



# CS & IT ENGINEERING



## PYQ SERIES

C Programming and Data Structure

Lecture No.- 01

By- Pankaj Sharma Sir



# Topics to be Covered

Topic

C Programming





## Topic : 2022 : CS

ISRO PYQ SERIES

P  
W

#Q. Consider the following C code:

```
#include<stdio.h>
int *assignval (int *x, int val) {
    *x = val;
    return x;
}
int main () {
    int *x = malloc(sizeof(int));
    if (NULL == x) return;
    x = assignval (x,0); assignval (1024,0)
    if (x) {
        x = (int *)malloc(sizeof(int));
        if (NULL == x) return;
    }
}
```

*(Note: dynamic memory allocate)*

```
x = assignval (x,10);
}
printf("%d\n", *x);
free(x);
}
```

The code suffers from which one of the following problems:





## Topic : 2022 : CS

ISRO PYQ SERIES

P  
W

#Q. Consider the following C code:

```
#include<stdio.h>
int *assignval (int *x, int val) {
    *x = val;
    return x;
}
int main () {
    int *x = malloc(sizeof(int));
    if (NULL == x) return;
    x = assignval (x,0); assignval (1024,0)
    if (x) {
        x = (int *)malloc(sizeof(int));
        if (NULL == x) return;
    }
}
```

*(Note)*  
*dyn. memory allocate*

```
x = assignval (x,10);
}
printf("%d\n", *x); 10
free(x);
}
```

The code suffers from which one of the following problems:



**A**

compiler error as the return of malloc is not typecast appropriately.

**B**

compiler error because the comparison should be made as `x==NULL` and not as shown.

**C**

compiles successfully but execution may result in dangling pointer.

**D**

compiles successfully but execution may result in memory leak.



## Topic : 2022: CS

ISRO PYQ SERIES

P  
W

#Q. What is printed by the print statements in the program P1 assuming call by reference parameter passing?

Program P10

{

x = 10;

y = 3;

func1(y,x,x);

print x;

print y;

}

func1(x,y,z)

{

y = y+4;

z = x+y+z;

}

- A 10, 3
- ~~B 31, 3~~
- C 27, 7
- D 33, 5



## Topic : 2022: CS

ISRO PYQ SERIES

P  
W

#Q. What is printed by the print statements in the program P1 assuming call by reference parameter passing?

Program P10

{

x = 10;

y = 3;

func1(y,x,x);

print x; → 31

print y; → 3

}

func1(x,y,z)

{

\*y = \*y + 4;

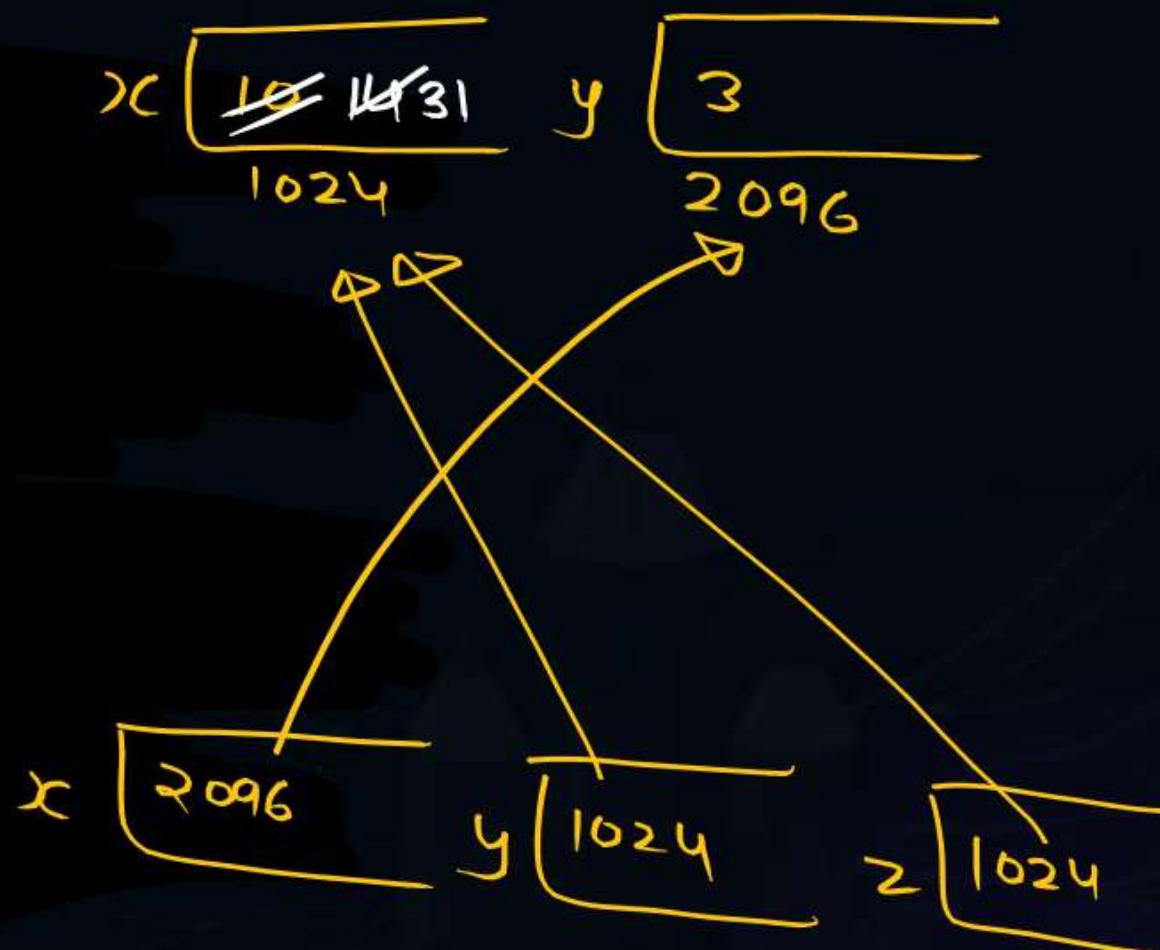
\*z = \*x + \*y + \*z;

}

func1(y,x,x)

\*y = \*y + 4

$$\begin{aligned} *z &= 3 + 14 + 14 \\ &= 31 \end{aligned}$$





## Topic : 2022: CS

ISRO PYQ SERIES

P  
W

#Q. What does the following C-statement declare?

int ( \* f) (int \* );

10 sec

**A** A function that takes an integer pointer as argument and returns an integer.

**B** A function that takes an integer as argument and returns an integer pointer.

**C** A pointer to a function that takes an integer pointer as argument and returns an integer.

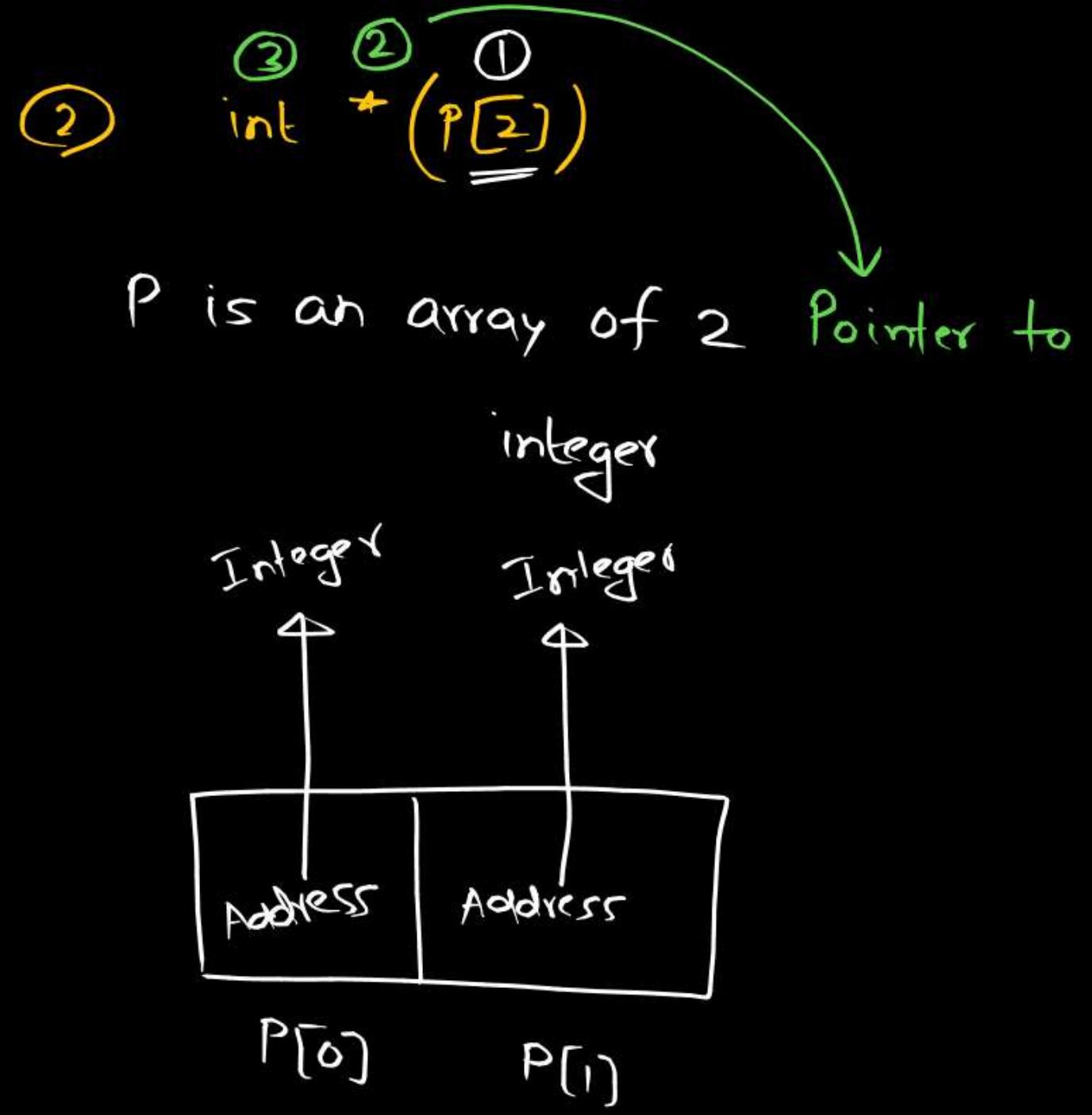
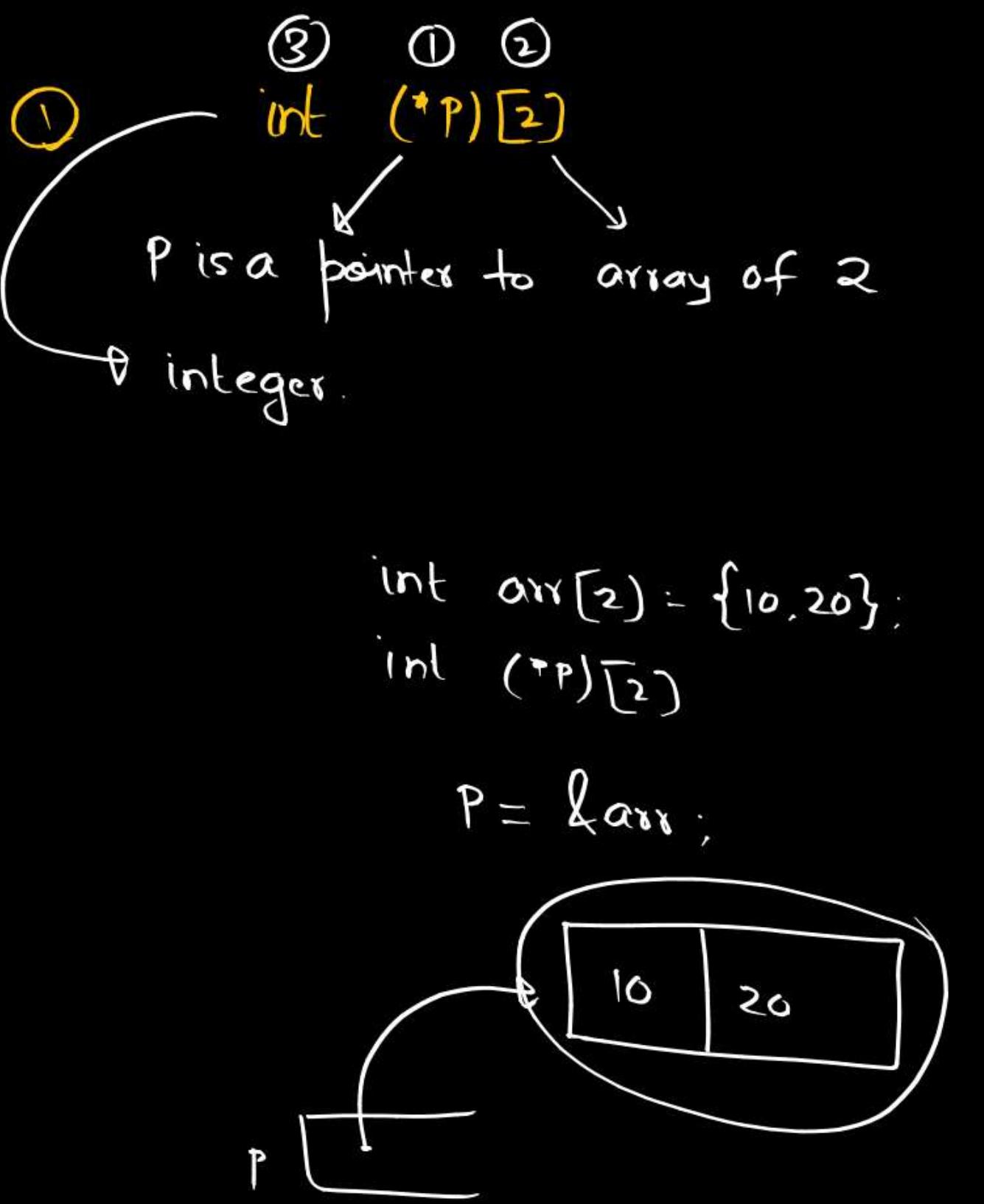
**D** A function that takes an integer pointer as argument and returns a function pointer

declaration

①	( )	function	①	L to R
②	[ ]	Array		
③	*	Pointer	②	R to L
④	Identifier	Name		
⑤	data-type	int, float, char	③	

int \*P;

- ① int is a pointer P.
- ② pointer is a int P
- ③ P is a \_\_\_\_\_



int ( \*f )( int, int );

f is a pointer to a function that takes 2  
integer arguments and return  
integer value.

int Add( int, int );



## Topic : 2022: CS

ISRO PYQ SERIES

P  
W

#Q. Consider these two functions and two statements S1 and S2 about them

```
int work1(int *a, int i, int j)
{
```

```
    int x = a[i+2];
```

```
    a[j] = x+1;
```

```
    return a[i+2] - 3;
```

```
}
```

```
int work2(int *a, int i, int j) {
```

```
    int t1 = i+2;
```

```
    int t2 = a[t1];
```

```
    a[j] = t2+1;
```

```
    return t2 - 3;
```

```
}
```

a

i = 0  
j = 2

x [3]

1	2	4	4	5	6
0	1	2	3	4	5

x = a[0+2)

a[2] = 3+1

return a[2]-3

4-3

return 1



## Topic : 2022: CS

ISRO PYQ SERIES

P  
W

#Q. Consider these two functions and two statements S1 and S2 about them

```
int work1(int *a, int i, int j)
{
    int x = a[i+2];
    a[j] = x+1;
    return a[i+2] - 3;
}
```

*work2*      a  
i=0  
j=2

```
int work2(int *a, int i, int j) {
    int t1 = i+2;
    int t2 = a[t1];
    a[j] = t2+1;
    return t2 - 3;
}
```

1	2	4	4	5	6
0	1	2	3	4	5

$$\begin{aligned}t1 &= 2 \\t2 &= a[2] = 3 \\a[2] &= a[2] + 1 \\&= 4\end{aligned}$$

return  $t2 - 3$   
 $3 - 3$   
0

- S1: The transformation from work1 to work2 is valid, i.e., for any program state and input arguments, work2 will compute the same output and have the same effect on program state as work1
- S2: All the transformations applied to work1 to get work2 will always improve the performance (i.e reduce CPU time) of work2 compared to work1

**A**

S1 is false and S2 is false



**C**

S1 is true and S2 is false

**B**

S1 is false and S2 is true

**D**

S1 is true and S2 is true



#Q. Consider the following two functions.

```
void fun1(int n) {  
    if(n == 0) return;  
    printf("%d", n);  
    fun2(n - 2);  
    printf("%d", n);  
}  
  
void fun2(int n) {  
    if(n == 0) return;  
    printf("%d", n);  
    fun1(++n);  
    printf("%d", n);  
}
```

A

53423122233445

B

53423120112233

C

53423122132435

D

53423120213243

The output printed when fun1(5) is called is



## Topic : 2022 : CS

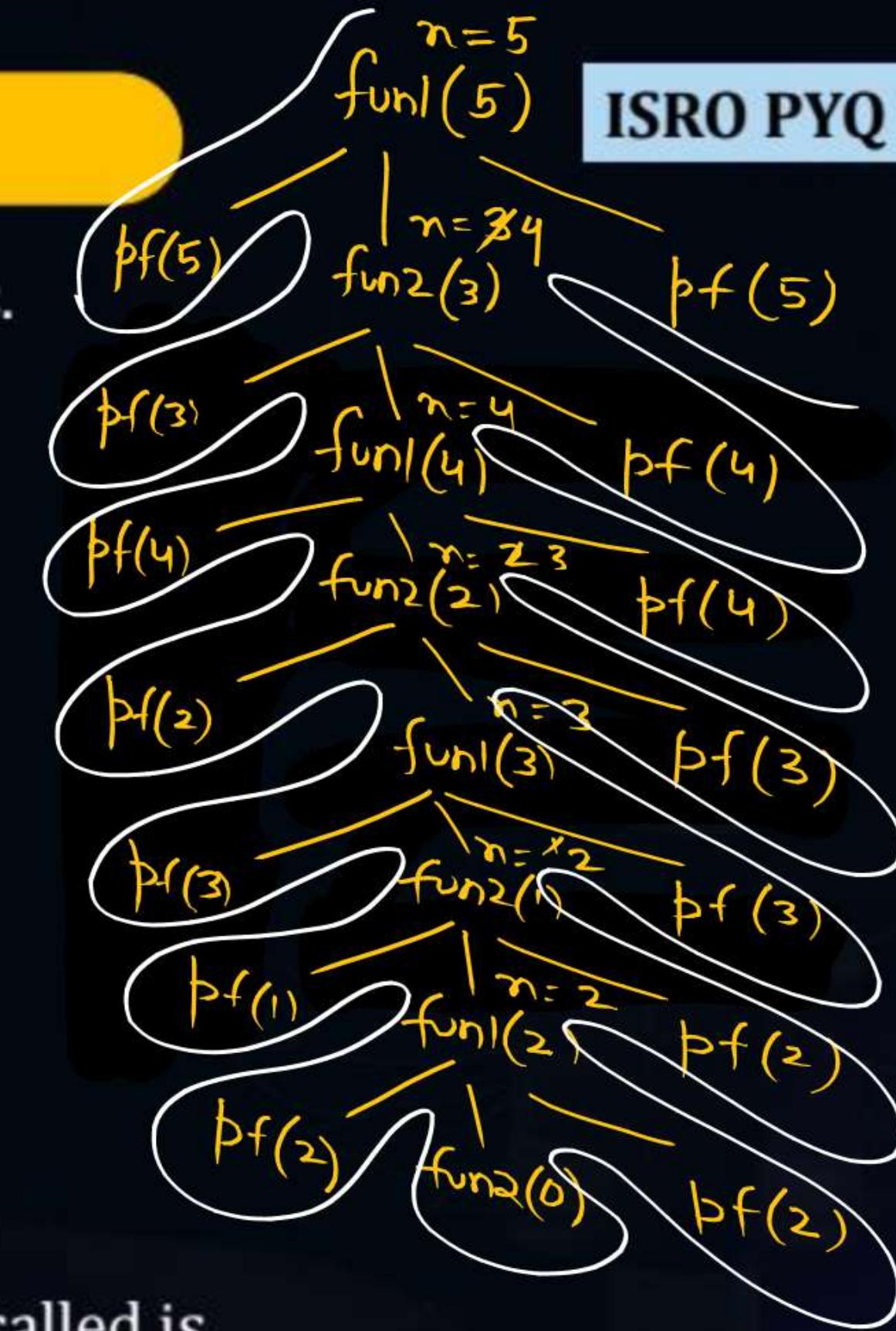
#Q. Consider the following two functions.

```
void fun1(int n) {  
    if(n == 0) return;  
    printf("%d", n);  
    1. fun2(n - 2);  
    2. printf("%d", n);  
}
```

```
void fun2(int n) {  
    if(n == 0) return;  
    printf("%d", n);  
    1. fun1(++n);  
    2. printf("%d", n);  
}
```

ISRO PYQ SERIES

P  
W



The output printed when `fun1(5)` is called is



## Topic : 2017 : CS

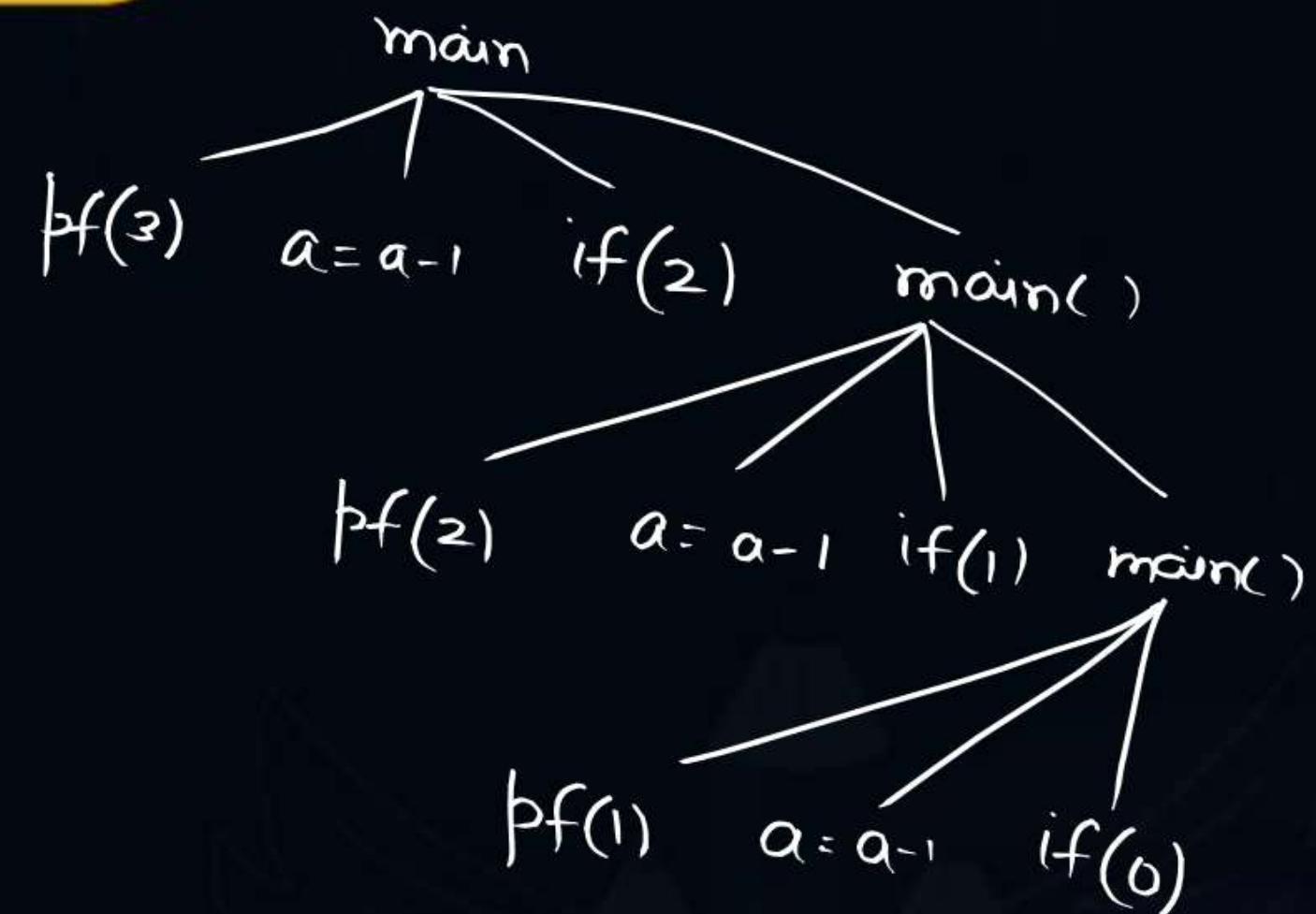
ISRO PYQ SERIES

P  
W

#Q. What will be the output of following?

```
main()
{
    static int a = 3;
    printf("%d", a--);
    If (a)
        main();
}
```

a 3 2 1 0



A 3

C 333

B 321

D Program will fall in continuous loop and print 3



#Q. What will be the output of following?

```
main()
```

```
{
```

```
    int a = 3;
```

```
    printf("%d",a--);
```

```
    If (a)
```

```
    main();
```

```
}
```



A 3

C 333

main( )

B 321

D Program will fall in continuous loop and print 3



## Topic : 2017: CS

#Q. The following program fragment prints  
int i = 5;  
do { putchar(i+100); printf ("%d", i-;) }  
while (i);



i5h4g3f2el



14h3g2f1e0



C an error message



D none of the above

5432

putchar(105)

(104)

(103)

97 - a

98 - b

99 - c

100 - d

101 - e

102 - f

103 - g

104 - h

105 - i

106 - j

107 - k

108 - l

109 - m

n o

p q

r s

t

ISRO PYQ SERIES

P  
W



## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Consider the following declaration,

```
int a, *b = &a, **c = &b;
```

```
a = 4;
```

```
**c = 5;
```

If the statement

```
b = (int*) **c
```

Is appended to the above program fragment then

a 5  
1000

b 1000  
2000

c 2000

b = (int\*) 5

A

Value of b becomes 5

C

Value of b is unaffected

B

Value of b will be the address of c

D

None of these



## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

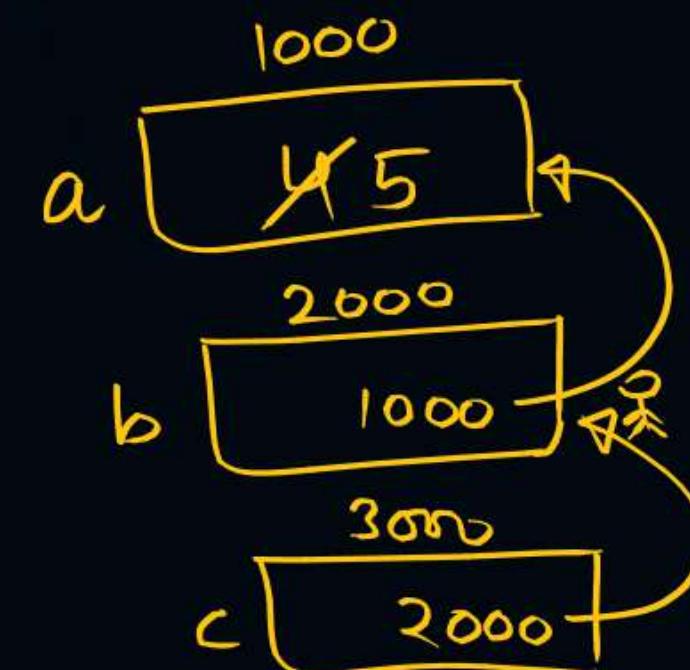
#Q. Consider the following declaration,

```
int a, *b = &a, **c = &b;
```

```
a = 4;
```

```
**c = 5;
```

*a*  $\Rightarrow$



A

Value of *b* becomes 5

C

Value of *b* is unaffected

B

Value of *b* will be the address of *c*

D

None of these



## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Let A be a square matrix of size  $n \times n$ . ( $n \geq 1$ )

Consider the following program. What is the expected output?

