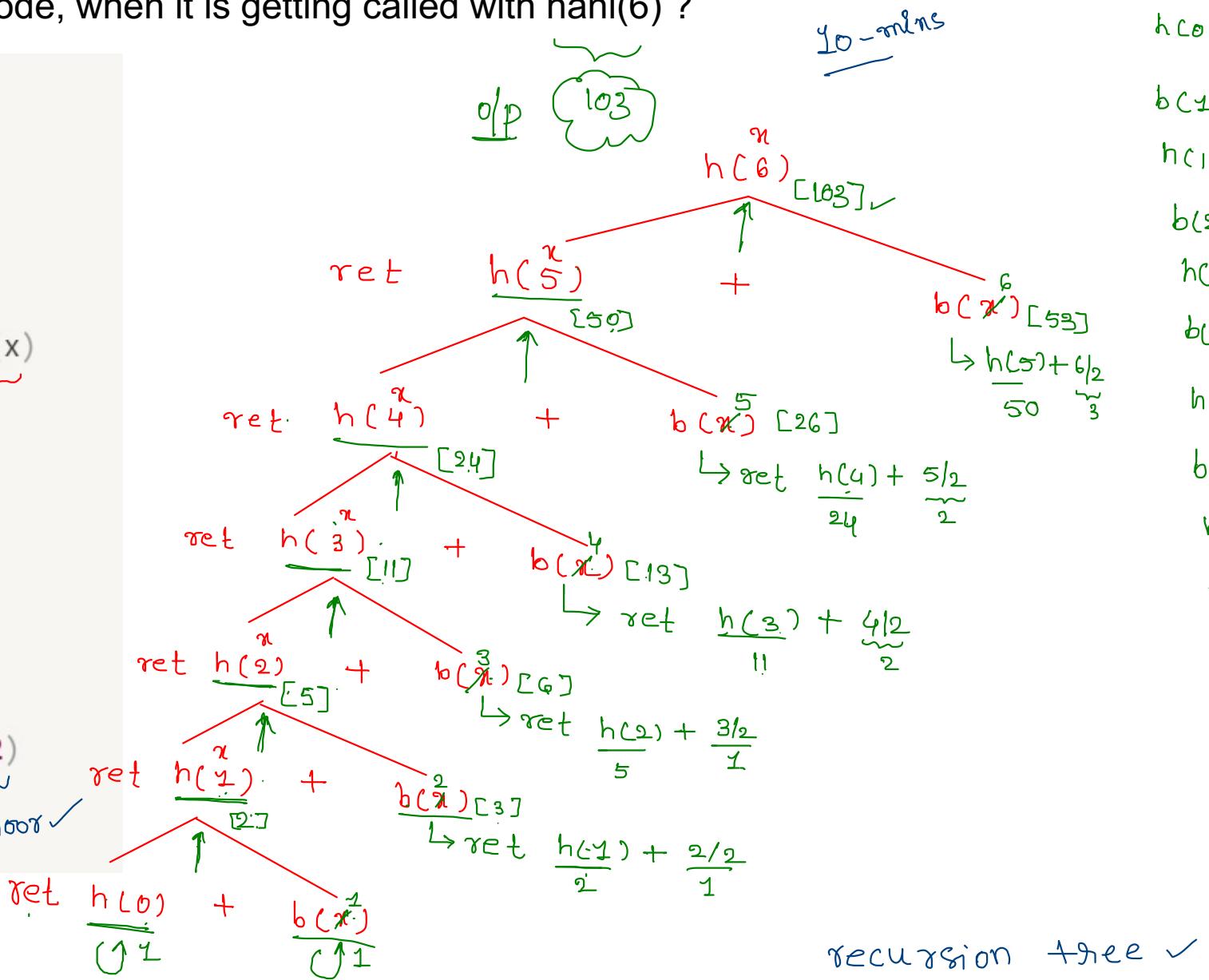
what is the return value of the code, when it is getting called with $hani(6)$?

```

function hani(x)
{
    if(x<1) ✗✗✗✗✓
        return 1✓
    else
        return hani(x-1)+bani(x)
}

function bani(x)
{
    if(x<2) ✗✗✗✗✓
        return 1
    else
        return hani(x-1)+ (x/2)
    }

```



```
✓function fun(count)
{
    print(count);

    if(count < 3)
    {
        fun(fun(fun(++count)));
    }

    return count;
}

main()
{
    res=fun(1)
    print(res)
}
```

Factorial-Recursion

-84:8:48

Edit

Description

The factorial of a positive integer N is the product of all positive integers less than or equal to n:

Given a number N your task is to write a program that calculates factorial of N

Input

Input Format

First and the only line contains N

Constraints

N<15

Output

Output the factorial value of N

Sample Input 1

5

Sample Output 1

120

Hint

• Rec. problem.

✓ ① applied on some number
 (n) ✓

$$\text{factorial}(5) = ?$$

② applied on some array.
Find sum of all
array elements.
Using recursion.

$$5! = 5 \times 4 \times \underbrace{3 \times 2 \times 1}_{4!}$$

$$f(5) = 5 \times 4!$$

$$f(n) = n * f(n-1)$$

$$f(2) = 2 \checkmark$$

$$f(1) = 1 \checkmark$$

$$\underline{f(0) = 1 \checkmark}$$

$$f(-1) = ?$$

$$n \Rightarrow 5$$



$$n-1 \Rightarrow 4$$



$$n-2 \Rightarrow 3$$



$$2$$



$$1$$



$\boxed{0} \checkmark \Rightarrow$ stopping

$$f(3) = 3 \times f(1)$$



$$2 \times f(1)$$



$$1 * f(0)$$



$$0 * f(-1)$$



$$-1 * f(-2)$$



intuitively

start with bigger
values, slowly decrement
for next iteration

```

function fact(n)
{
    if(n==0)
        return 1
    else
        return n*fact(n-1)
}