$C = 100$

for  $i = 1$  to  $n$  do

for  $j = 1$  to  $n$  do

{

    Temp =  $A[i][j] + C$

$A[i][j] = A[j][i]$

$A[j][i] = Temp - C$

}

for  $i = 1$  to  $n$  do

for  $j = 1$  to  $n$  do

Output ( $A[i][j]$ );

$i = 1, 2$   
 $j = 1, 2$

①  $i = 1, j = 1 \Rightarrow Temp = A_{11} + C = 1 + 100 = 101$

$$A_{11} = A_{11}$$

$$A_{11} = Temp - C = 101 - 100 = 1$$

②  $i = 1, j = 2 \Rightarrow Temp = A_{12} + C = 102$

$$A_{12} = A_{21}$$

$$A_{21} = 102 - 100 = 2$$

③  $i = 2, j = 1 \Rightarrow Temp = A_{21} + C = 102$

$$A_{21} = A_{12}$$

$$A_{12} = 102 - 100 = 2$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}$$

↑

**A**

The matrix A itself

**B**

Transpose of matrix A

**C**

Adding 100 to the upper diagonal elements and subtracting 100 from diagonal elements of A

**D**

None of the option

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$



## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Output of following program

```
//include <stdio.h>
int main()
{
    int i = 5;
    printf("%d %d %d", i++, i++, i++);
    return 0;
}
```

① Sequence point  
② side effect  
③ order of eval



A

7 6 5

C

7 7 7

B

5 6 7

D

Compiler Dependent



## Topic : 2017 : CS

#Q. Output of following program ?

```
#include<stdio.h>
void dynamic(int s,...)
{
    printf("%d", s);
}
int main()
{
    dynamic(2, 4, 6, 8);
    dynamic(3, 6, 9);
    return 0;
}
```

*variadic functions*

- A 2 3
- B Compiler Error
- C 4 3
- D 3 2

```
int Add(int x, int y)  
{  
    return x+y;  
}
```

```
void main()  
{  
    pf("Add", Add(10,20));  
}
```

Add(10);  
Add(10,20,30);

Error

✓ printf(" %d %d", a, b);

✓ printf(" %d %d %d", a, b, c);

[variadic function] A (int s, ...)

→ var. arguments



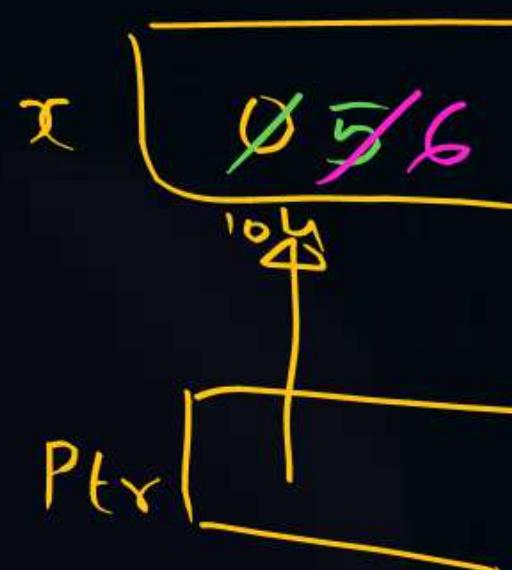
## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Output of following program?

```
# include <stdio.h>
int main()
{
    int *ptr;
    int x;
    ptr = &x;
    *ptr = 0;
    printf(" x = %d\n", x);  $\star_{ptr} = 0$ 
    printf("*ptr = %d\n", *ptr);
    *ptr += 5;  $\star_{ptr} = \star_{ptr} + 5$ 
    printf("x = %d\n", x);  $\star_x = 5$ 
}
```



```
printf(" *ptr = %d\n", *ptr);
(*ptr)++;
printf(" x = %d\n", x);  $\star_x = 6$ 
printf ("*ptr = %d\n", *ptr);
return 0;
}
```

$$\star_{ptr} = \star_{ptr} + 1$$

**A**

x = 0  
\* ptr = 0  
x = 5  
\* ptr = 5  
x = 6  
\* ptr = 6

**B**

x = garbage value \*ptr = 0  
x = garbage value \*ptr = 5  
x = garbage value \*ptr = 6

t.me\PWpanRajsirP

**C**

x = 0  
\*ptr = 0  
x = 5  
\*ptr = 5  
x = garbage value  
\*ptr = garbage value

**D**

x = 0  
\*ptr = 0  
x = 0  
\*ptr = 0  
x = 0  
\*ptr = 0

C videos →

Doubt ?

6 PM

## 2 mins Summary



THANK - YOU

/



# CS & IT ENGINEERING



## PYQ SERIES

Programming & Data  
Structures

Lecture No.- 02

By- Pankaj Sharma Sir



# Recap of Previous Lecture



Topic

Topic

Topic

# Topics to be Covered

Topic

C Programming





#Q. Assume that float takes 4 bytes, predict the output of following program.

```
#include <stdio.h>
int main()
{
    float arr[5]={12.5,10.0,13.5,90.5,0.5};
    float *ptr1 = &arr[0];
    float *ptr2 = ptr1 + 3;
    printf("%f", *ptr2);
    printf("%d", ptr2 - ptr1);
    return 0;
}
```

A

90.500000 3

B

90.500000 12

C

10.000000 12

D

0.500000 3



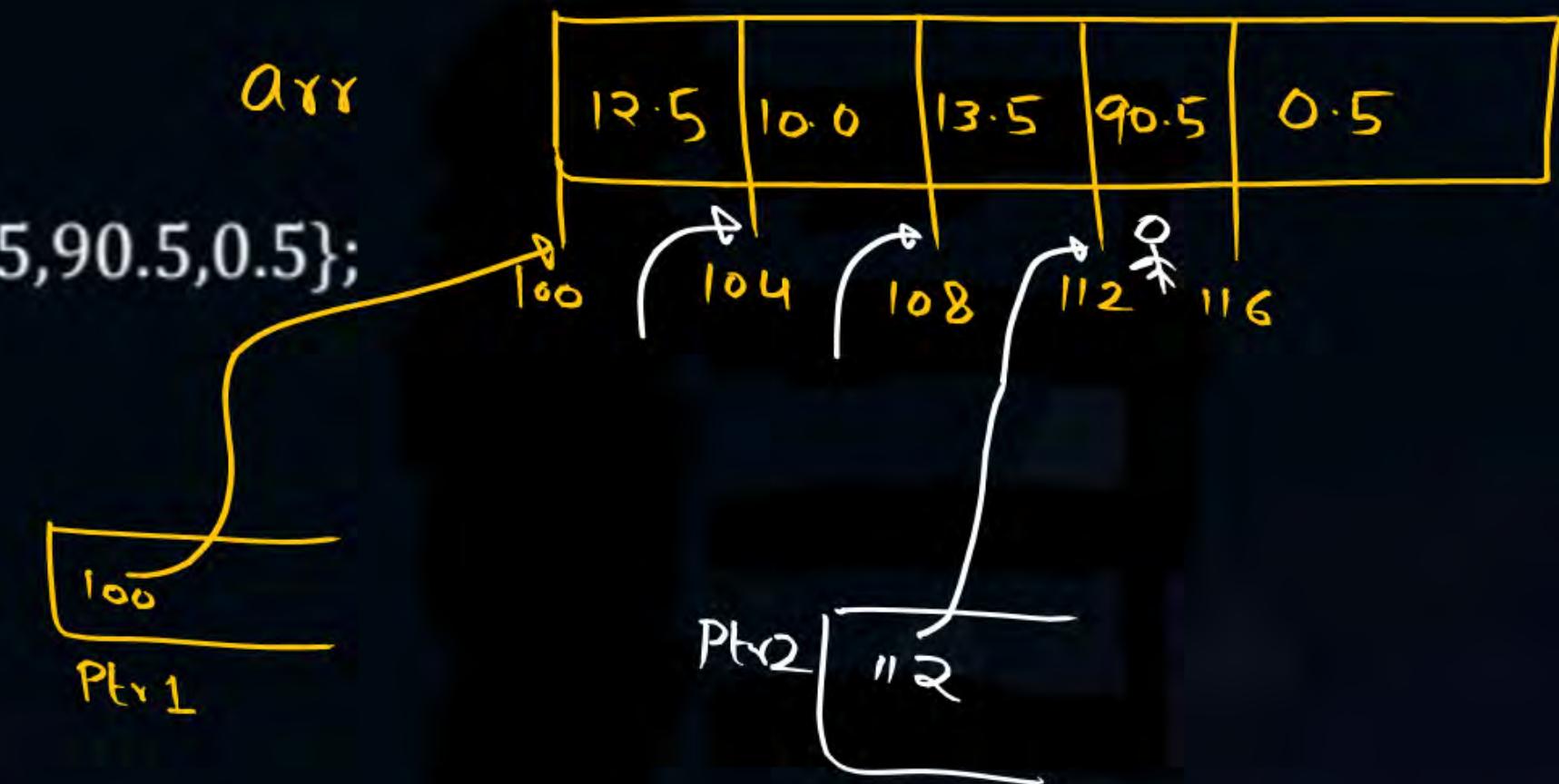
## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Assume that float takes 4 bytes, predict the output of following program.

```
#include <stdio.h>
int main()
{
    float arr[5]={12.5,10.0,13.5,90.5,0.5};
    float *ptr1 = &arr[0];
    float *ptr2 = ptr1 + 3;
    printf("%f", *ptr2);
    printf("%d", ptr2 - ptr1);
    return 0;
}
```



\* `ptr2` → value at (Memory loc. 112)  
90.5



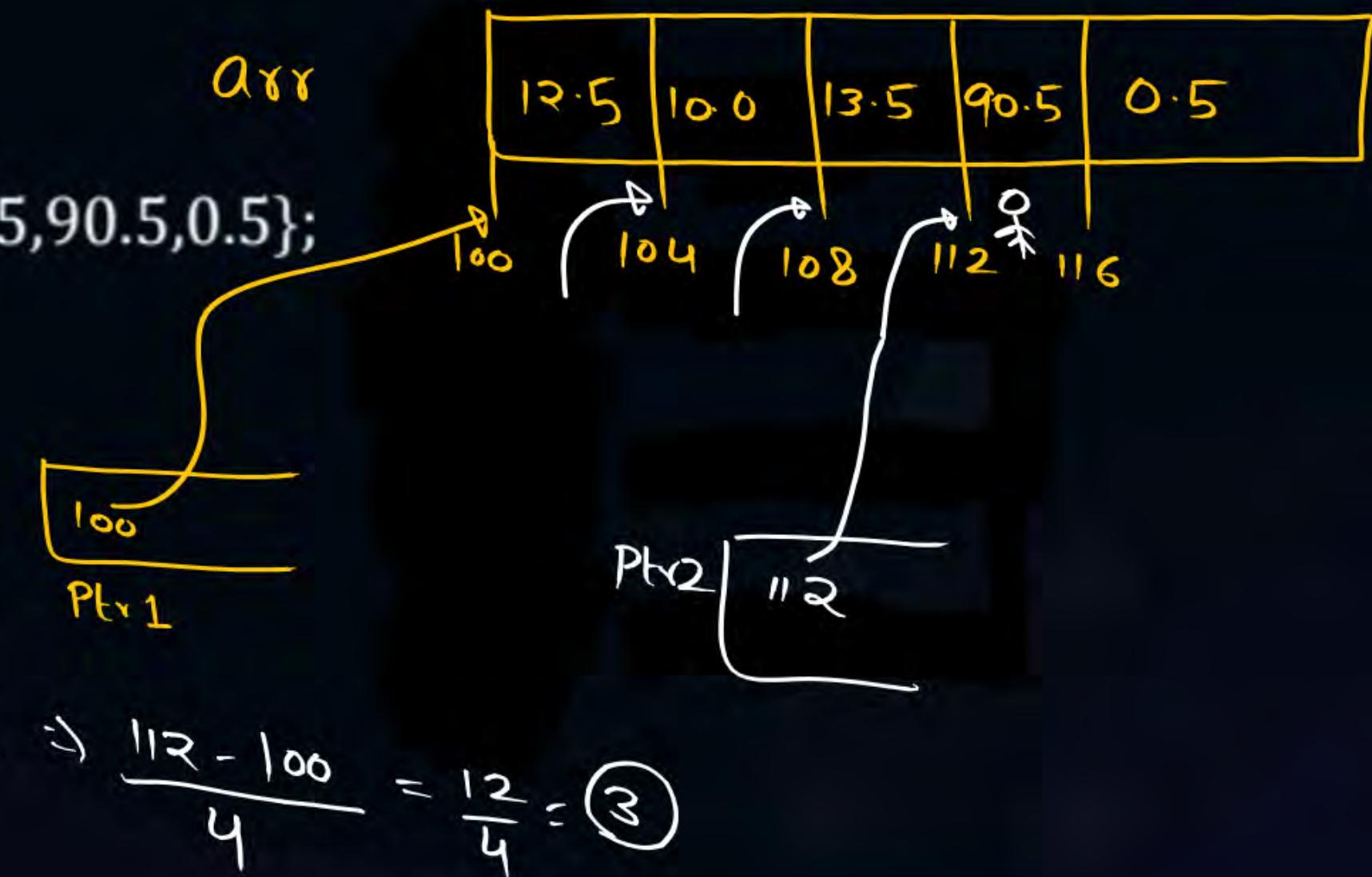
## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Assume that float takes 4 bytes, predict the output of following program.

```
#include <stdio.h>
int main()
{
    float arr[5]={12.5,10.0,13.5,90.5,0.5};
    float *ptr1 = &arr[0];
    float *ptr2 = ptr1 + 3;
    printf("%f", *ptr2);
    printf("%d", ptr2 - ptr1);
    return 0;
}
```





1 min

#Q. Assume that size of an integer is 32 bit. What is the output of following ANSI C program ?

```
#include<stdio.h>
struct st
{
    int x;
    static int y;
};
int main()
{
    printf("%d, sizeof(struct st));
    return 0;
}
```

*\* \* \* a structure can't have any static members*

- A 4
- B 8
- C Compiler Error
- D Runtime Error



## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Consider the following C declaration  
struct {

① short s[5];  $\rightarrow 5 \times 2 \Rightarrow 10$

union {

float y;  $\rightarrow 4$

long z;  $\rightarrow 8$

}u;

}t;

$10 + 8 \Rightarrow 18$

$sizeof(u) = max(4, 8) = 8$

Assume that objects of the type short, float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively. The memory requirement for variable t, ignoring alignment considerations, is

A

22 bytes

C

18 bytes ✓

B

14 bytes

D

10 bytes



Output of the following program ?

#Q. #include<stdio.h>

```
structure  
connect  
be  
recursive  
#include <stdio.h>  
  
struct st  
{  
    ① int x;  
    ② struct st next;  
};  
int main()  
{  
    struct st temp;  
    temp.x = 10;  
    temp.next = temp;  
    printf("%d", temp.next.x);  
    return 0;  
}
```

*Extra 'r'*

A

B

C

D

Compiler Error ✓

10

Runtime Error

Garbage value



Topic : 2016 : CS

ISRO PYQ SERIES

PW

#Q.

What will be output of the following C code ?

by default signed

```
#include<stdio.h>
int main() {
    char c = 125;
    c = c + 10;
    printf("%d", c);
    return 0;
}
```

}

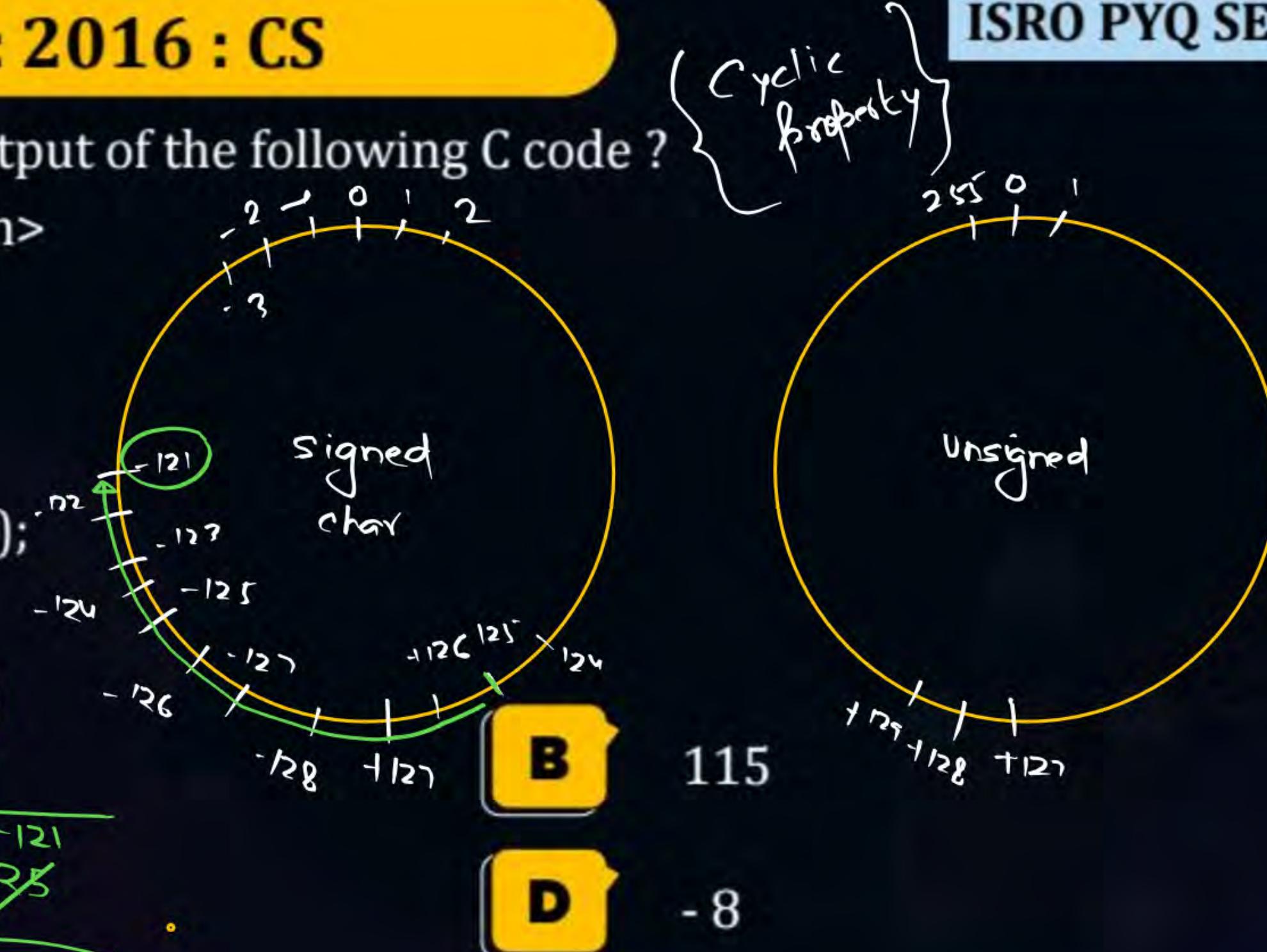
135

A

C

115

- 8





#Q. What will be output if you will compile and execute the following c code ?

```
void main0 ()  
printf ("%d", sizeof(5.2));  
}
```

→ double → atleast size ≥ 8B

- A 4
- C 8

- B 2
- D 16



## Topic : 2016 : CS

ISRO PYQ SERIES

P  
W

#Q. What is the meaning of following declaration?

```
int(*p[7])();
```

- A** p is pointer to function.
- B** p is pointer to such function which return type is array.
- C** p is array of pointer to function.
- D** p is pointer to array of function.

{  
① structure  
② complex dec.  
③ variadic function  
④ Array & pointers}

videos

Honest

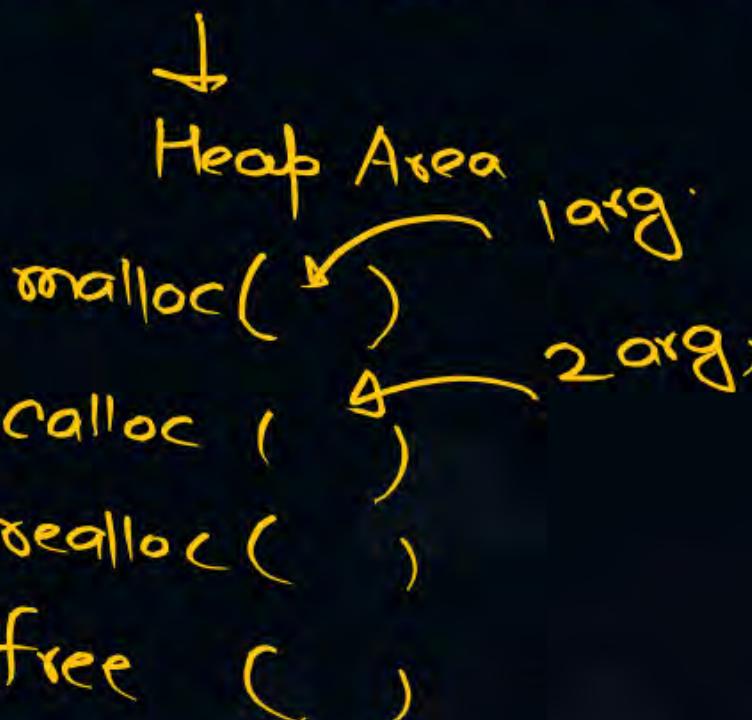


#Q. How will you free the allocated memory ?

- A remove (var-name)
- B free(var-name)
- C delete(var-name)
- D dalloc (var-name)

malloc is cheaper but not reliable  
calloc is expensive but reliable

- { (1) malloc ( ) }
- (2) calloc ( , )
- (3) realloc ( , )
- (4) free ( )



memory leakage  
problem

# C language

- ① cyclic property → int, long int, char
- ② expression → statement with same value

→ 2 · 3  
→ 2 + 3  
→ !2 + 3

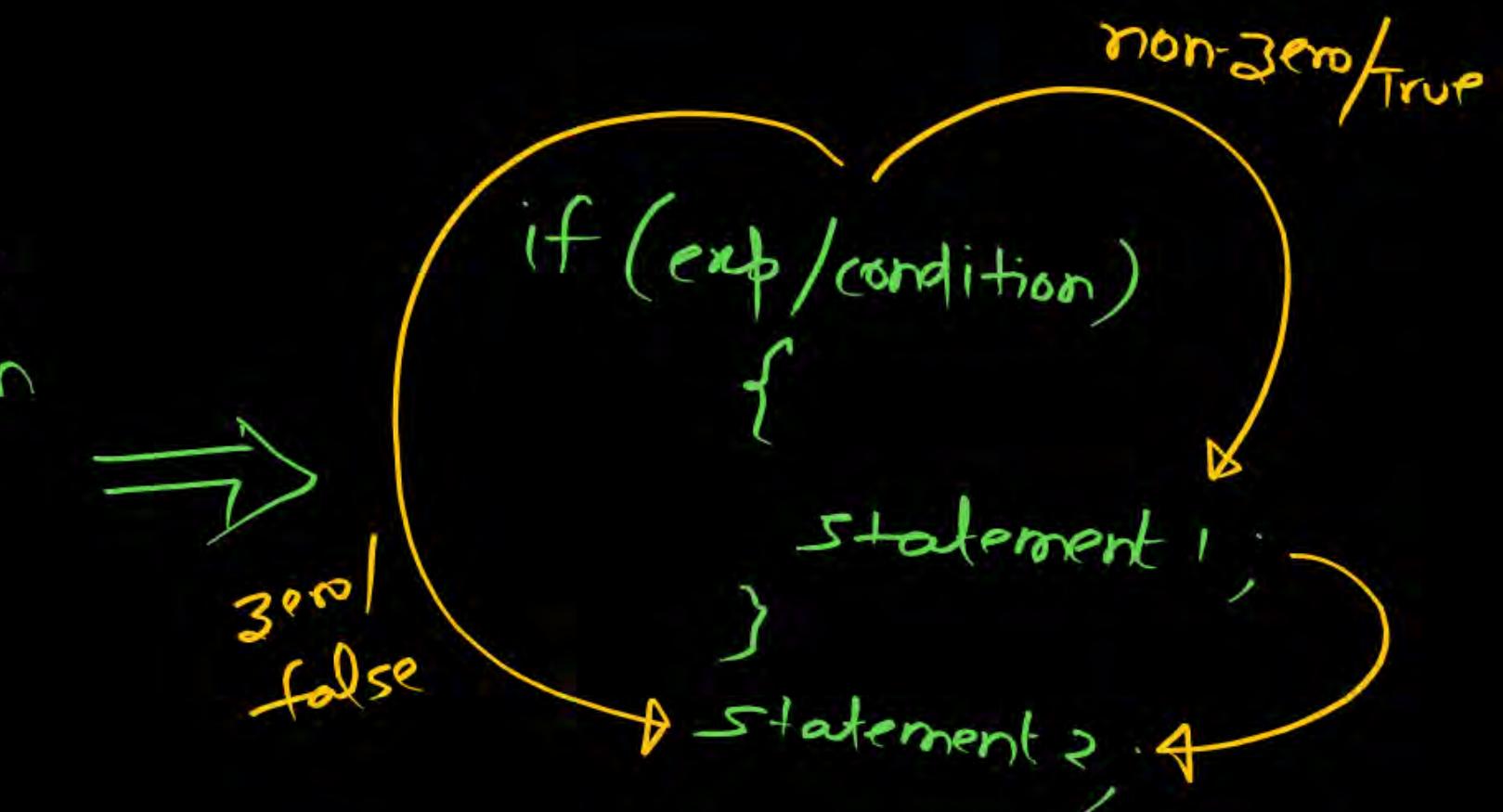
void main() {

(1) ;  
2 + 3 ;  
}

```
void main() {  
    S1;  
    S2;  
    S3;  
}
```

non-zero — true  
zero — false

if ( expression/condition )  
statement 1 ;  
statement 2 ;



if (exp / condition)

{

True /  
non-zero

false / zero

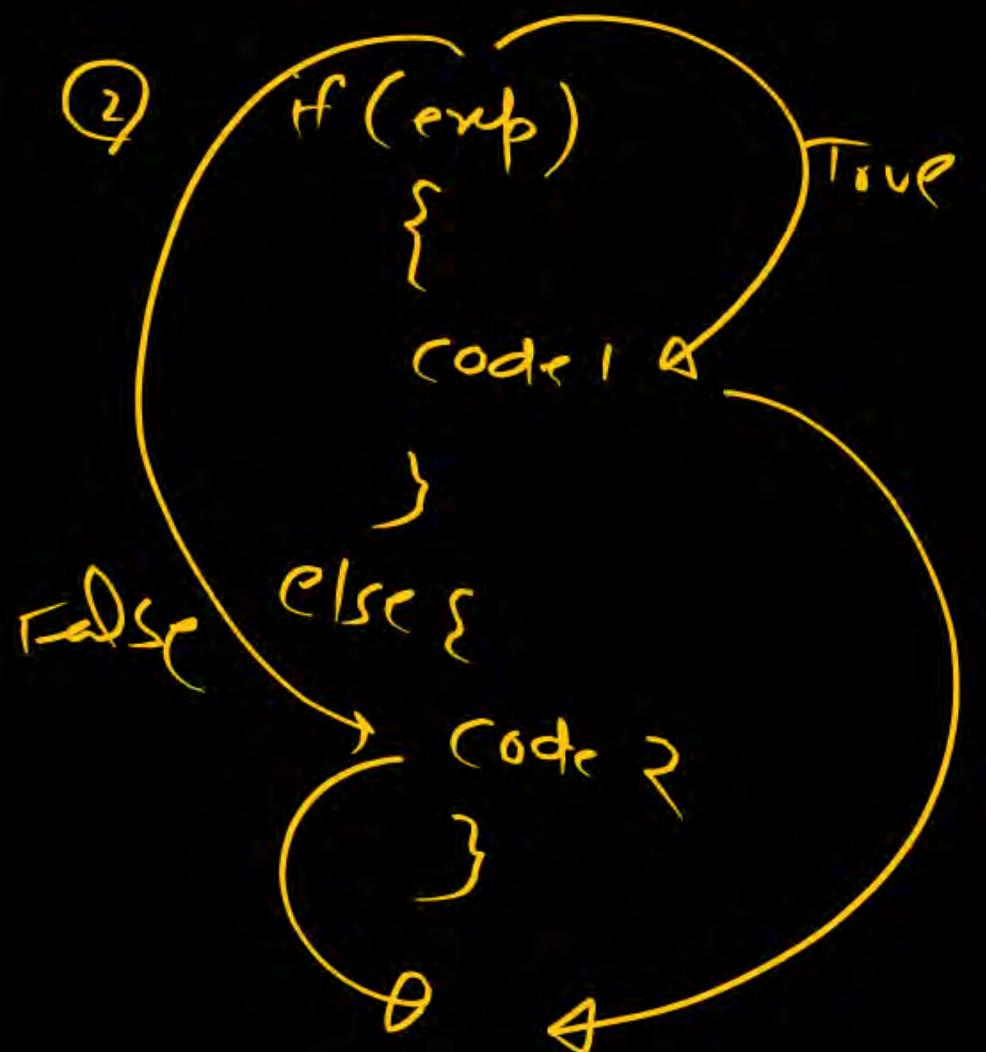
code

0

}

q

① if(exp)  
{  
    code  
}



③ if(exp1)  
{  
    code 1  
}  
else if(exp2)  
{  
    code 2  
}  
else if(exp3)  
{  
    code 3  
}  
else {  
    code 4  
}

↓

Exactly one among code1, code2, code3, code4 execute

```
if(exp1)
{
    code 1
}
else if(exp2)
{
    code 2
}
else if(exp3)
{
    code 3
}
```

It must  
will execute

```
if(0)
{
    code 1
}
else if(0.0)
{
    code 2
}
else if(0)
{
    code 3
}
```

```
if (exp)
{
    )
    → und ke float
    printf("Hello");
}
else {
    =
}
```

```
switch (exp)
{
```

Case label1 :

```
    —
    break;
```

Case label2 :

```
    —
    break;
```

default :

```
    —
    break;
```

```
}
```

- ① break optional
- ② default is optional

```
switch(1){
```

Case 2 : pf("2");

```
break;
```

Case 10 : pf('0'),  
break,

```
}
```

- ③ Position of default does not matter.

④

switch(2){

valid

}



dummy  
switch

⑤

switch(3), ✓

⑥

switch(); X

⑦

switch(orb)

{

integer

}

Compiler  
Joan  
ie lego

switch(10.2){

=  
=

int a=2, b=3;

① switch(exb)

- a) a + b
- b) a - b
- c) !a
- d) !2
- e) 2 + 3 \* 4
- f) printf("Hello"),
- g) 'A'
- h) 'A' + 2

case label → constant

int a=2, b=3

switch(10){

case a : =  
=

Can't be variable

case label?

case 2 :

case  $2+3 \times 4$ :

case 2||4

Valid Case A → char literal

Case 'A' + z ;

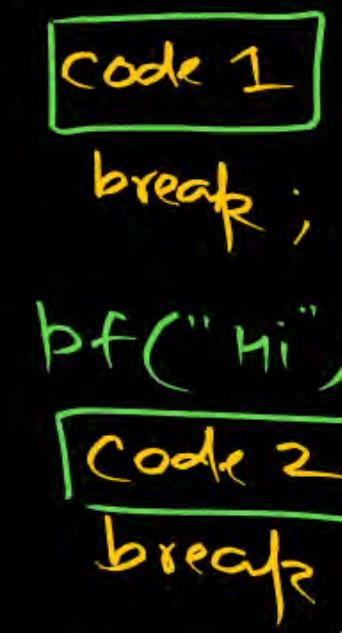
duplicate  
case labels  
not allowed

Case 2: code  
break;

Case 3-1: code  
break;

Exit

switch(exp){

    Case 10 : 

```
        pf("Hello");
        Code 1
        break;
        pf("Hi")
        Code 2
        break;
```

    Case 2 :

}

ignore (will not execute)

① `for(exp1; exp2; exp3){  
 code  
}`

② `for(exp1; exp2; exp3)`

$s_1,$

$s_2,$

$\rightarrow$

First semicolon

for compiler

`for(exp1; exp2; exp3){`

$s_1;$

$s_2;$

③ All 3 exp. are optional

`for( ; ; )  
{  
 code  
}`

valid

for (①; ②; ③)

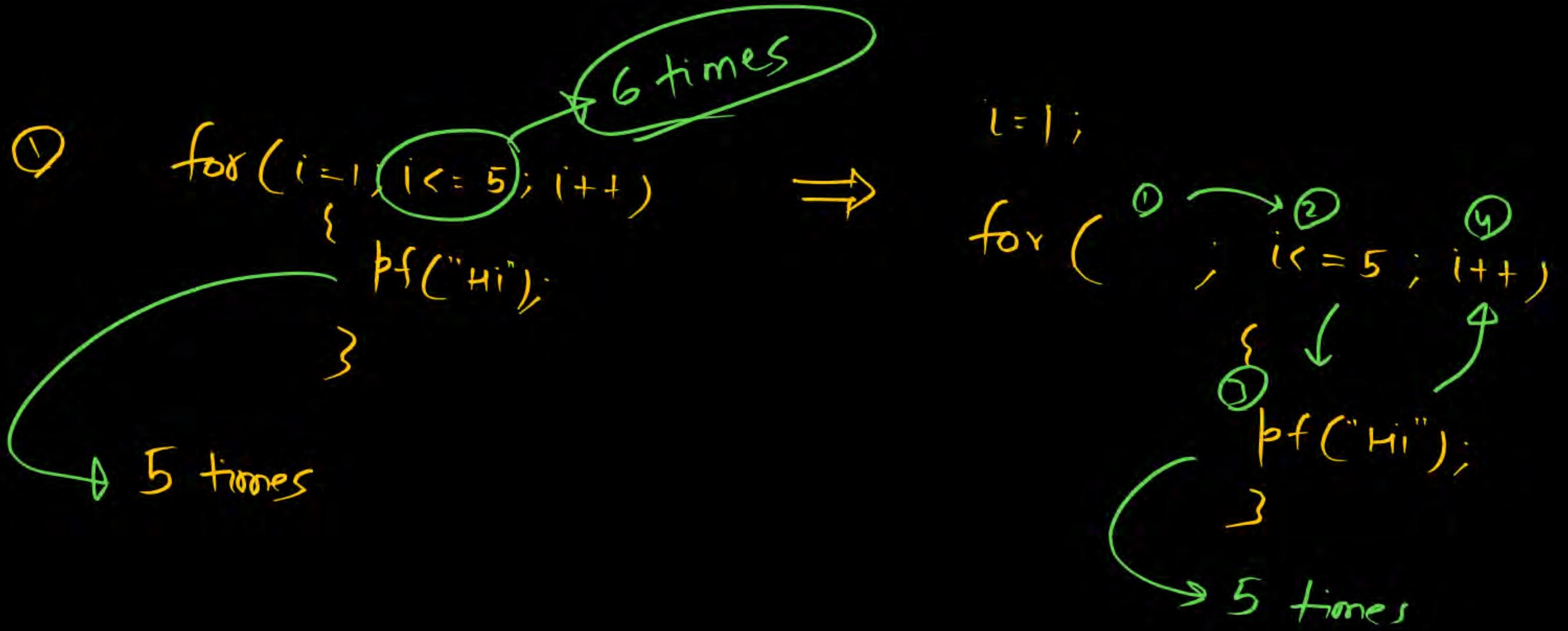
{  
    ④ non-zero  
    ⑤ code

zero

}

→

for ( Initialization ; condition; Inc/dec )  
{  
    code  
}



```
i = 1  
for ( ; i <= 5 ; )  
{  
    pf("Hi");  
    i = i + 1;  
}  
  
for ( ; i )  
{  
    // assume true  
}
```



```
while(exp/cond )  
{  
    Code  
}
```

white( ) → vđ ke  
 laat

```
do {  
    Code  
} while(exp);
```

①

```
for( i=1 ; i<=0 ; i++ )  
{  
    pf("Hi");  
}
```

②

```
i=1  
while( i<=0 )  
{  
    pf("Hi");  
    i = i + 1;  
}
```

③

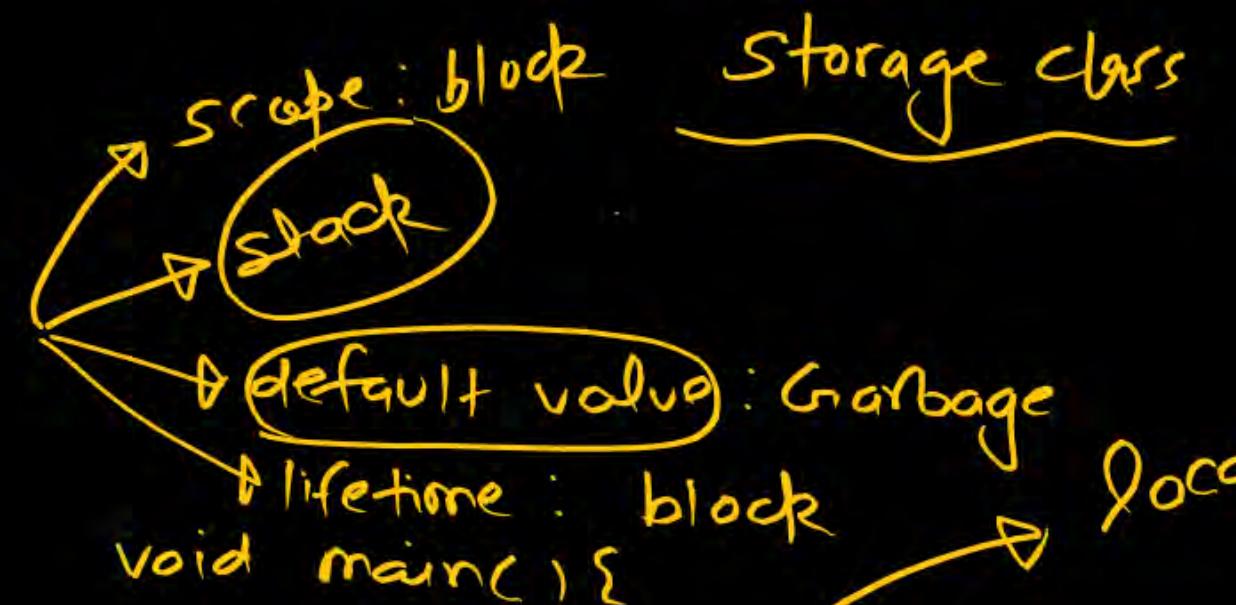
```
i=1 ;  
do {  
    pf("Hi");  
} while( i<=0 );
```

At least 1

time

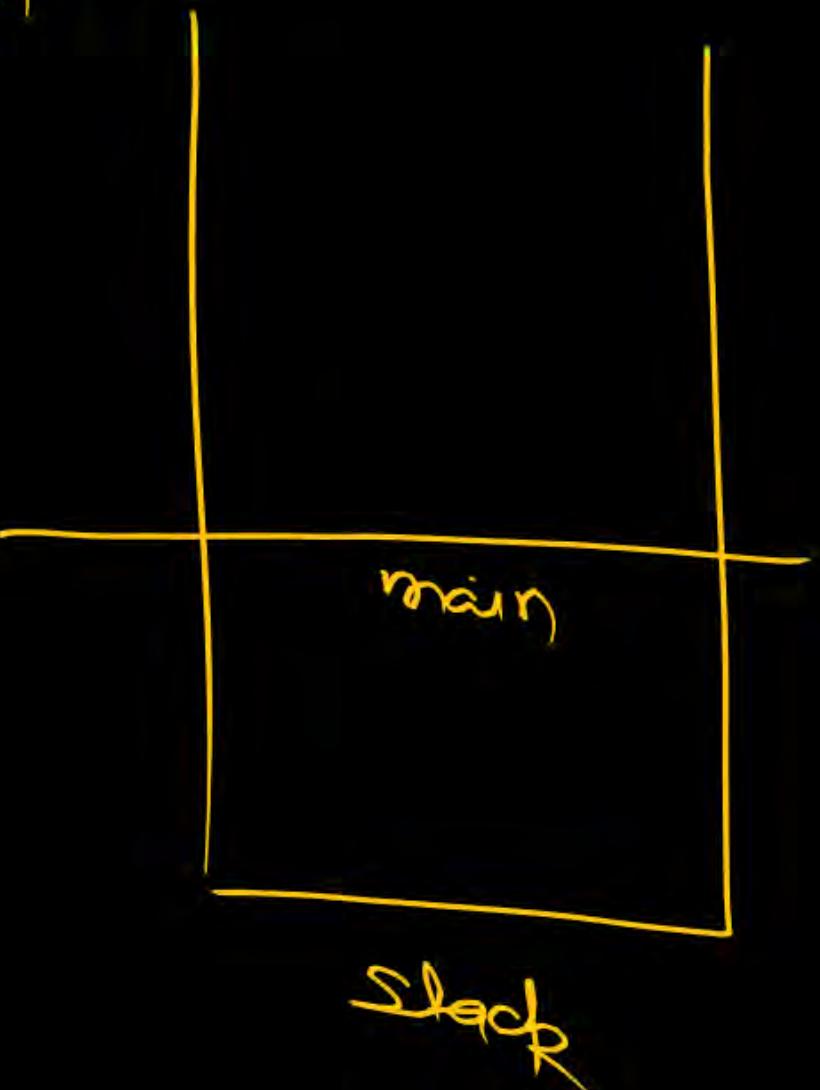
→ code ✓

① auto / local



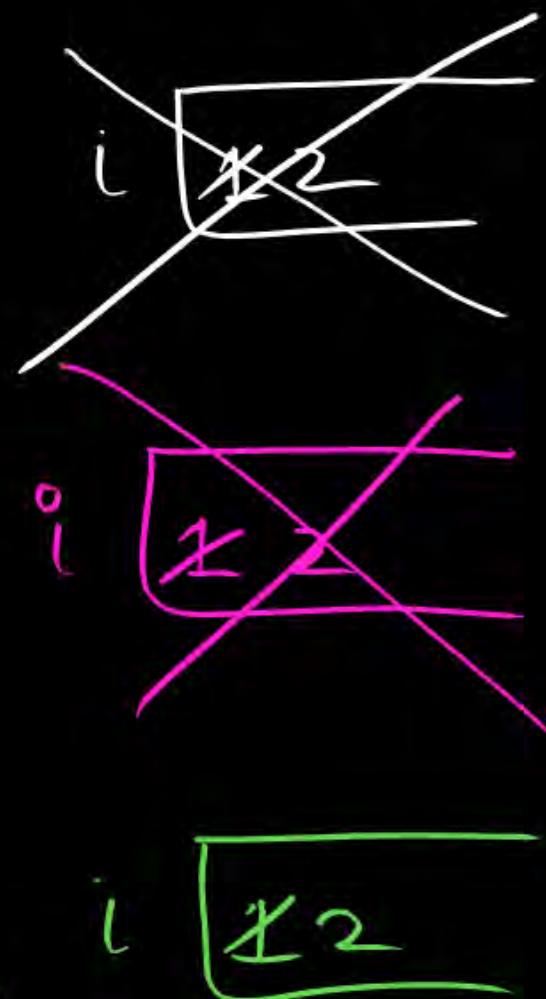
```
void main(){  
    int a;  
    pf("%d",a);  
}
```

int(a,b);  
=



222

```
void f() {  
    int i = 1;  
    ++i;  
    pf("%d", i);  
}  
  
void main() {  
    f();  
    f();  
    f();  
}
```



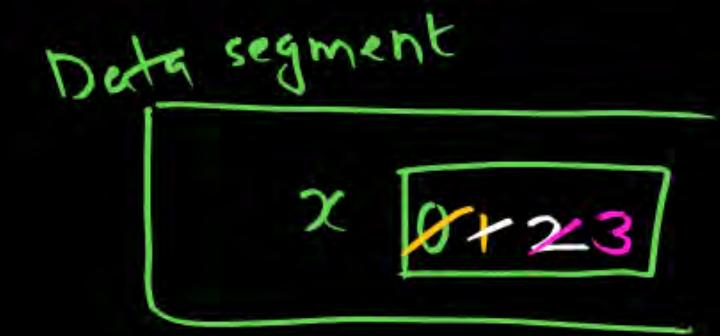
② register var. → CPU register/stack

③ static variable → (i) scope: block  
(ii) default: 0

(iii) lifetime: program

(iv) Memory: data section

- a) value persist b/w diff. calls.
- b) No redeclaration



1 2 3

void f(){

static int x = 0;

++x;

printf("%d", x);

}

void main(){

f();

f();

f();

}

Global var.

→ 0  
→ scope: multi file  
→ Data segment

H.W → videos → storage class Rx → speed  
[  
  Array & pointer  
  structure  
]

① DS → PYO's

SUCCESS

10:00 AM





## 2 mins Summary



Topic

C Programming

Topic

**THANK - YOU**



# CS & IT ENGINEERING



## PYQ SERIES

C Programming & Data  
Structures

Lecture No.- 03

By- Pankaj Sharma Sir



# Topics to be Covered

Topic

Data Structures





#Q. Suppose we are sorting an array of eight integers using heapsort, and we have just finished some heapify (either maxheapify or minheapify) operations. The array now looks like this 16 14 15 10 12 27 28. How many heapify operations have been performed on root of heap ?