```

$\times \rightarrow \textcircled{n}$

$5 \times \underline{4} \times 3 \times 2 \times \underline{1}$

$\frac{5}{1} \checkmark$

top $[x \neq 3 \vee \underline{x}]$

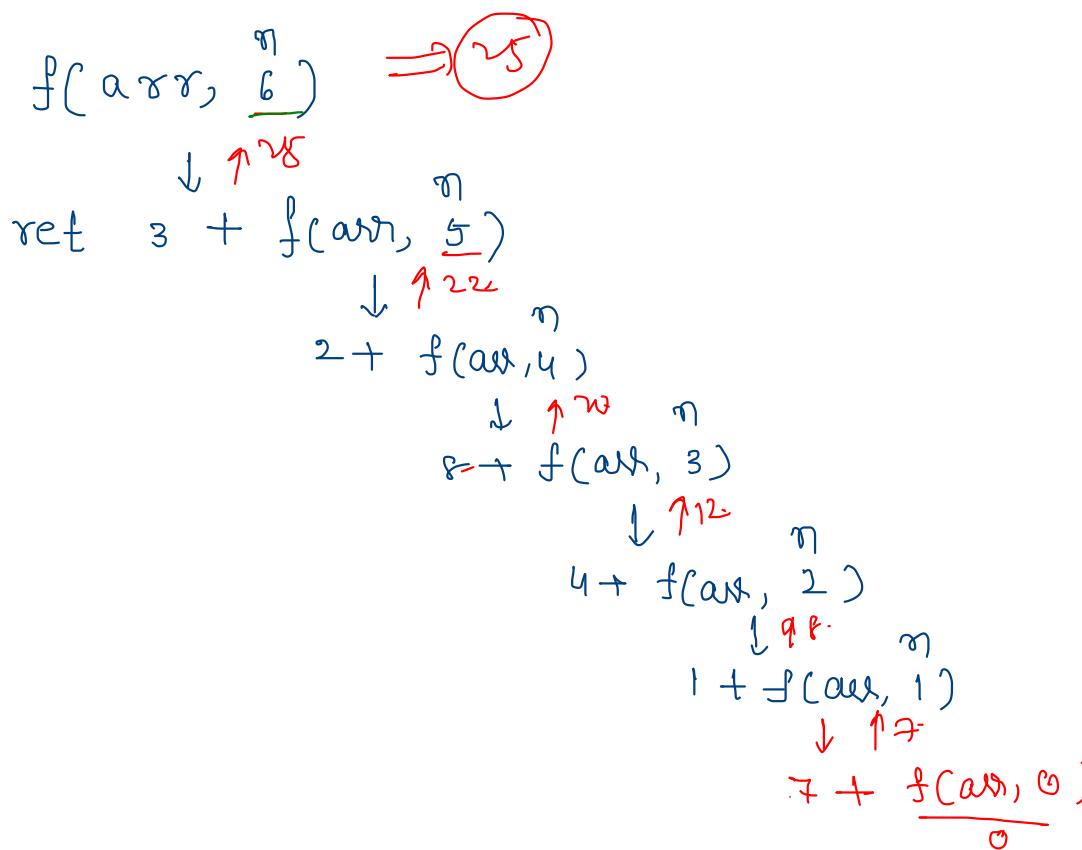
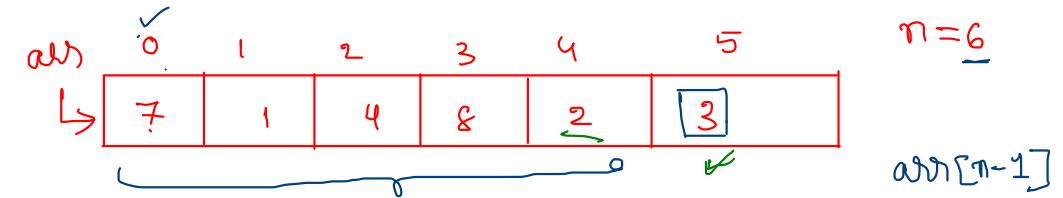
$f(5) = 120 \checkmark$
 $\downarrow \uparrow$
 $5 * f(4)$
 $\downarrow \uparrow$
 ret $4 * f(3)$
 $\downarrow \uparrow$
 ret $3 * f(2)$
 $\downarrow \uparrow$
 ret $2 * f(1)$
 $\downarrow \uparrow$
 ret $1 * f(0)$
 $\downarrow \uparrow$

0
 5
 \downarrow
 4
 \downarrow
 3
 \downarrow
 2
 \downarrow
 1
 \downarrow
 stop

```

function fun(arr[], n)
{
    if(n==0)
        return 0 ✗ ✓
    else
        → return arr[n-1]+fun(arr,n-1)
}

```



$$\begin{aligned}
& n \swarrow \\
& \downarrow \\
& 3 + \underbrace{(n-1)}_{\downarrow} \\
& \downarrow \\
& 2 + (n-2)
\end{aligned}$$

Description

Given an array of non-negative integers, and a value 'sum', determine if there is a subset of the given set (array) with sum equal to given sum.

If there is a subset whose sum is equal to the required sum then print "yes" else print "no" without quotes.

Input

Input Format :

The first line of input contains an integer N - denoting size of the array.

The second line contains the N space seperated integers.

The third line of input will contain S - the required Sum value.

iP
arr[]✓

Sum

Constraints :

$1 \leq n \leq 18$

$1 \leq A[i] \leq 10^9$

Output

Print "yes" or "no" without quotes.

Sample Input 1

9
1 2 3 4 5 6 7 8 9 ✓
5 ✓

Sample Output 1

yes

Subset:-

possible sub-sets

$$\checkmark A = \{1, 2, 3\} \Rightarrow \{\underline{\underline{\cdot}}\}, \{\underline{1}\}, \{\underline{1, 2}\}, \{\underline{1, 2, 3}\}$$

0 → not present ✓ ↓ ↓ ↓ $\Rightarrow 2 \times 2 \times 2 = \frac{3}{2} = 8$ ⇒ 8 subsets

1 → present ✓ 2 2 2

$\{\underline{1, 3}\}$ is circled in green and crossed out with a red line.