2 times

A

1

C

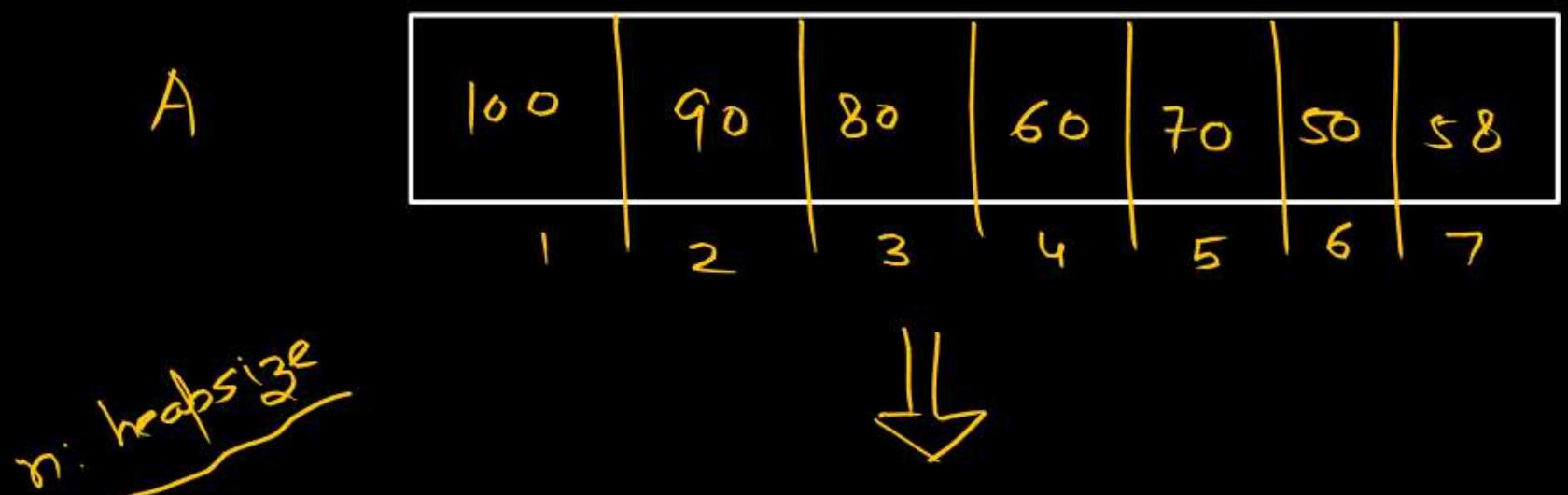
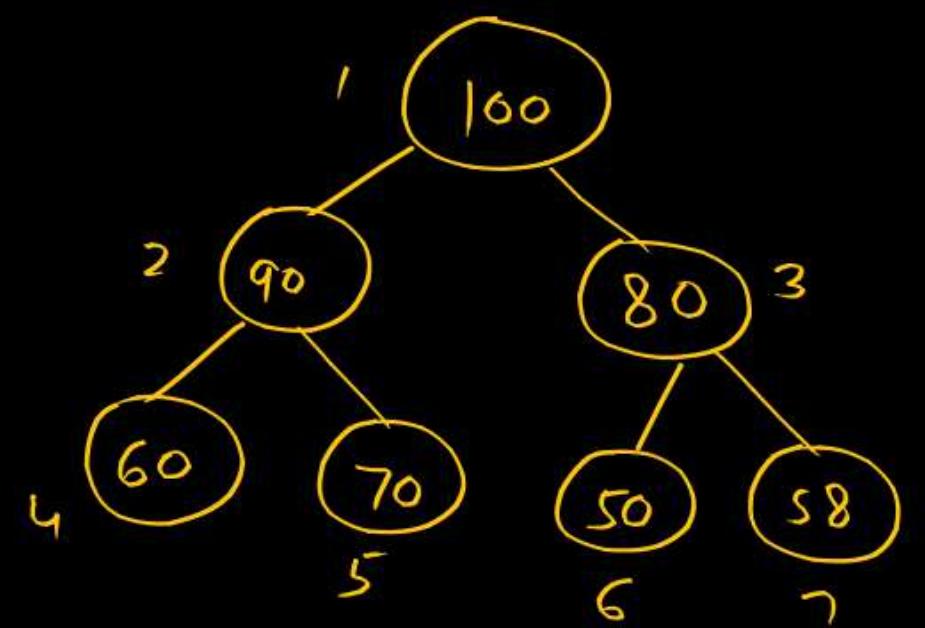
3 or 4

B

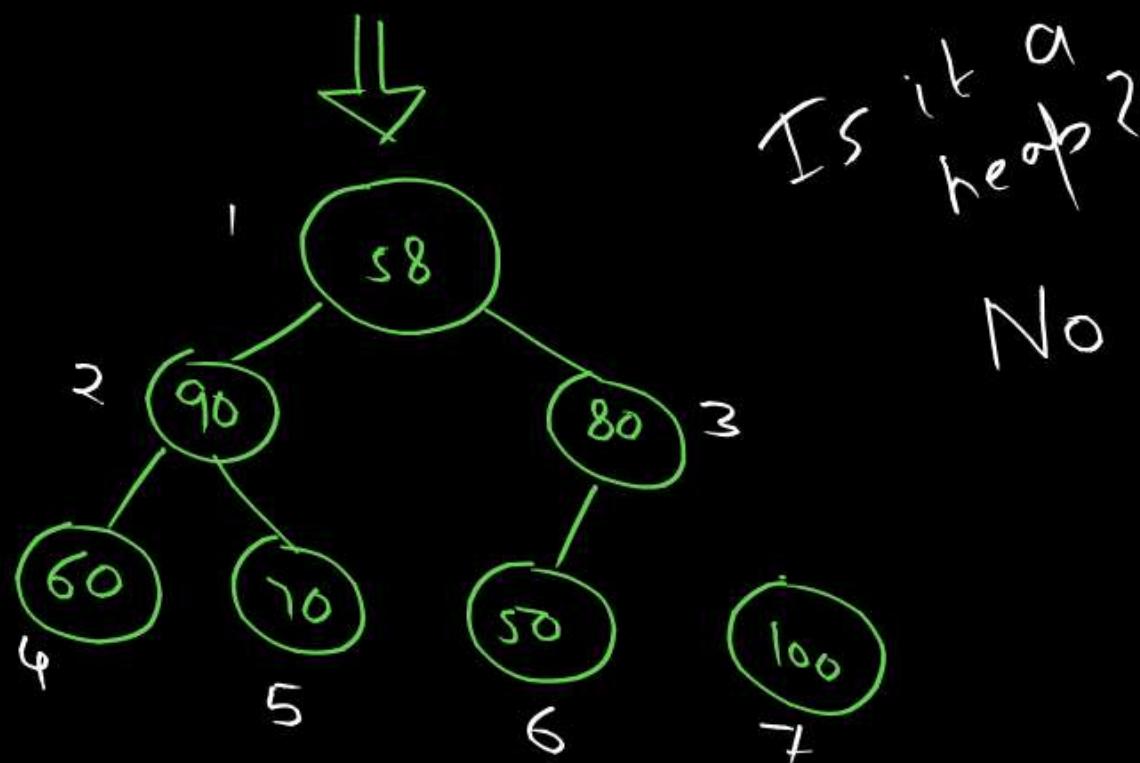
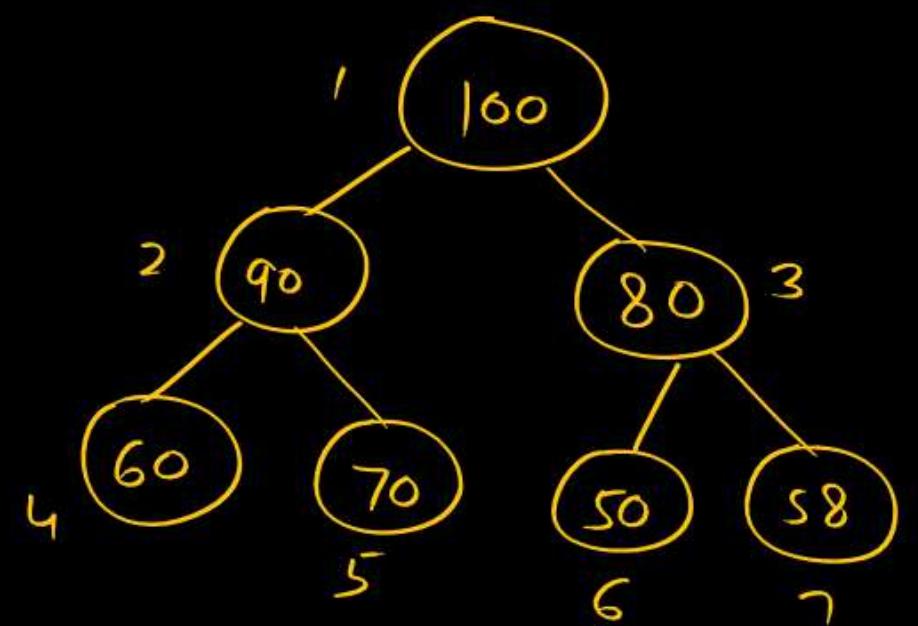
2

D

5 or 6



1.  $A[1] \leftrightarrow A[n] \Rightarrow A[1] \leftrightarrow A[7]$

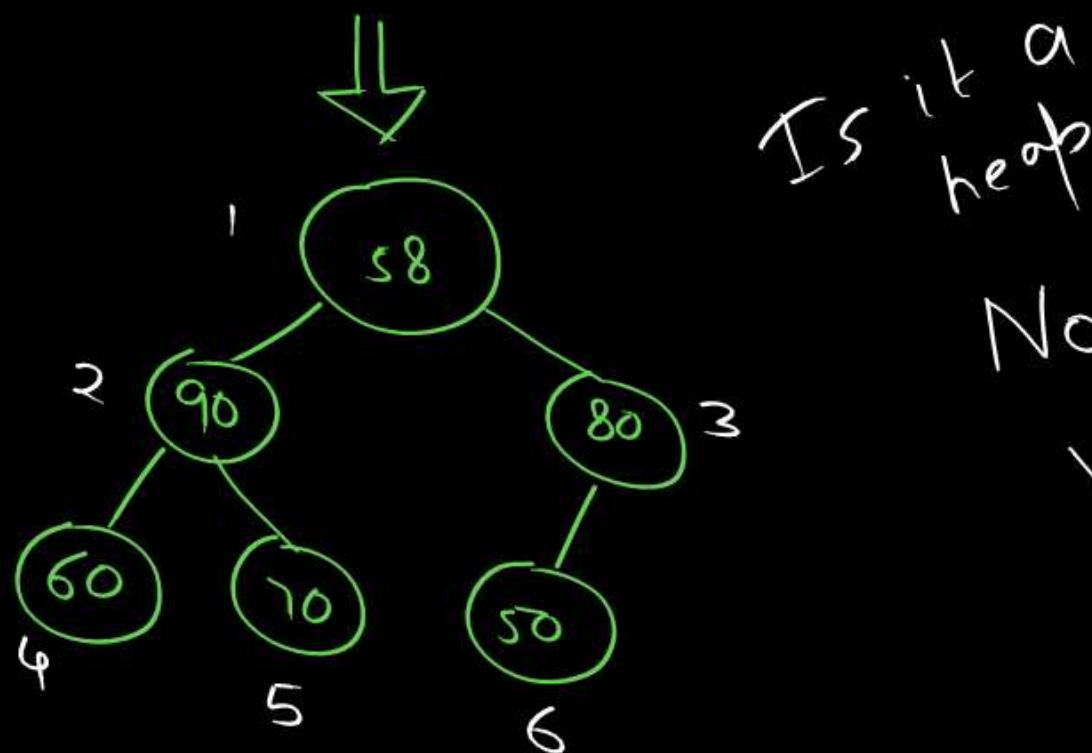
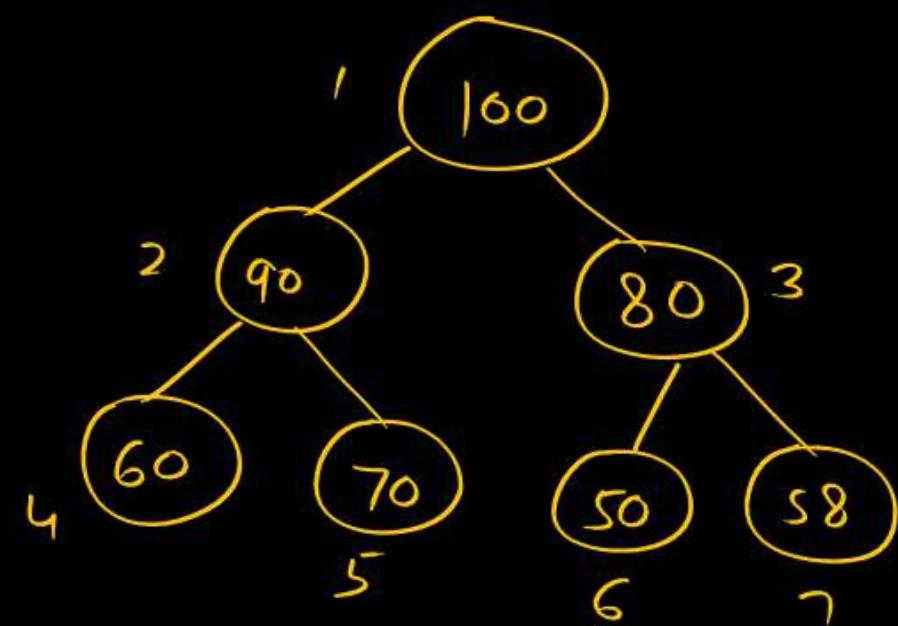


$n \cdot \text{heapsize}$

1.  $A[1] \leftrightarrow A[n] \Rightarrow A[1] \leftrightarrow A[7]$

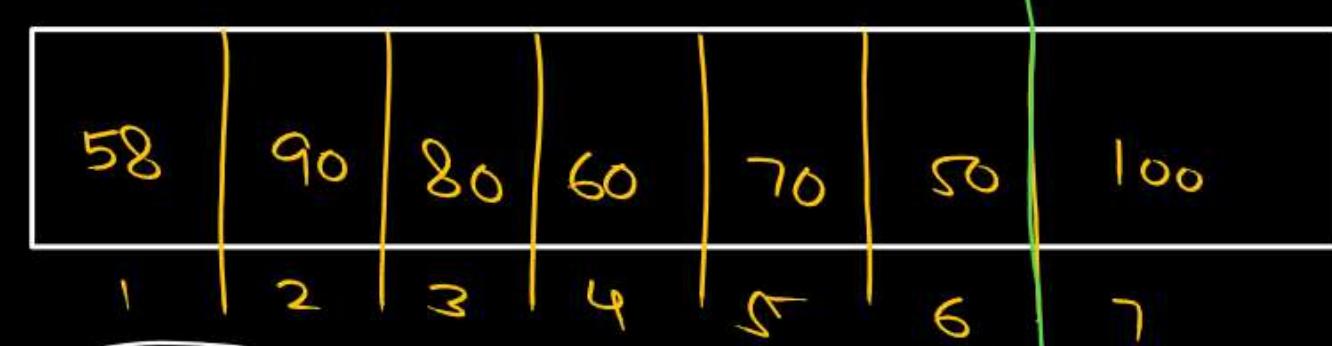


2)  $n = n-1$



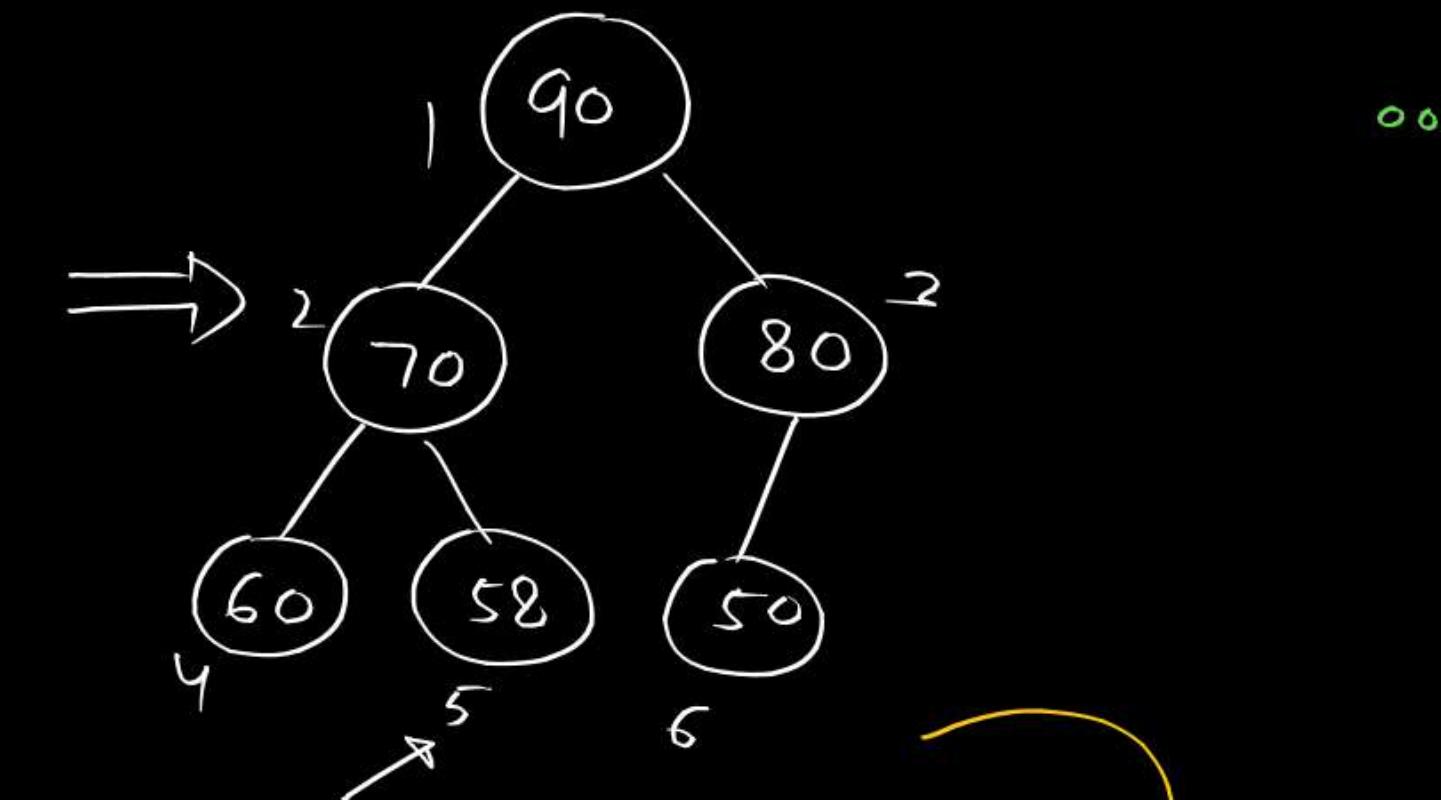
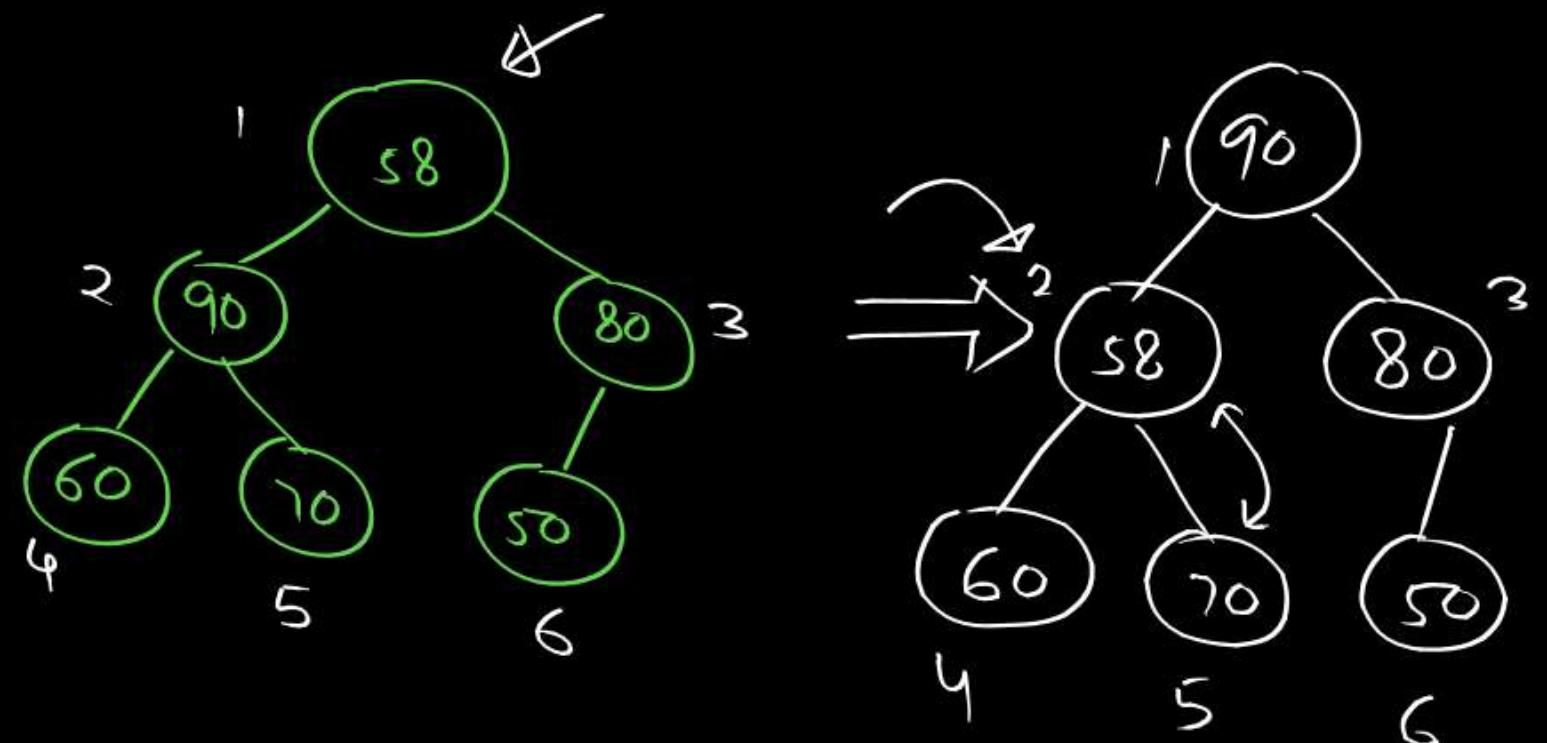
$n \cdot \text{heapsiz}$

1.  $A[1] \leftrightarrow A[n] \Rightarrow A[1] \leftrightarrow A[7]$



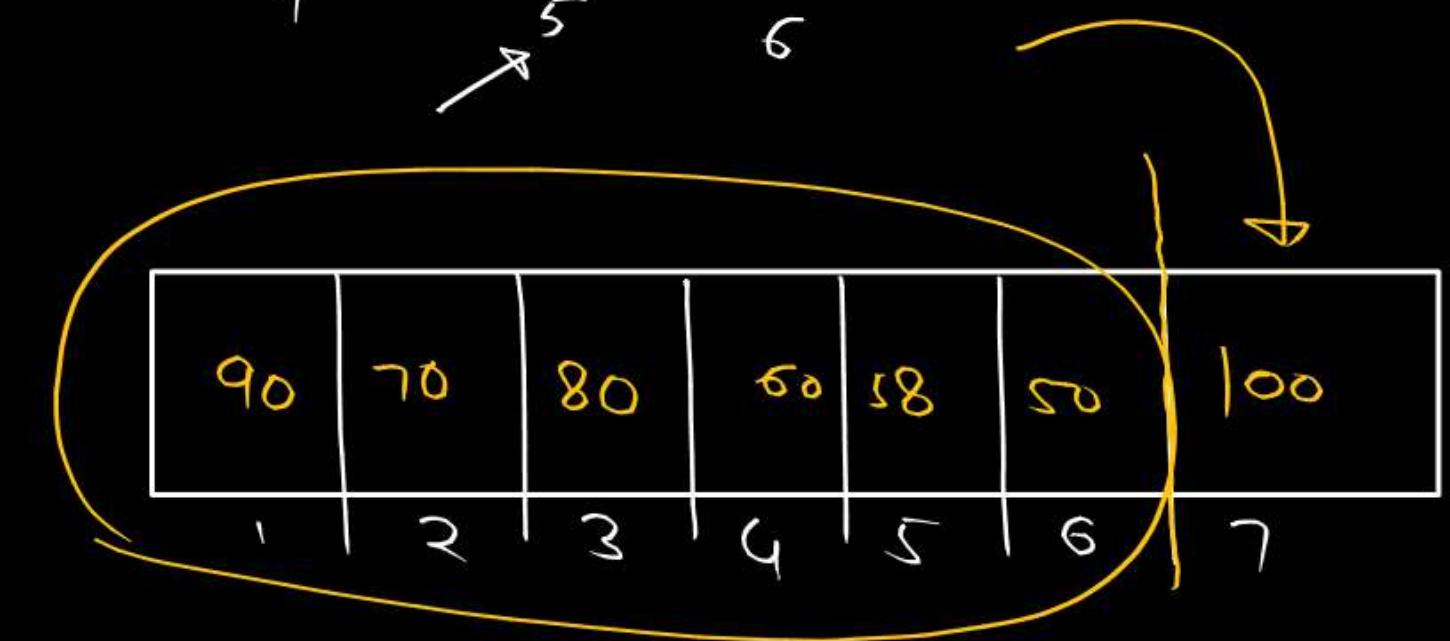
2.)  
3.)

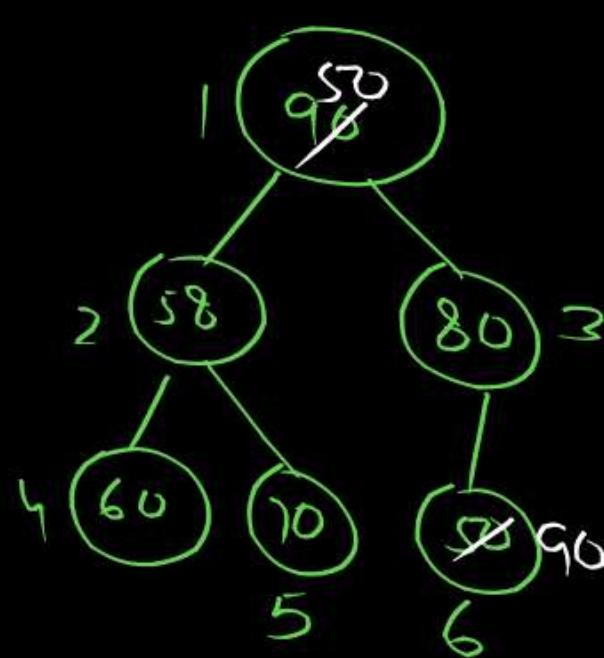
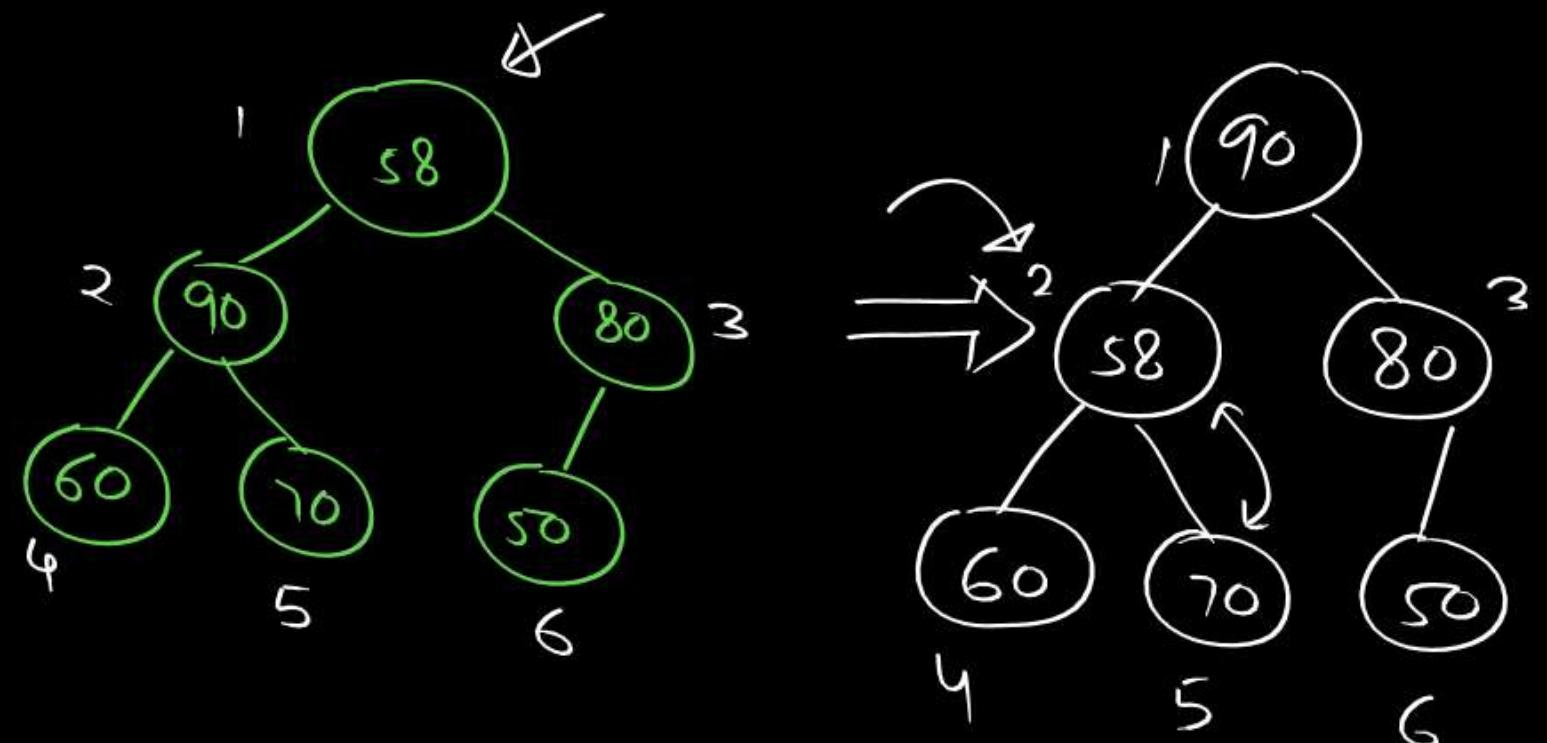
$n = n - 1$   
Heapify( $A, 1, n$ )



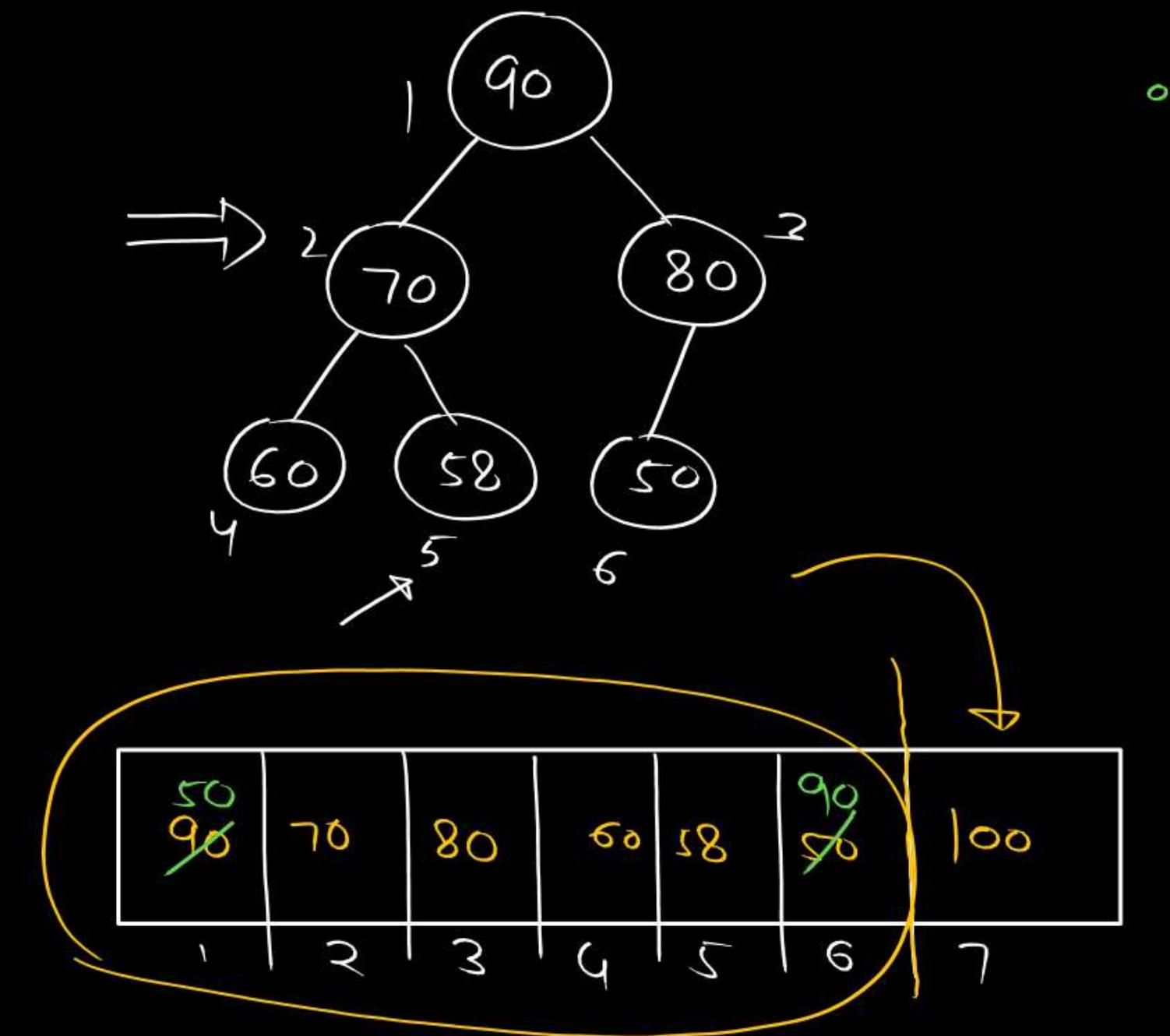
$n=6$

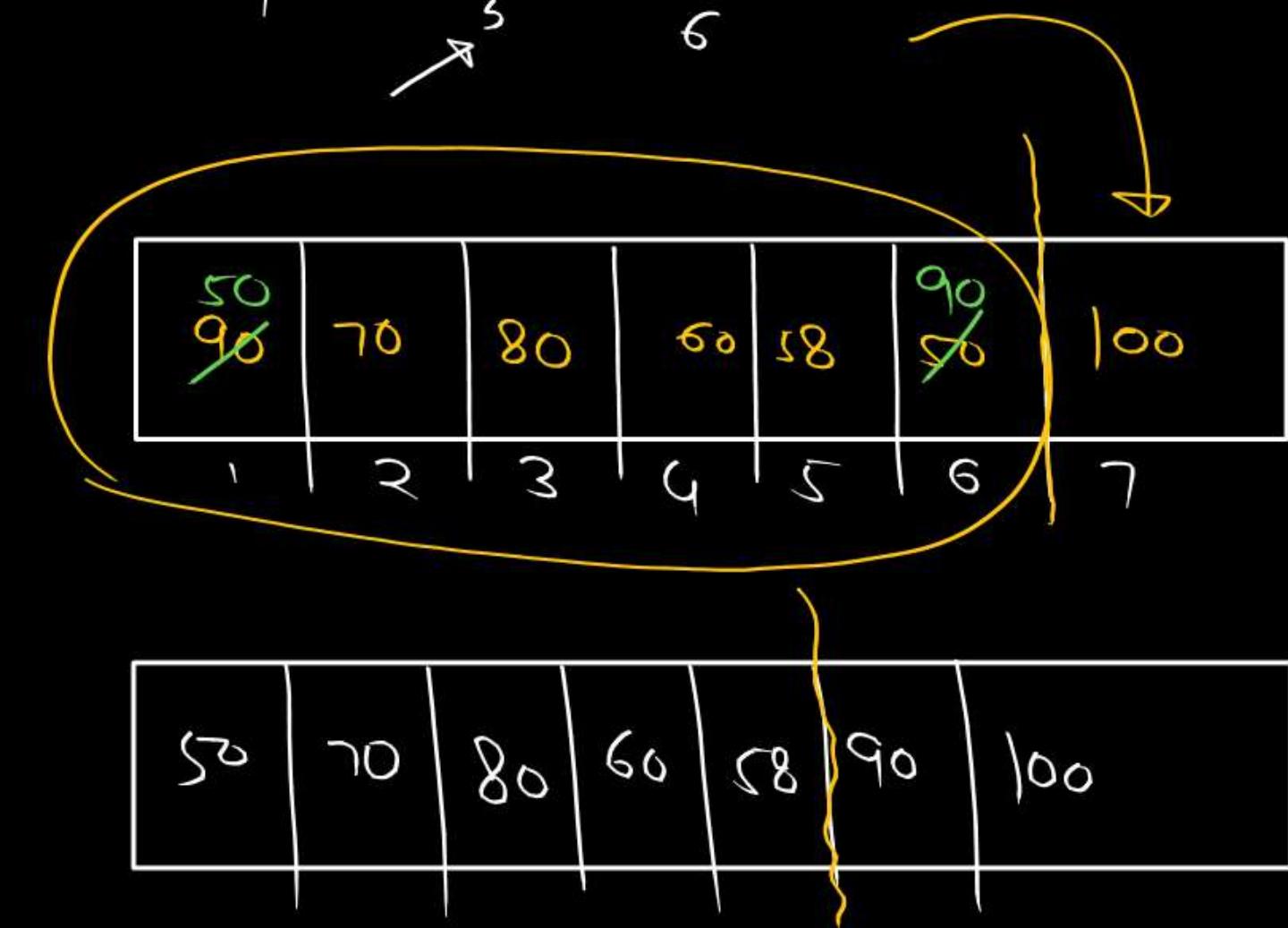
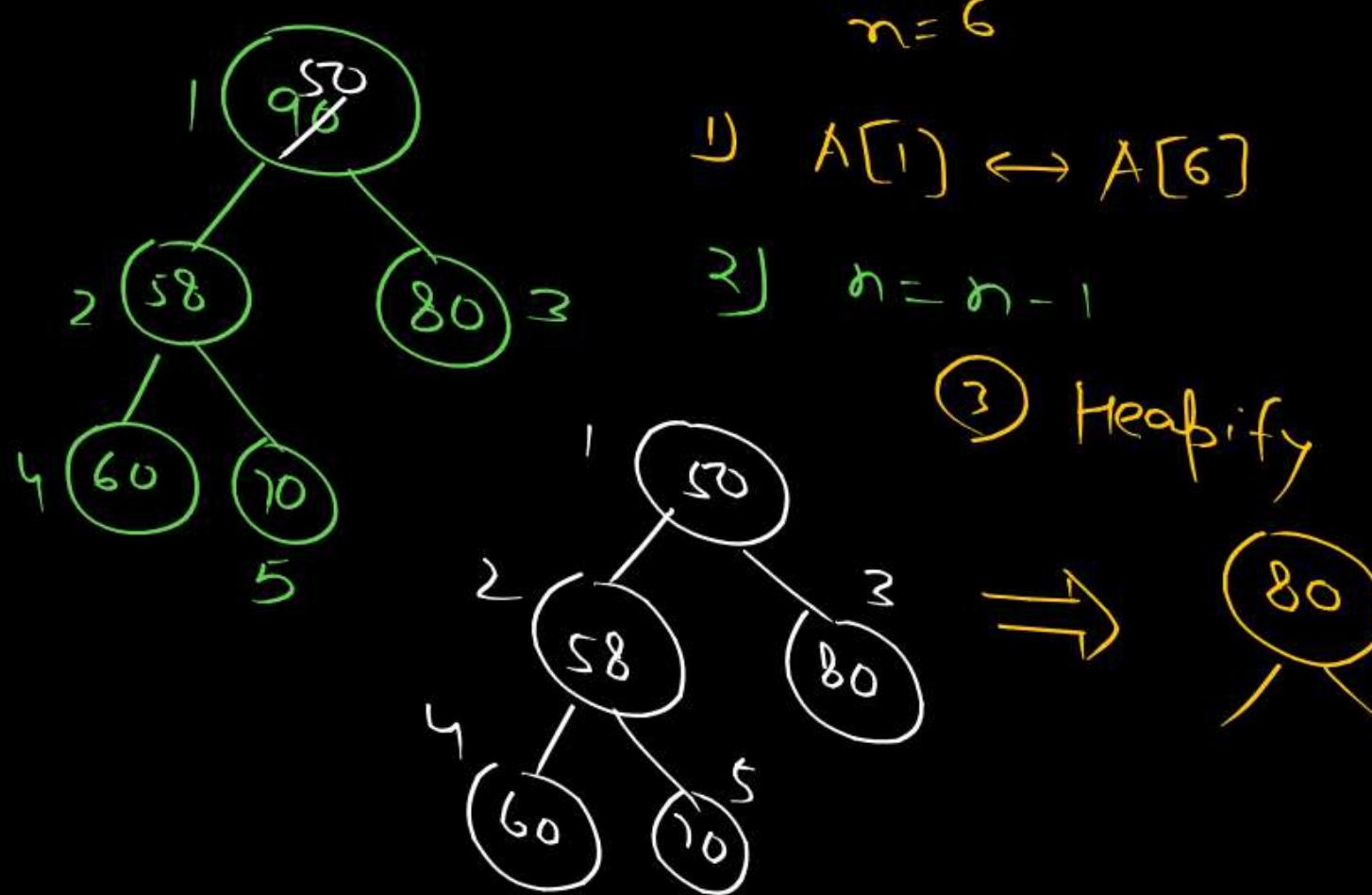
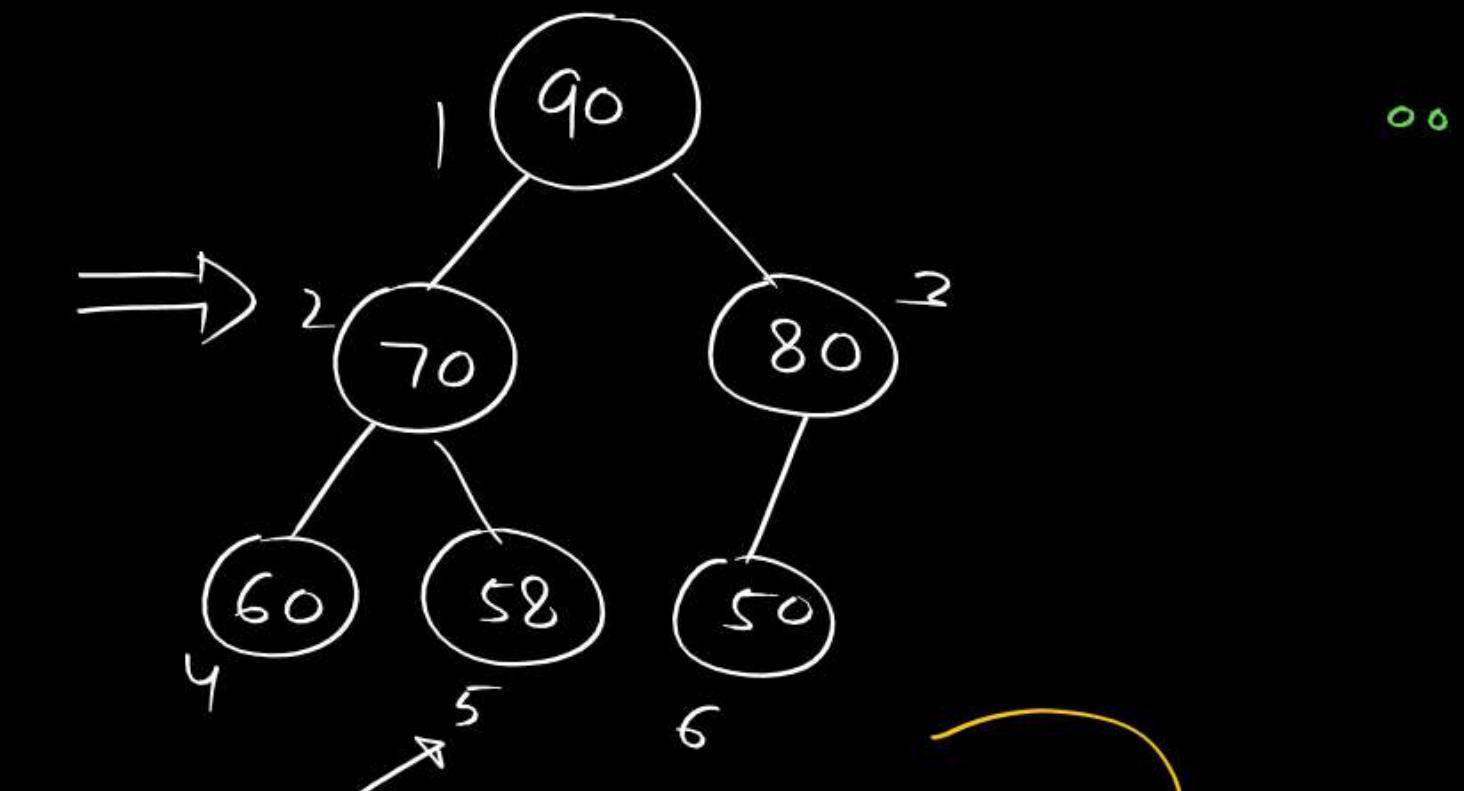
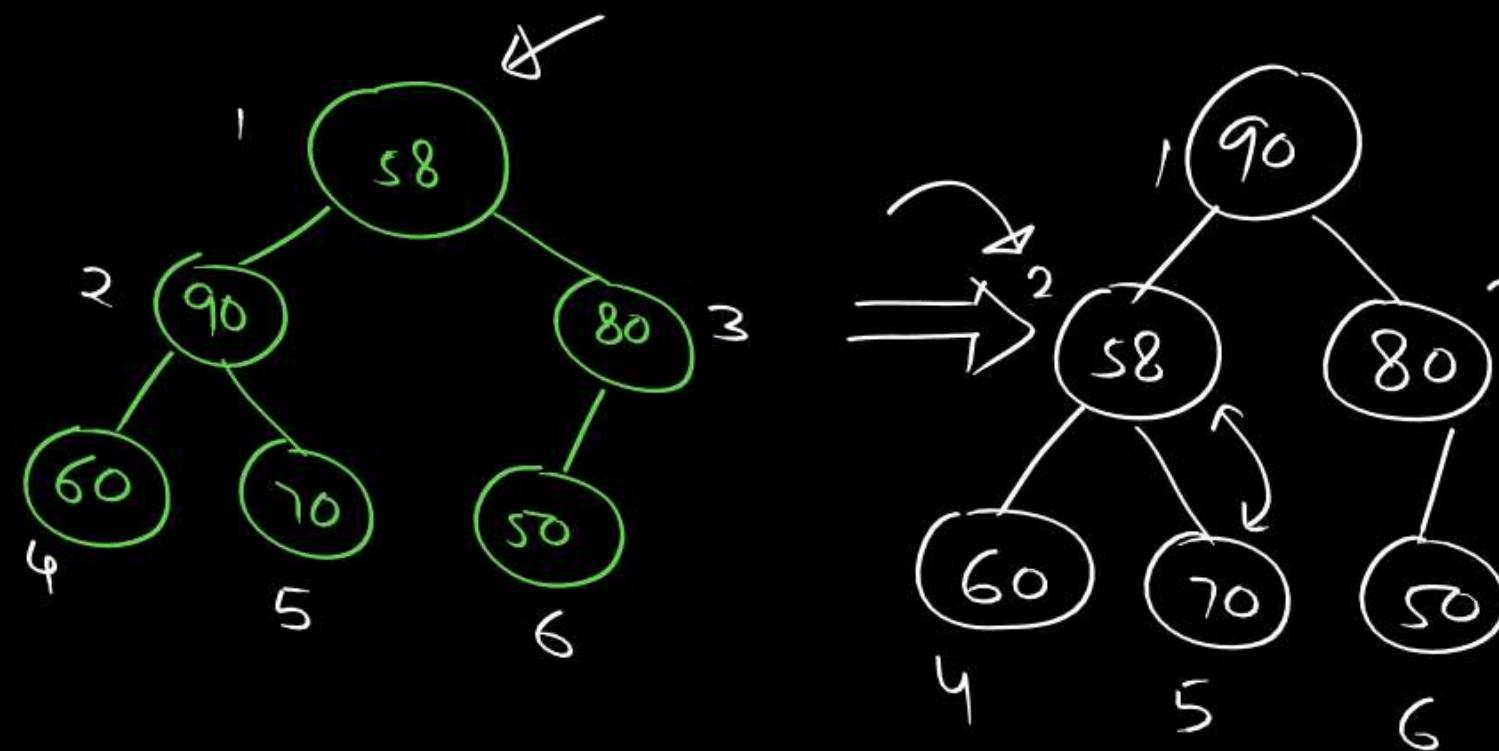
①  $A[1] \leftrightarrow A[6]$





$n=6$   
 ①  $A[1] \leftarrow A[6]$   
 ②  $n=n-1$







## Topic : 2002 : CS

ISRO PYQ SERIES

P  
W

#Q. A Young tableau is a 2D array of integers increasing from left to right and from top to bottom. Any unfilled entries are marked with  $\infty$ , and hence there cannot be any entry to the right of, or below a  $\infty$ . The following Young tableau consists of unique entries.



1	2	5	14
3	4	6	23
10	12	18	25
31	$\infty$	$\infty$	$\infty$

When an element is removed from a Young tableau, other elements should be moved into its place so that the resulting table is still a Young tableau (unfilled entries may be filled in with a  $\infty$ ). The minimum number of entries (other than 1) to be shifted, to remove 1 from the given Young tableau is \_\_\_\_\_

A 2

B 5

C 6

D 18



## Topic : 2002 : CS

ISRO PYQ SERIES

P  
W

#Q. A Young tableau is a 2D array of integers increasing from left to right and from top to bottom. Any unfilled entries are marked with  $\infty$ , and hence there cannot be any entry to the right of, or below a  $\infty$ . The following Young tableau consists of unique entries.

X	2	5	14
3	4	6	23
10	12	18	25
31	$\infty$	$\infty$	$\infty$

When an element is removed from a Young tableau, other elements should be moved into its place so that the resulting table is still a Young tableau (unfilled entries may be filled in with a  $\infty$ ). The minimum number of entries (other than 1) to be shifted, to remove 1 from the given Young tableau is \_\_\_\_\_

A 2

B 5

C 6

D 18



## Topic : 2002 : CS

ISRO PYQ SERIES

P  
W

#Q. A Young tableau is a 2D array of integers increasing from left to right and from top to bottom. Any unfilled entries are marked with  $\infty$ , and hence there cannot be any entry to the right of, or below a  $\infty$ . The following Young tableau consists of unique entries.