Sum = 4 ✓

1 2 3

0 0 0

1 0 0

0 1 0

0 0 1

1 1 0

1 0 1

0 1 1

1 1 1

Yes

$$A = \{1, 2, 3, 4\}$$

$$\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow = 16 \quad \text{✓}$$

n ele's $\Rightarrow 2^n$ subsets

Sum Exists or Not

● -83:

Description

Given an array of non-negative integers, and a value 'sum', determine if there is a subset of the given set (array) with sum equal to given sum.

If there is a subset whose sum is equal to the required sum then print "yes" else print "no" without quotes.

Input

Input Format :

The first line of input contains an integer N - denoting size of the array.

The second line contains the N space seperated integers.

The third line of input will contain S - the required Sum value.

Constraints :

$1 \leq n \leq 18$

$1 \leq A[i] \leq 10^9$

Output

Print "yes" or "no" without quotes.

Sample Input 1

```
9
1 2 3 4 5 6 7 8 9
5
```

Sample Output 1

```
yes
```

arr ↦

0	1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8	9

$n = 9$

sum = 5

sum = 10

$a[i] = 3$

⇒ 7

ss(arr, n, sum)

→ $a[i]$ one element

$a[i] > sum$

x

ss(arr, n-1, sum)

$a[i] \leq sum$

x

ss(arr, n-1, sum),

ss(arr, n-1, sum - a[i])

{5} ✓

{1, 4} ✓ → {~~1, 4~~} ✓

{2, 3} ✓

⇒ un-ordered
collection of
distinct elems

or

||

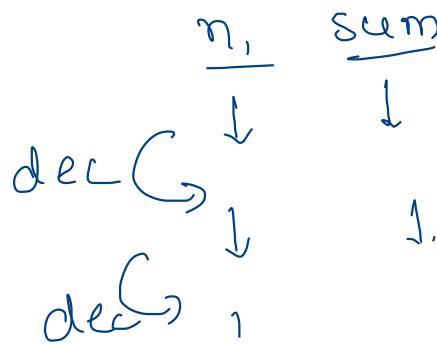
```

function ss(arr[], n, sum)
{
    if(sum==0)
        return true
    if(n==0)
        return false

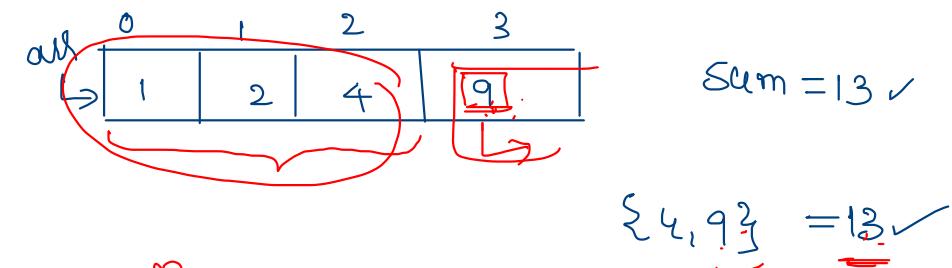
    if(arr[n-1]>sum) // don't want to include X
        return ss(arr, n-1, sum)

    else
        return ss(arr, n-1, sum) || ss(arr, n-1, sum-arr[n-1])
}

```



$n = 0$	$n = 5 \checkmark$
$\text{sum} = \frac{1}{2} \dots$	$\text{sum} = 0 \cancel{\checkmark}$
	empty sub-set



$\frac{n}{\downarrow}$
 $\text{ss}(\text{arr}, 4, 13)$

$\frac{\text{ss}(\text{arr}, 3, 3)}{\text{F} \downarrow} \quad \parallel \quad \text{ss}(\text{arr}, 3, 4)$
 $\hookrightarrow ?$