2  
 $\swarrow$   
5 entries

<del>2</del>	<del>4</del>	5	14
3	<del>4</del>	<del>6</del> $\uparrow$	<del>18</del>
10	12	<del>18</del> $\uparrow$ <del>25</del>	<del>25</del> $\infty$
31	$\infty$	$\infty$	$\infty$

When an element is removed from a Young tableau, other elements should be moved into its place so that the resulting table is still a Young tableau (unfilled entries may be filled in with a  $\infty$ ). The minimum number of entries (other than 1) to be shifted, to remove 1 from the given Young tableau is \_\_\_\_\_

A 2

C 6

B 5

D 18



## Topic : 2022: CS

#Q. A single array  $A[1..MAXSIZE]$  is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables  $\text{top1}$  and  $\text{top2}$  ( $\text{top1} < \text{top2}$ ) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for “stack full” is

**A**

$(\text{top1} = \text{MAXSIZE}/2)$  and  $(\text{top2} = \text{MAXSIZE}/2+1)$

**B**

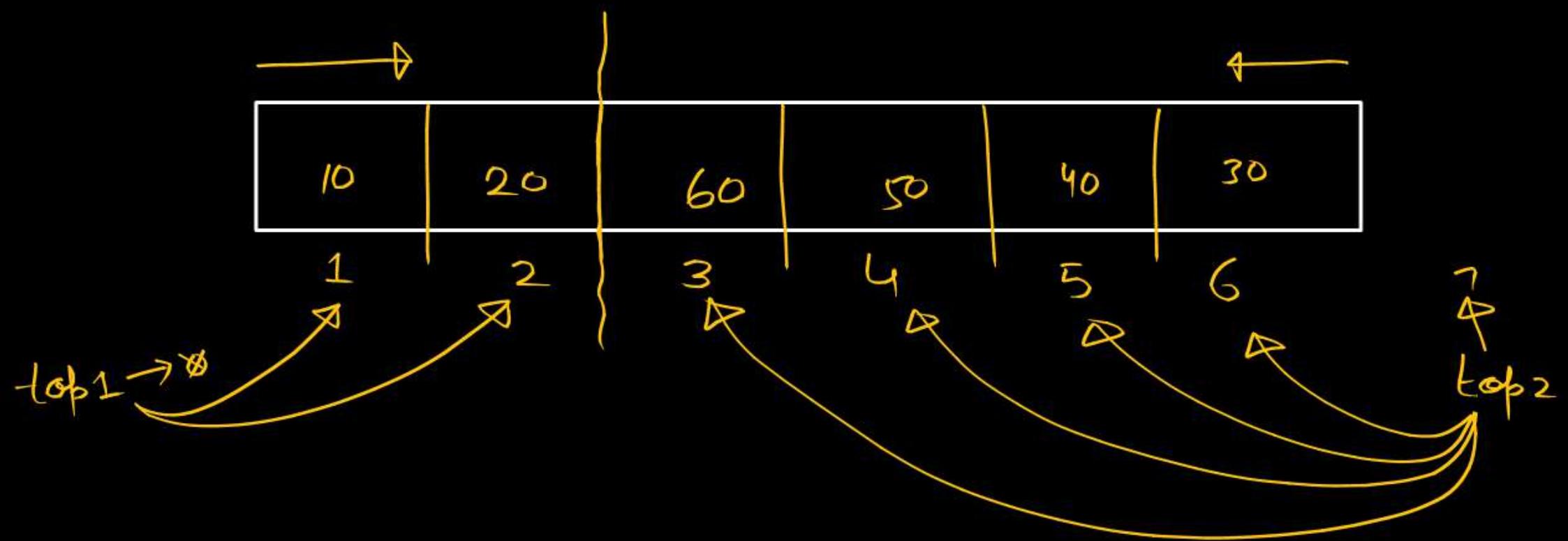
$\text{top1} + \text{top2} = \text{MAXSIZE}$

**C**

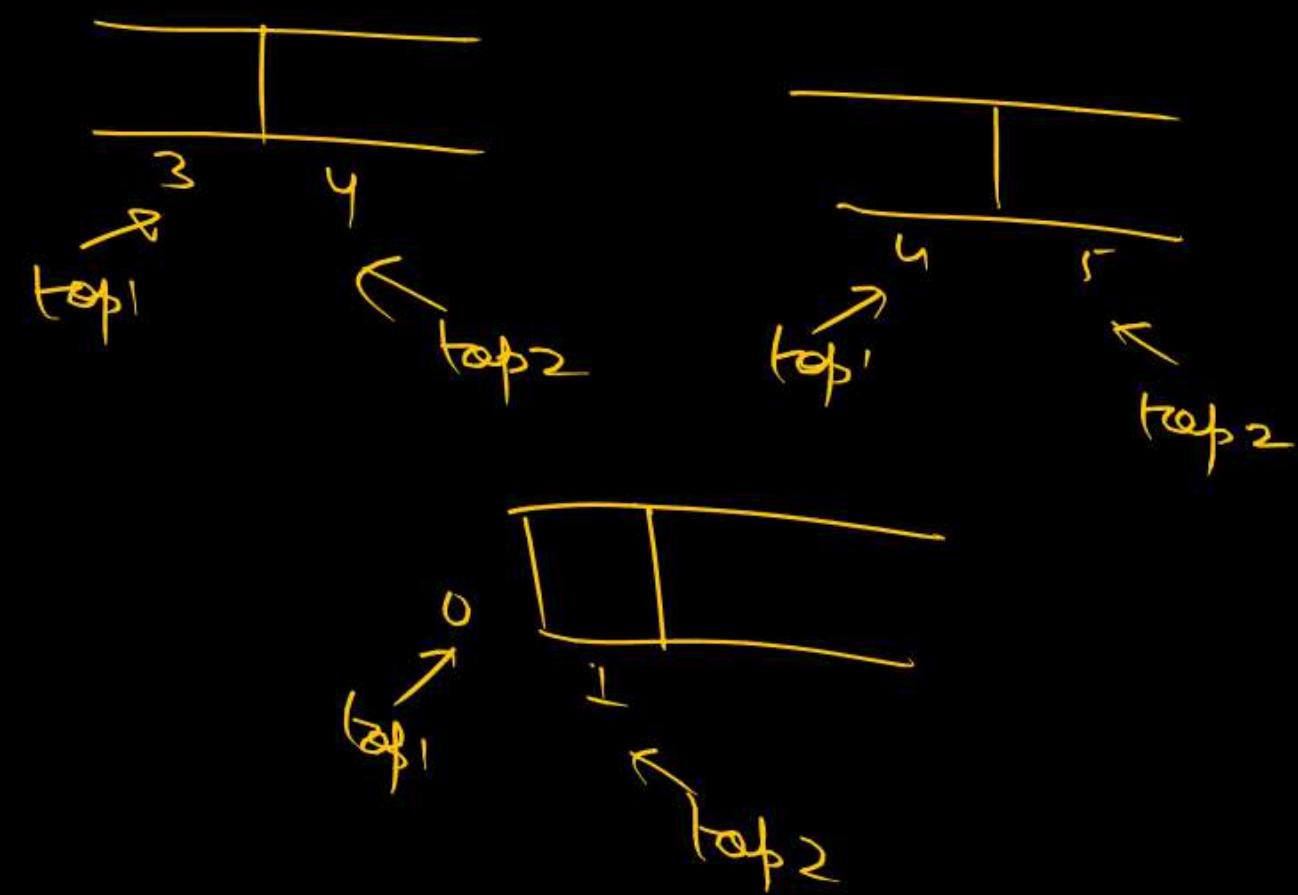
$(\text{top1} = \text{MAXSIZE}/2)$  or  $(\text{top2} = \text{MAXSIZE})$

**D**

$\text{top1} = \text{top2} - 1$  ✓



$$\text{top2} = \text{top1} + 1$$





#Q. Let T be a binary search tree with 15 nodes. The minimum and maximum possible heights of T are:  
The height of a tree with a single node is 0.

- A** 4 and 15 respectively
- C** 4 and 14 respectively

- B** 3 and 14 respectively
- D** 3 and 15 respectively



3 and 15 respectively



$$n_{\min} = h+1 \Rightarrow 15 = h+1 \Rightarrow \boxed{h=14}$$

$$n_{\max} = 2^{h+1}-1 \Rightarrow 15 = 2^{h+1}-1 \Rightarrow 16 = 2^{h+1} \Rightarrow h+1 = 4 \Rightarrow \boxed{h=3}$$



#Q. Consider a max heap, represented by the array: 40, 30, 20, 10, 15, 16, 17, 8, 4. Now consider that a value 35 is inserted into this heap. After insertion, the new heap is

Array index	1	2	3	4	5	6	7	8	9
Value	40	30	20	10	15	16	17	8	4

Now consider that a value 35 is inserted into this heap . After insertion , the new heap is

A

40, 30, 20, 10, 15, 16, 17, 8, 4, 35

C

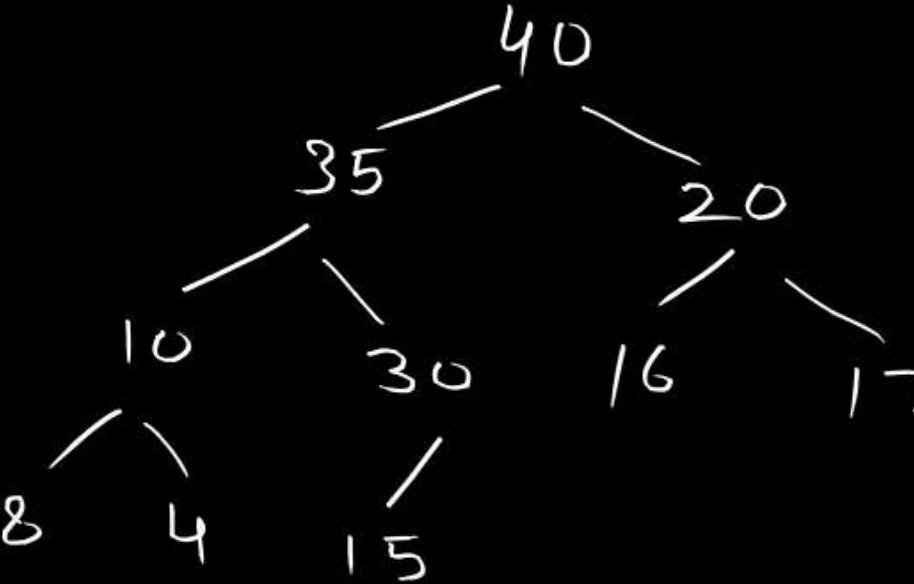
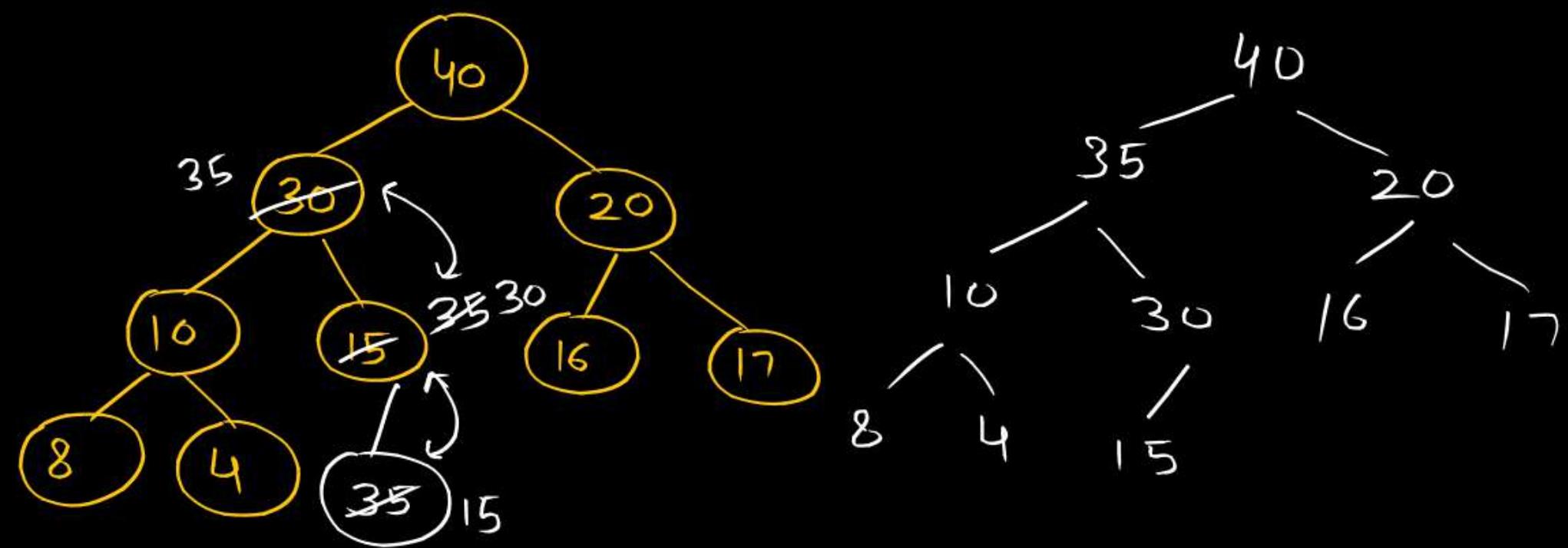
40, 30, 20, 10, 35, 16, 17, 8, 4, 15

B

40, 35, 20, 10, 30, 16, 17, 8, 4, 15

D

40, 35, 20, 10, 15, 16, 17, 8, 4, 30





#Q. A queue is implemented using a non-circular singly linked list. The queue has a head pointer and a tail pointer, as shown in the figure. Let  $n$  denote the number of nodes in the queue. Let 'enqueue' be implemented by inserting a new node at the head, and 'dequeue' be implemented by deletion of a node from the tail.

Which one of the following is the time complexity of the most time-efficient implementation of 'enqueue' and 'dequeue', respectively, for this data structure?



- A**  $\Theta(1), \Theta(1)$
- B**  ~~$\Theta(1), \Theta(n)$~~
- C**  $\Theta(n), \Theta(1)$
- D**  $\Theta(n), \Theta(n)$

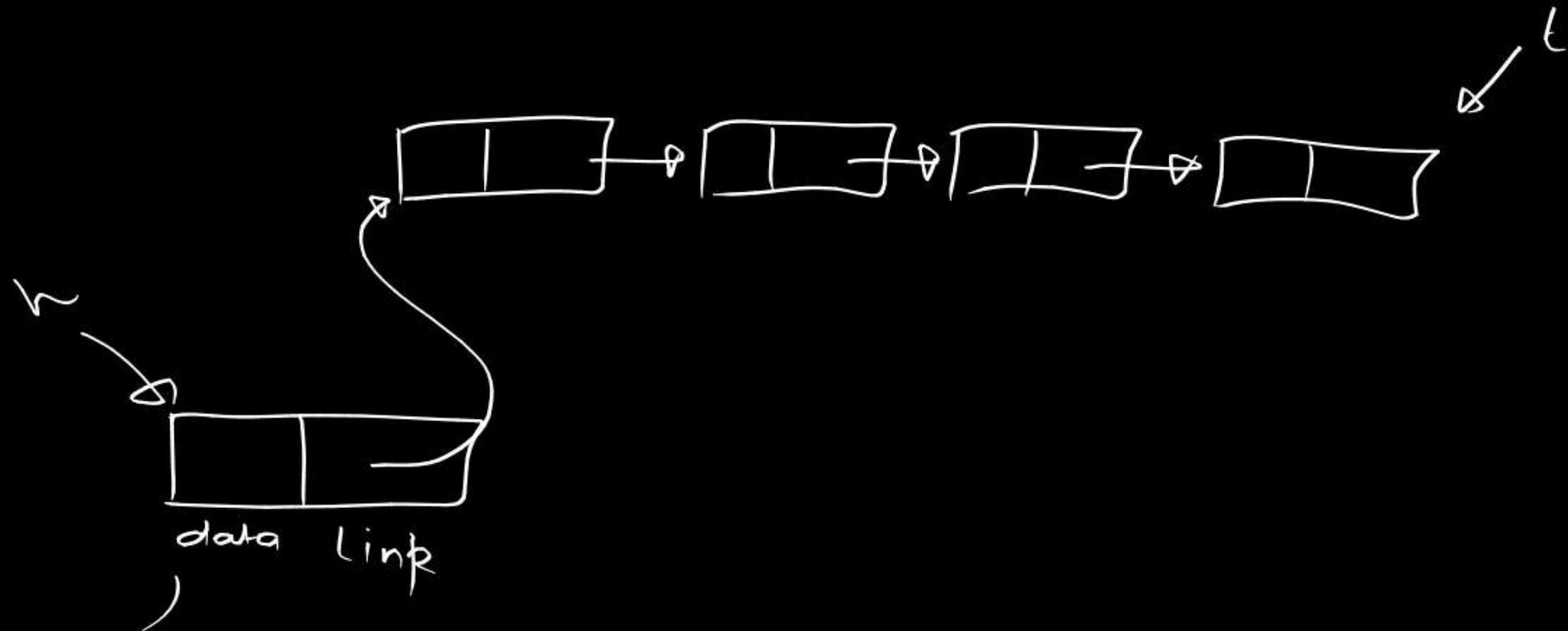
$O(1)$

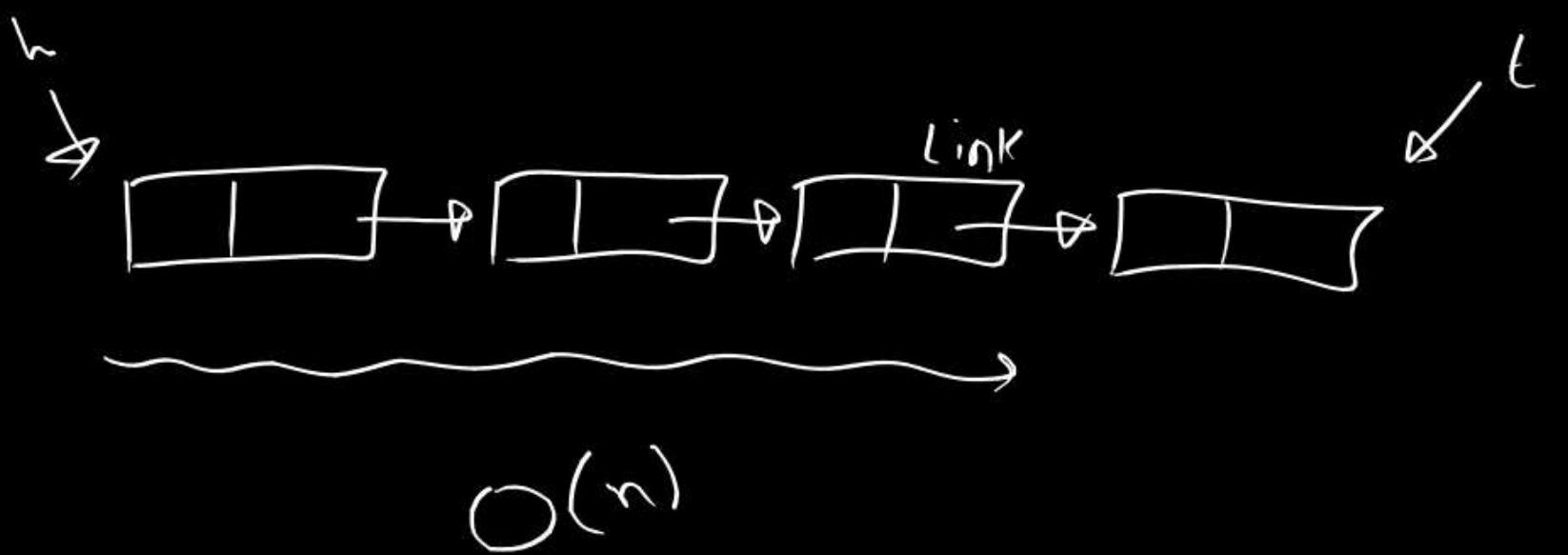
{  
temp = malloc(

temp → data = n

temp → link = h

h = temp







## Topic : 2017: CS

ISRO PYQ SERIES

P  
W

#Q. The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is:

$$n_{\max} = 2^{h+1} - 1$$

- A  $2^h - 1$
- B  $2^{h-1} - 1$
- C  $2^{h+1} - 1$
- D  $2^{h+1}$



## Topic : 2017: CS

ISRO PYQ SERIES

P  
W

#Q. M is a square matrix of order ‘n’ and its determinant value is 5. If all the elements of M are multiplied by 2, its determinant value becomes 40. The value of ‘n’ is

- A** 2
- B** 3
- C** 5
- D** 4

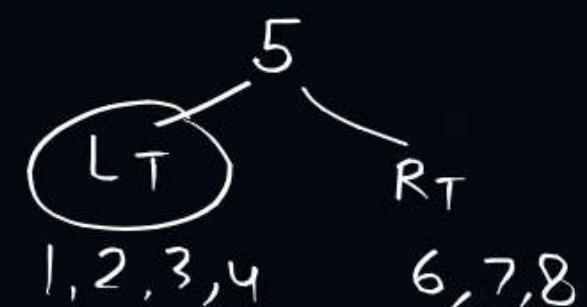


#Q. A binary search tree contains the values 1,2,3,4,5,6,7 and 8. The tree is traversed in preorder and the values are printed out. Which of the following sequences is a valid output?

Root L<sub>T</sub> R<sub>T</sub>

- A** 5 3 1 2 4 7 8 6
- C** 5 3 2 4 1 6 7 8

- B** 5 3 1 2 6 4 8 7 X
- D** 5 3 1 2 4 7 6 8





## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. A binary search tree contains the values 1,2,3,4,5,6,7 and 8. The tree is traversed in preorder and the values are printed out. Which of the following sequences is a valid output?

A

5 3 1 2 4 7 8 6

C

5 3 2 4 1 6 7 8

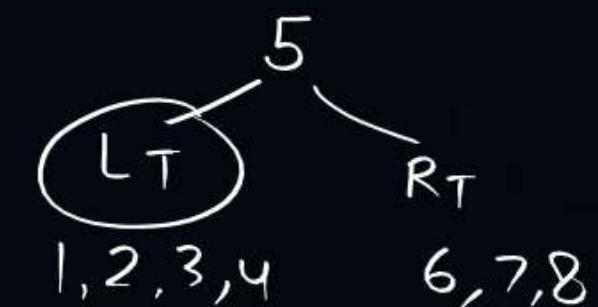
Root L<sub>T</sub> R<sub>T</sub>

B

5 3 1 2 6 4 8 7

D

5 3 1 2 4 7 6 8



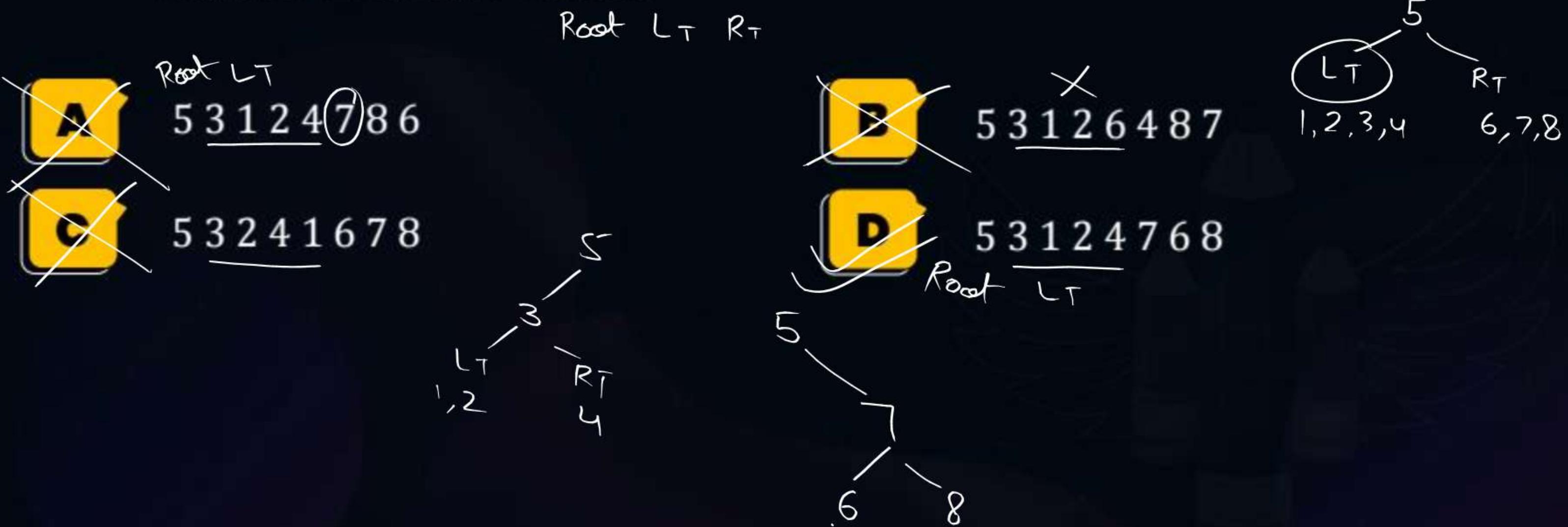


## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. A binary search tree contains the values 1,2,3,4,5,6,7 and 8. The tree is traversed in preorder and the values are printed out. Which of the following sequences is a valid output?





#Q. What does the following function do for a given Linked List with first node as head?

```
void fun1 (struct node* head)
{
    if(head==NULL)
        return;
    fun1(head->next);
    printf("%d", head->data);
}.
```

A

Prints all nodes of linked lists

C

Prints alternate nodes of Linked List

B

Prints all nodes of linked list in reverse order

D

Prints alternate nodes in reverse order



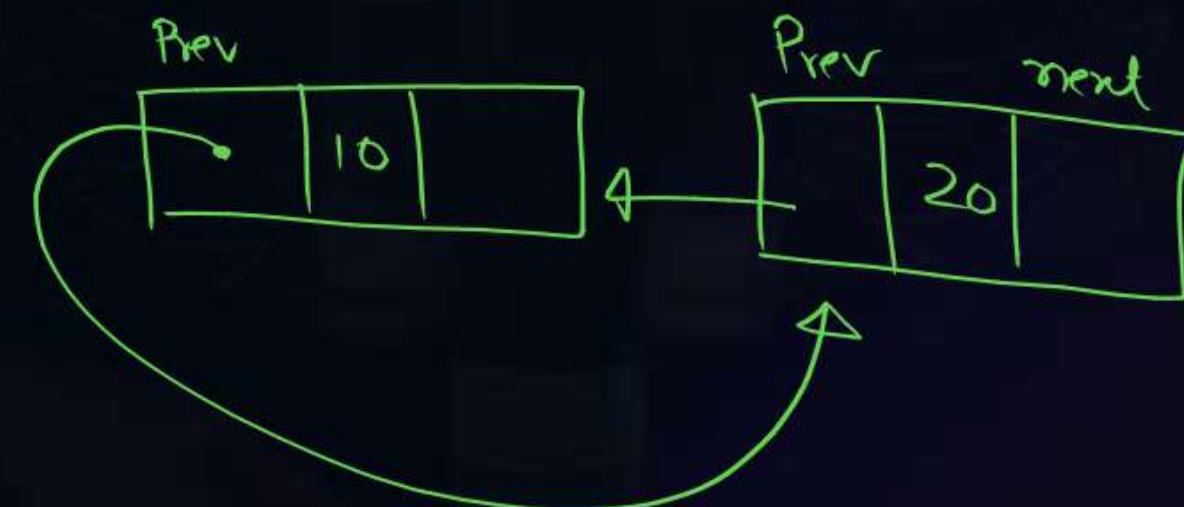
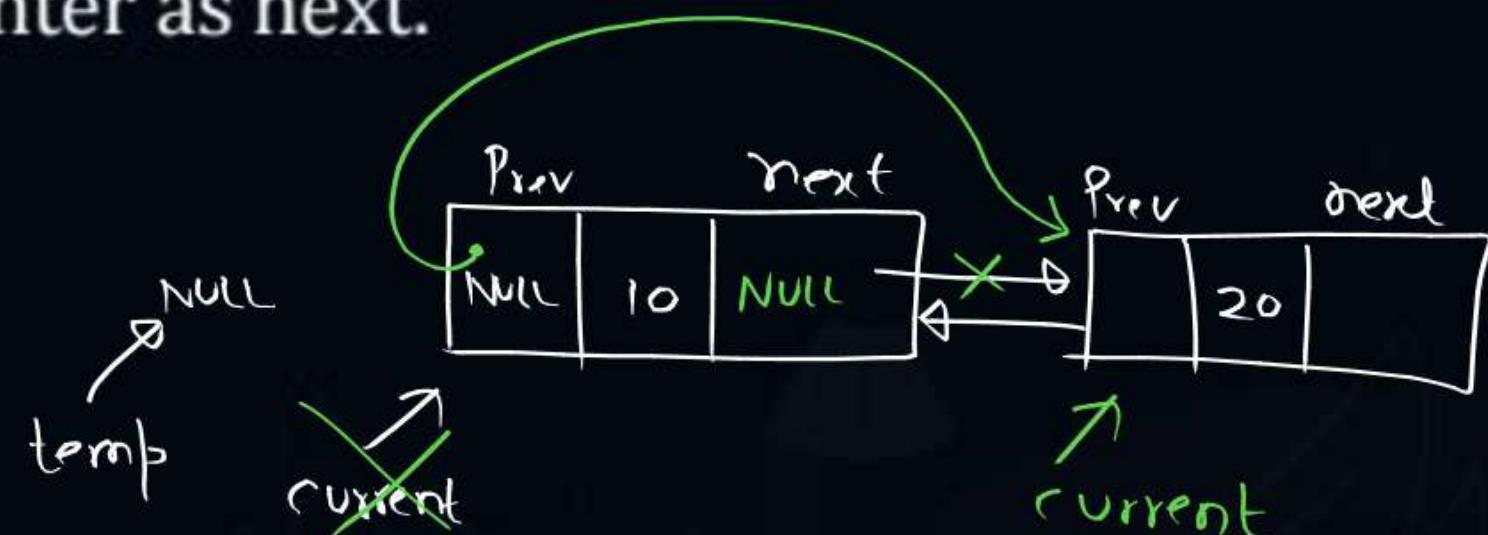
## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Consider the following function that takes reference to head of a Doubly Linked List as parameter. Assume that a node of doubly linked list has previous pointer as prev and next pointer as next.

```
void fun(struct node **head_ref)
{
    struct node *temp = NULL;
    struct node *current = *head_ref;
    while (current != NULL){
        temp = current->prev;
        current->prev = current->next;
        current->next = temp;
        current = current->prev;
    }
}
```



```
if(temp != NULL )  
*head_ref = temp->prev;  
}
```

Assume that reference of head of following doubly linked list is passed to above function  $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5 \leftrightarrow 6$ . What should be the modified linked list after the function call?

A

$2 \leftrightarrow 1 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 6 \leftrightarrow 5$

C

$6 \leftrightarrow 5 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 2 \leftrightarrow 1$



B

$5 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 2 \leftrightarrow 1 \leftrightarrow 6$

D

$6 \leftrightarrow 5 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 1 \leftrightarrow 2$



## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. Assume that the operators  $+$ ,  $-$ ,  $\times$  are left associative and  $\wedge$  is right associative. The order of precedence (from highest to lowest) is  $\wedge, \times, +, -$ . The postfix expression corresponding to the infix expression

$a + b \times c - d \wedge e \wedge f$  is

A

$a + b \times c - d \wedge e \wedge f$

C

$a + b \times c - d \wedge e \wedge f$

B

$a b c \times + d e \wedge f \wedge -$

D

$- + a b c \times + d e \wedge f \wedge -$

$a + [b c \times] - [d e f \wedge \wedge]$

$[a b c \times +] - [d e f \wedge \wedge]$



## Topic : 2017 : CS

ISRO PYQ SERIES

P  
W

#Q. A **balance** factor in AVL tree to check

- A** What rotation to make
- B** If all child nodes are at same level
- C** when the last rotation occurred
- D** if the tree is unbalanced.



#Q. A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is : 10, 8, 5, 3, 2. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is

A

10, 8, 7, 3, 2, 1, 5

C

10, 8, 7, 1, 2, 3, 5

B

10, 8, 7, 2, 3, 1, 5

D

10, 8, 7, 5, 3, 2, 1





#Q. In a complete k-ary tree, every internal node has exactly k children. The number of leaves in such a tree with n internal nodes is

A nk

$$\text{Total} = nk + 1$$

C  $n(k - 1) + 1$

$$L + n = nk + 1$$

$$L = nk - n + 1$$

$$\boxed{L = n(k - 1) + 1}$$

B  $(n - 1)k + 1$

D  $n(k - 1)$





#Q. Which of the following need not be a binary tree?

- A** Search tree
- C** AVL tree

- B** Heap
- D** B tree



## Topic : 2016 : CS

ISRO PYQ SERIES

P  
W

#Q. The maximum number of nodes in a binary tree of level  $k$ ,  $k \geq 1$  is :

- A  $2^k + 1$
- C  $2^k - 1$

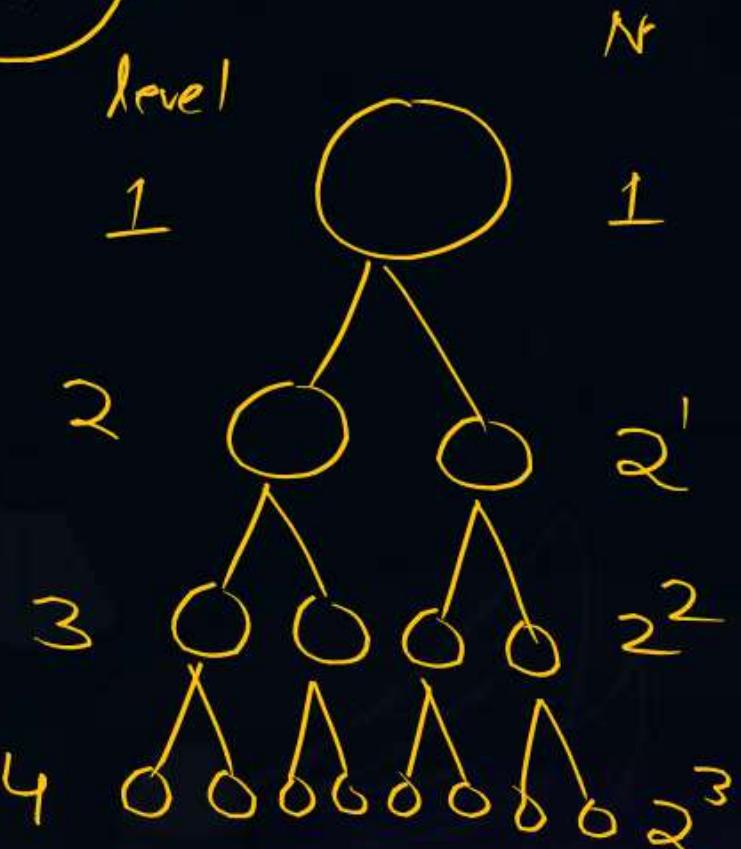
- B  $2^{k-1}$
- D  $2^{k-1} - 1$

$$N = 1 + 2^1 + 2^2 + 2^3$$

$N = 2^4 - 1$

$$2^k - 1$$

4 level  
 $k$  level

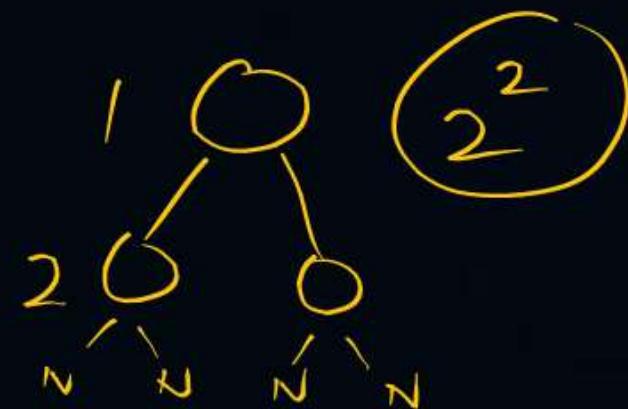
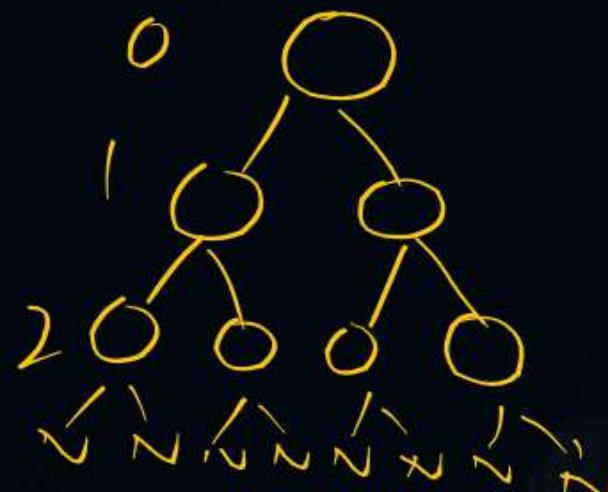




#Q. The number of unused pointers in a complete binary tree of depth 5 is :

- A 4
- B 16
- C 8
- D 32

$$\begin{array}{l} \text{depth} = k \\ \downarrow \\ 2^k \rightarrow 2^{k+1} \\ \text{---} \\ 2^5 \quad 2^6 \\ \text{---} \\ 2^3 \end{array}$$





#Q. A \_\_\_\_\_ is a linear list in which insertions and deletions are made to from either end of the structure.

- A** Circular queue
- B** Priority queue
- C** Stack
- D** Dequeue ✓



## Topic : 2016 : CS

ISRO PYQ SERIES

P  
W

#Q. Which of the following is the correct order of evaluation for the below expression ?

$$z = x + \underline{y * z} / \underline{4 \%} \underline{2 - 1}$$

A      \* / % + - =      ✓

B      = \* / % + -

C      / \* % - + =

D      \* % / - + =



## 2 mins Summary

ISRO PYQ SERIES

P  
W

**Topic One**

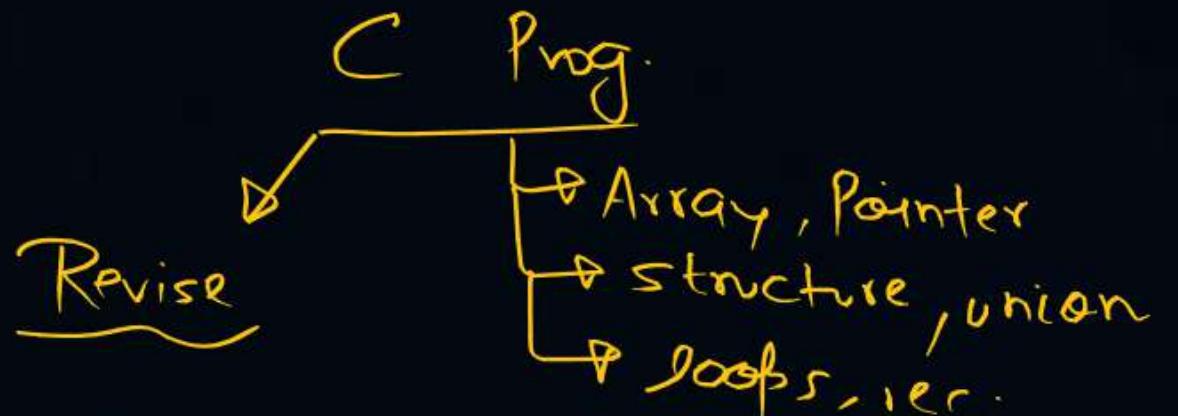
**Topic Two**

**Topic Three**

**Topic Four**

**Topic Five**

# THANK - YOU



DS

```
graph LR; DS[DS] --> DS_TL["tree, LL, (Heap)"]
```

Evening

```
graph LR; Evening["Evening"] --> Infix["* Infix, Prefix, Postfix"]; Evening --> DS["* DS"]
```

→ Complete



# CS & IT ENGINEERING



## PYQ SERIES

C Programming and Data Structure

Lecture No.- 04

By- Pankaj Sharma Sir



# Recap of Previous Lecture



Topic

Programming & Data Structures Practice Problems

DS problems  
90%  $\Rightarrow$  Gate PYQs

# Topics to be Covered



Topic

Programming & Data Structures

Revision



# Programming & Data Structures

ISRO PYQ SERIES

P  
W

① Arrays

unsorted array  
Insert at end  $\rightarrow O(1)$

A

$A[n] = x; n++;$

10	20	30	40				
0	1	2	3	4	5	6	7

$n=4$   
 $\Rightarrow O(n)$   
particular position:  
{ Insert 50  
at index 2 }

10	20	30	40	50			
0	1	2	3	4	5	6	7

10	20		30	40	50		- - -	
0	1	2	3	4	5			

To find the position - BS  $\rightarrow O(\log n)$  (No use)

Sorted Array - Insert 8

$O(n)$

10		20		30		40		50		60				
0	1	2	3	4	5	6	7	8	9					



## Deletion

at end  $\rightarrow O(1)$



10	20	30	40	50				
0	1	2	3	4	5	6	7	8

At particular position

 $\hookrightarrow O(n)$ 

10	20	30	40	50				
0	1	2	3	4	5	6	7	8

## Array disadvantages

- ①
- ②

Insertion/Deletion are expensive  
pre allocate the size

## Adv.

- ① search, sorting performance  $\rightarrow$  Cache friendly

③ Merge: Given 2 sorted array, merge them into sorted array.

$$m = 5$$

A

1	3	7	8	10
---	---	---	---	----

$$O(m+n)$$

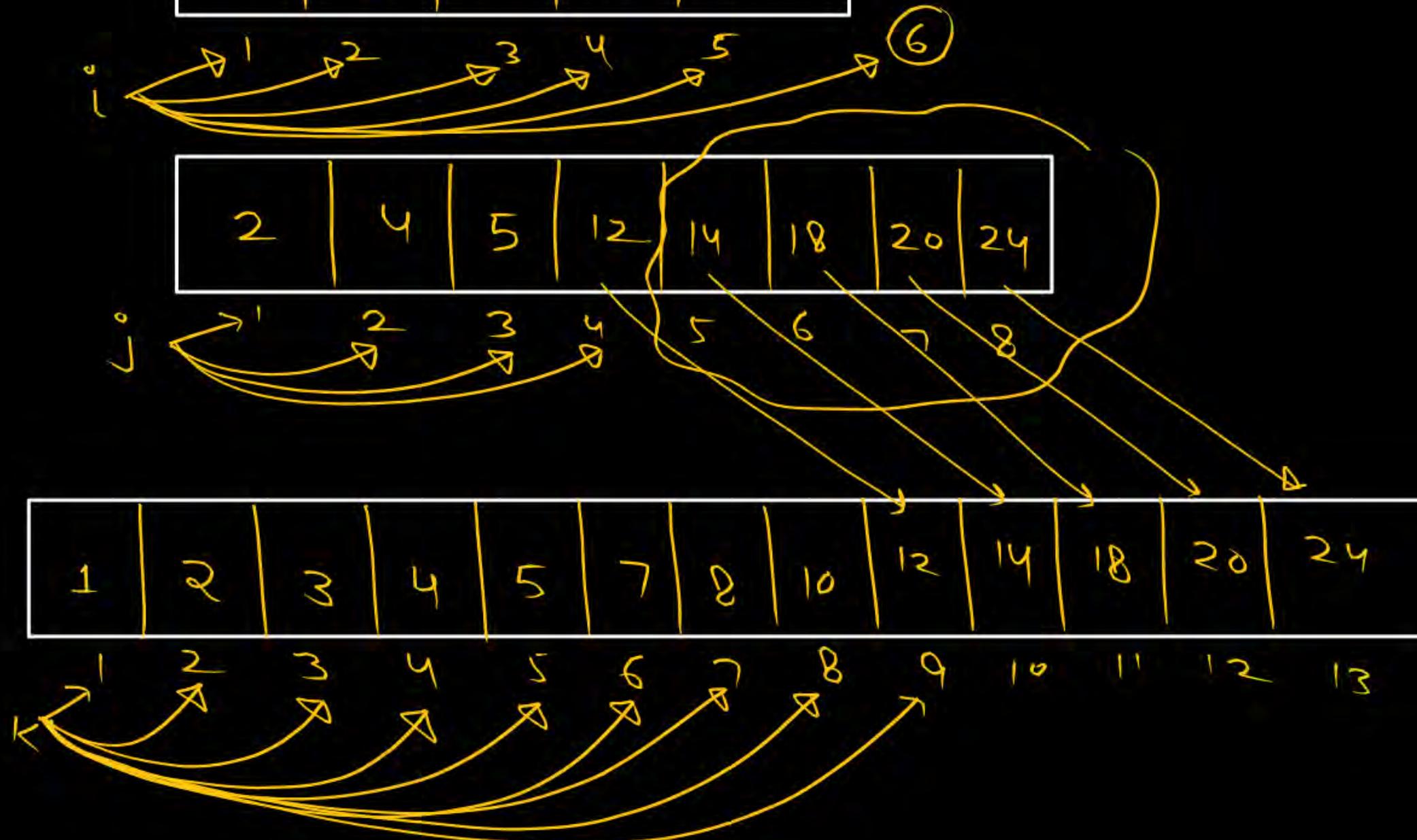
$$n = 8$$

B

2	4	5	12	14	18	20	24
---	---	---	----	----	----	----	----

C

1	2	3	4	5	7	8	10	12	14	18	20	24
---	---	---	---	---	---	---	----	----	----	----	----	----



④ Search      Linear Search

    ↳ Unsorted  $\rightarrow O(n)$

    ↳ Sorted  $\rightarrow$  Binary search  $\Rightarrow O(\log_2 n)$

⑤ Traverse :  $O(n)$

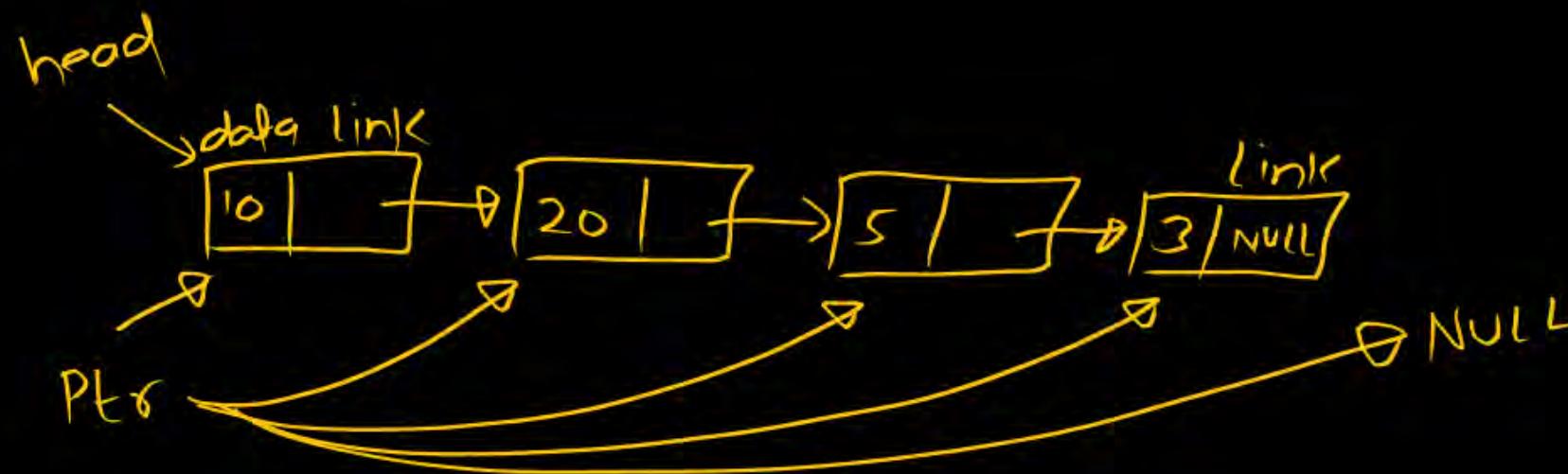
Code

START / Head  $\Rightarrow$  Global

① Traversal

$\hookrightarrow O(n)$

Linked list



```
struct Node * ptr = head;
```

```
while (ptr != NULL)
{
    pf("::d", ptr->data);
    ptr = ptr->link;
}
```

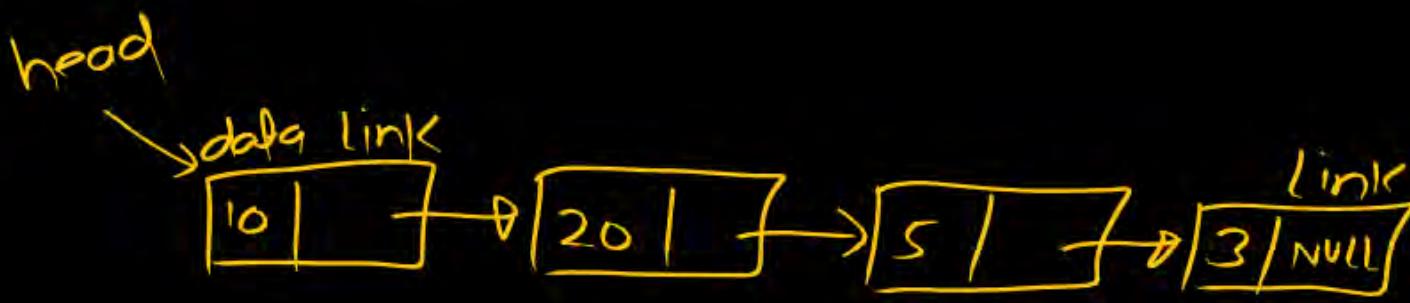
Code

START / Head  $\Rightarrow$  Global

Linked list

① Traversal :

$\hookrightarrow O(n)$

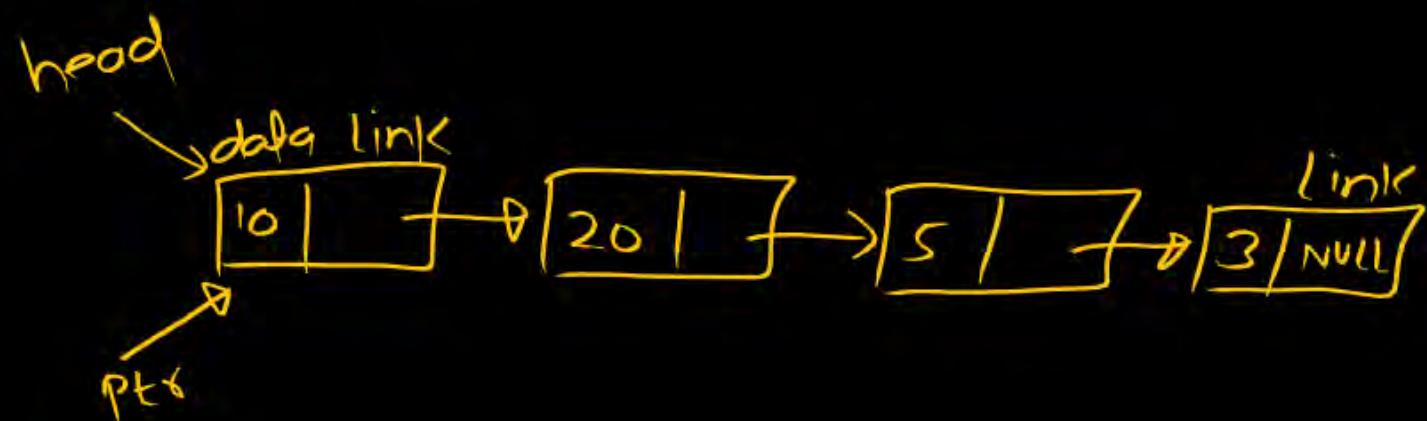


```
int count() {  
    int c = 0;  
    struct Node * Ptr = head;  
  
    while (Ptr != NULL)  
    {  
        c++;  
        printf("./d", Ptr->data);  
        Ptr = Ptr->link;  
    }  
    return c;  
}
```

Code      START / Head  $\Rightarrow$  Global

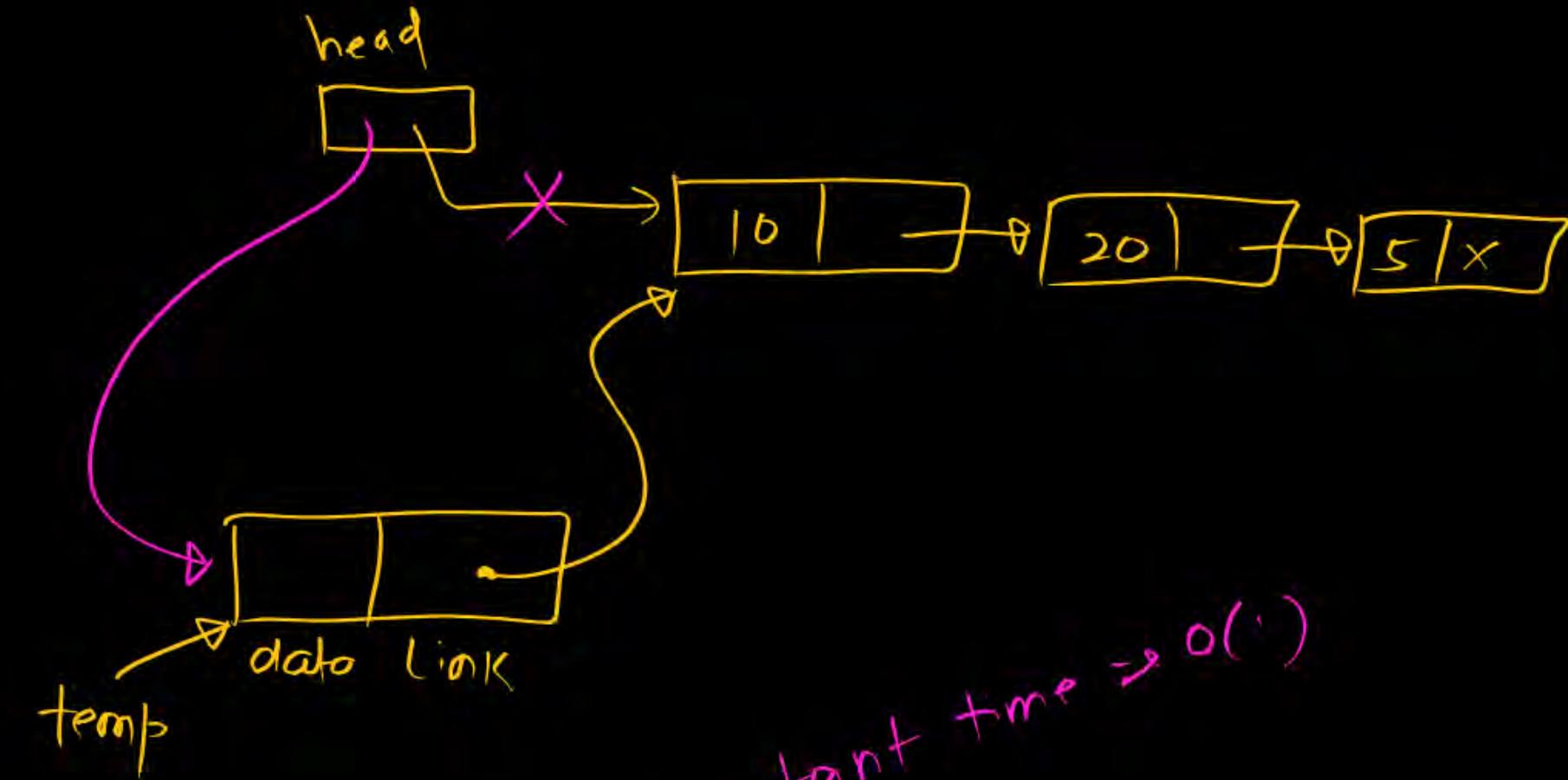
Search

Linked list



```
int Search( int x ) {  
    struct Node * Ptr = head ;  
  
    while ( Ptr != NULL )  
    {  
        if ( Ptr->data == x )  
            return 1 ;  
        Ptr = Ptr->link ;  
    }  
    return 0 ;  
}
```

Insert → At begin



```
struct Node *temp ;  
temp = malloc(sizeof(struct Node));  
temp->data = ele;  
temp->Link = head;  
head = temp;
```

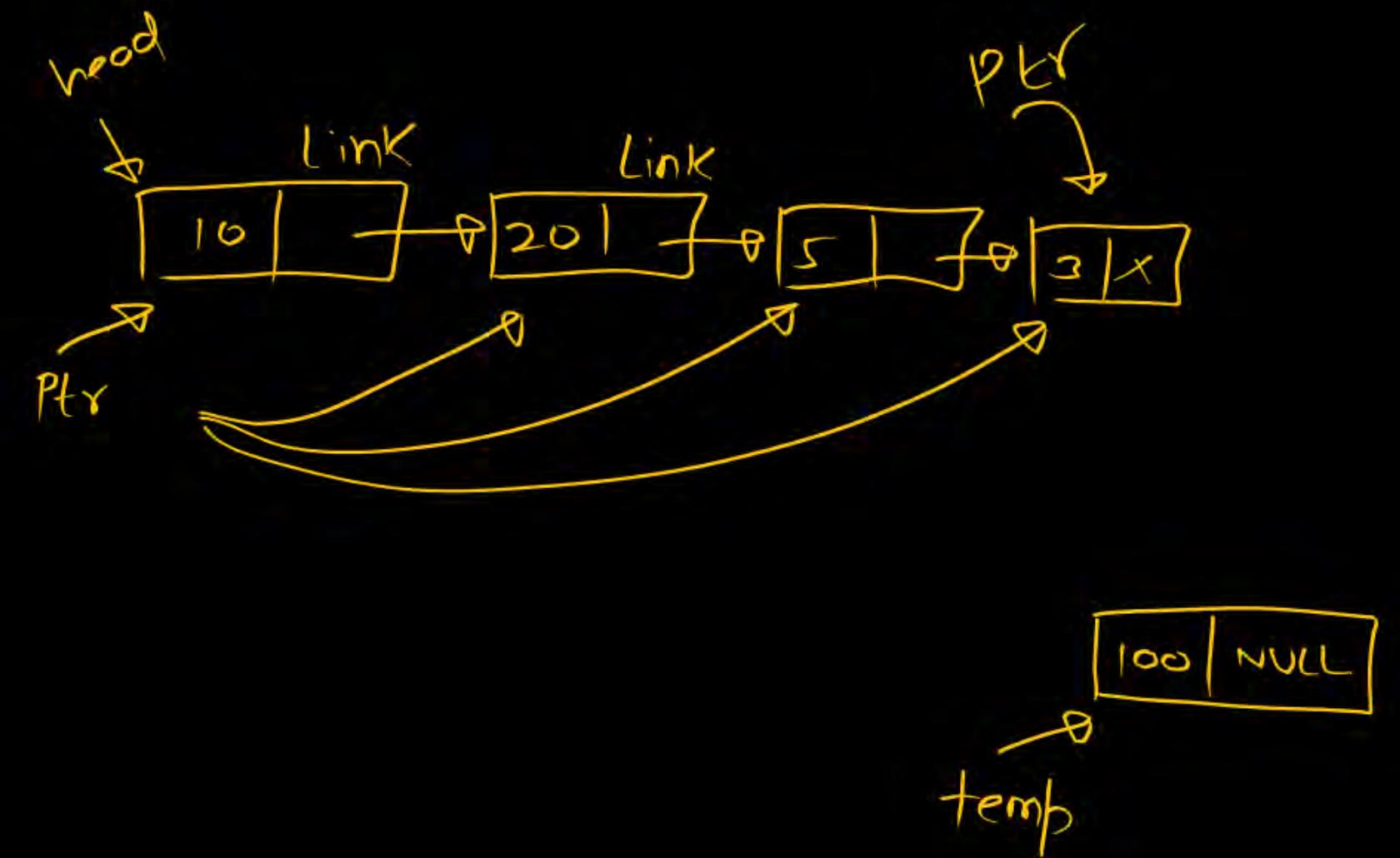
constant time  $\rightarrow O(1)$

Insert at end

```
head → NULL  
temp = malloc( );  
temp → data = ele;  
temp → link = NULL  
if (head == NULL)  
    head = temp;  
    return;  
}  
else {  
    }
```

we know  
that at least  
1 node  
exist  
↓

}

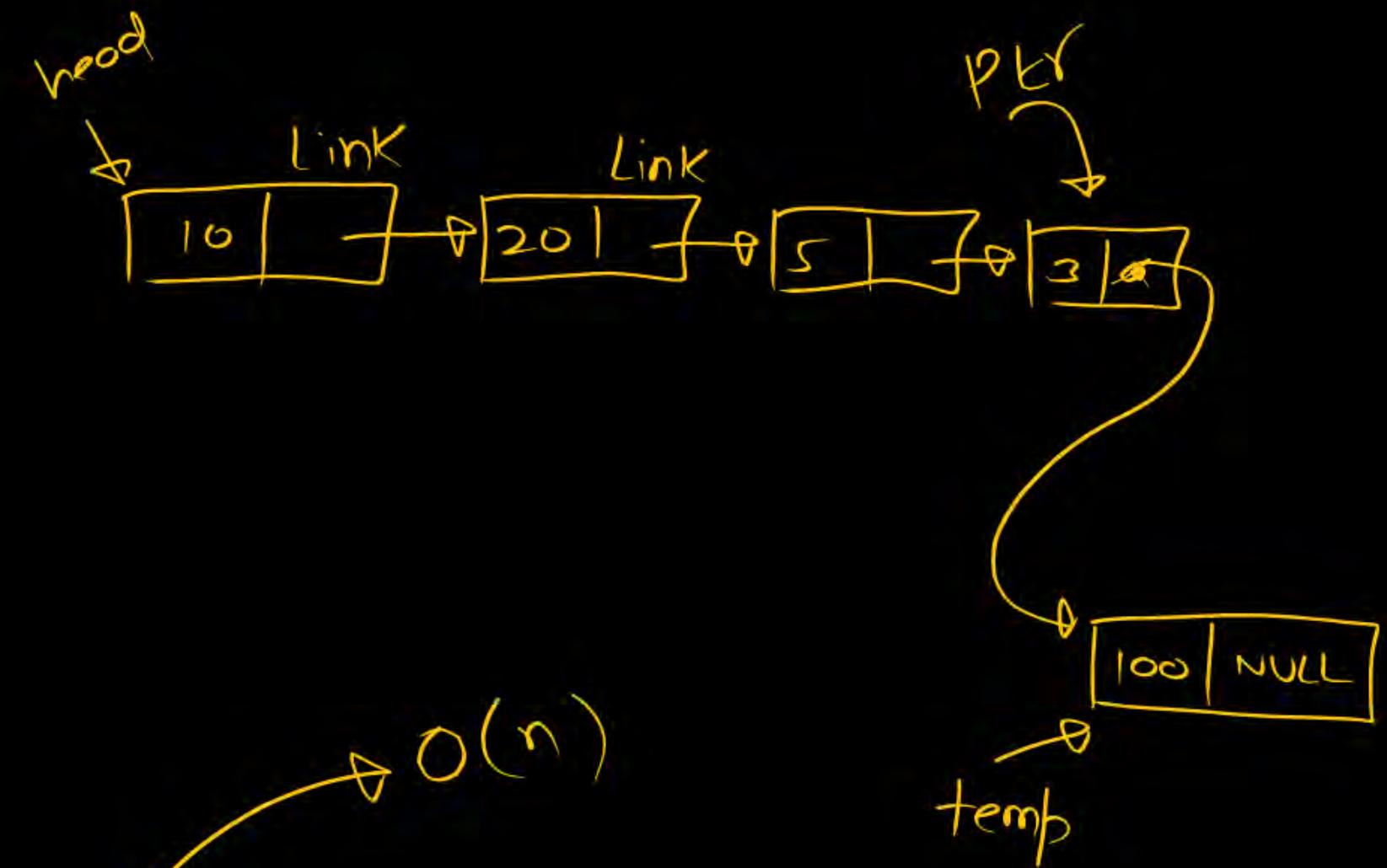


Insert at end

```
head → NULL  
temp = malloc( ____ );  
temp → data = ele;  
temp → link = NULL  
if (head == NULL)  
    head = temp;  
    return;  
}  
else {
```

1  
we know  
that at least  
1 node  
exist  
↓

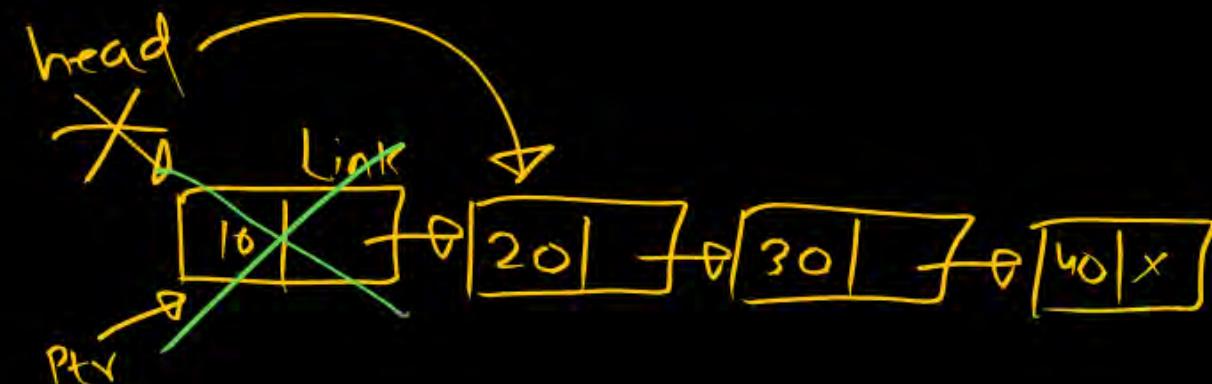
ptr = head;  
while (ptr → link != NULL)  
 ptr = ptr → link;  
 ptr → link = temp;  
}



Deletion

→ from begin!

$\uparrow$   
 $O(1)$

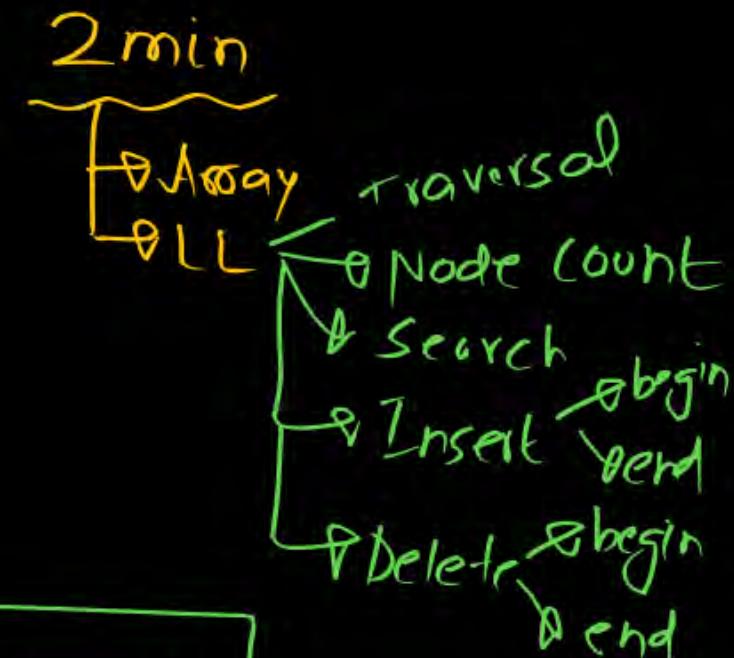


if ( $head = \text{NULL}$ )

return;

```

ptr = head;
head = head -> Link;
free(ptr);
    
```



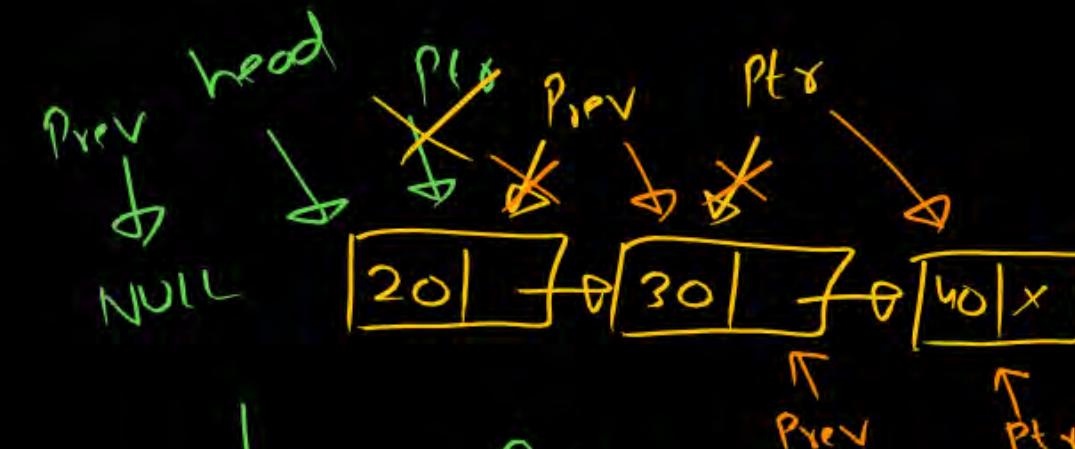
time / PW |pankajsip

Deletion → from begin!

struct Node \*ptr = head;

if (head == NULL)  
 return

if (head->link == NULL)  
{  
 head = NULL;  
 free(ptr);  
}



prev = NULL,  
ptr = head;

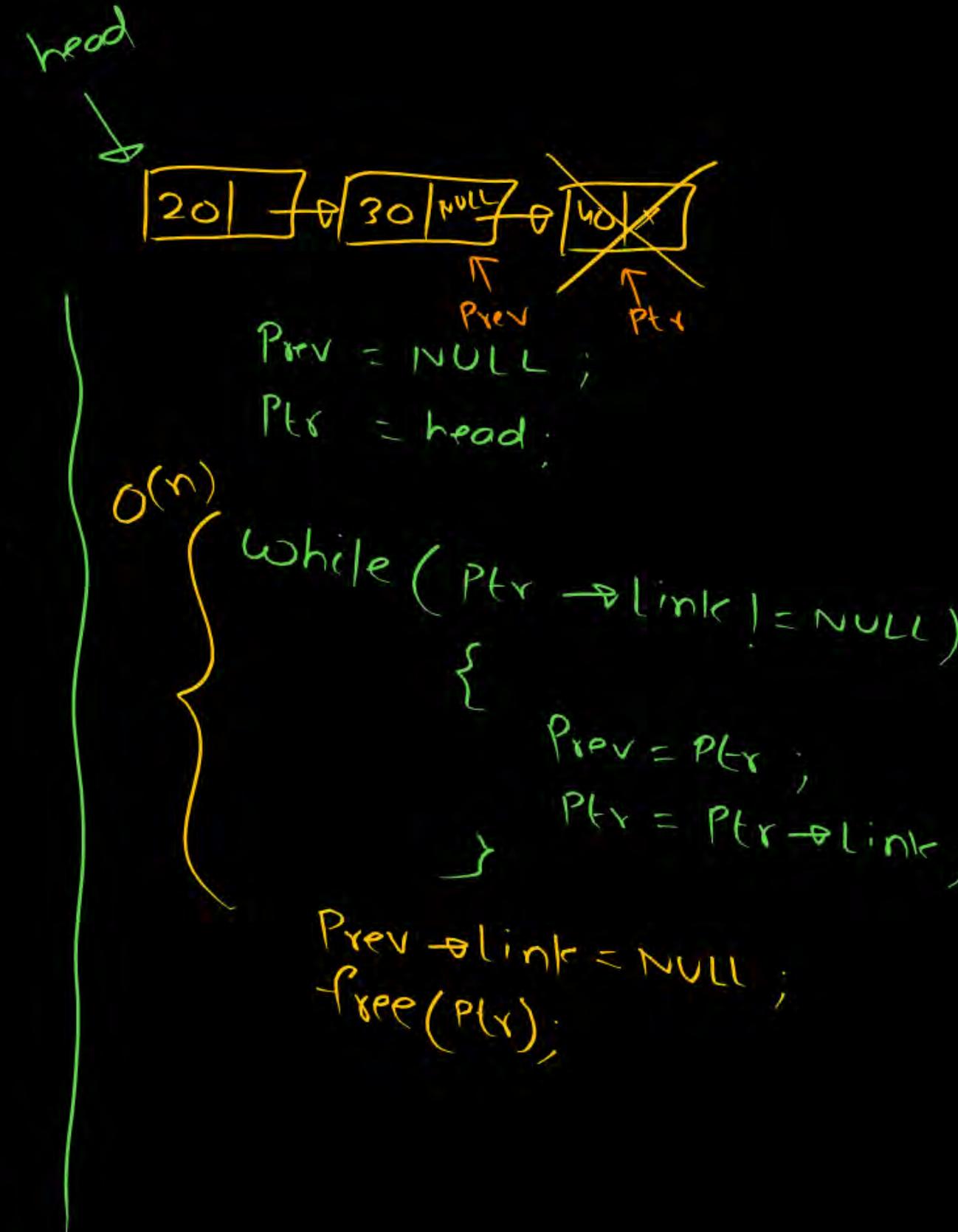
while (ptr->link != NULL)  
{  
 prev = pptr;  
 pptr = ptr->link;  
}

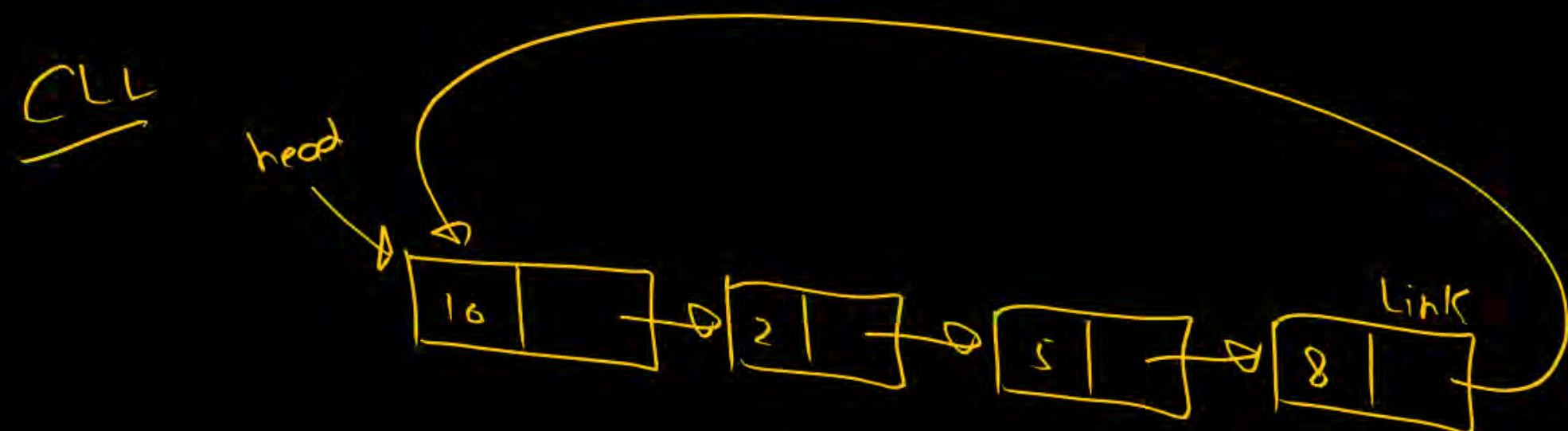
Deletion → from begin!

Struct Node \*ptr = head;

if (head == NULL)  
return

if (head → link == NULL)  
{  
head = NULL;  
free(ptr);  
}





## Stack

\* LIFO

\* Recursion

\* DFS

\* TOH

\* infix to postfix

\* infix to post-infix

\* postfix eval

\* prefix eval.

\* Balanced parenthesis check

wait ~~on call~~  
to delay / to  
postponed

infix to prefix



I/P :  $2 + 3$

O/P :  $+ 23$

operand 1  
↓  
2

operand 2  
↑  
3

I/P :  $2 + 3 * 4$

operand 1  
↓  
2

operand 2  
↓  
 $[ \frac{3}{4} ]$

prefix :  $+ 2 * 34$

## Queue

- \* FIFO
- \* Enqueue (Insert) : Rear
- \* Dequeue (Delete) : Front

## Stack

- \* LIFO
- \* Push(n) : Insert
- \* Pop() : delete

TOP

## Tree

① Binary Tree : Node can have max 2 child

0 child  $\rightarrow$  leaf node

1 child }  
2 child } internal node

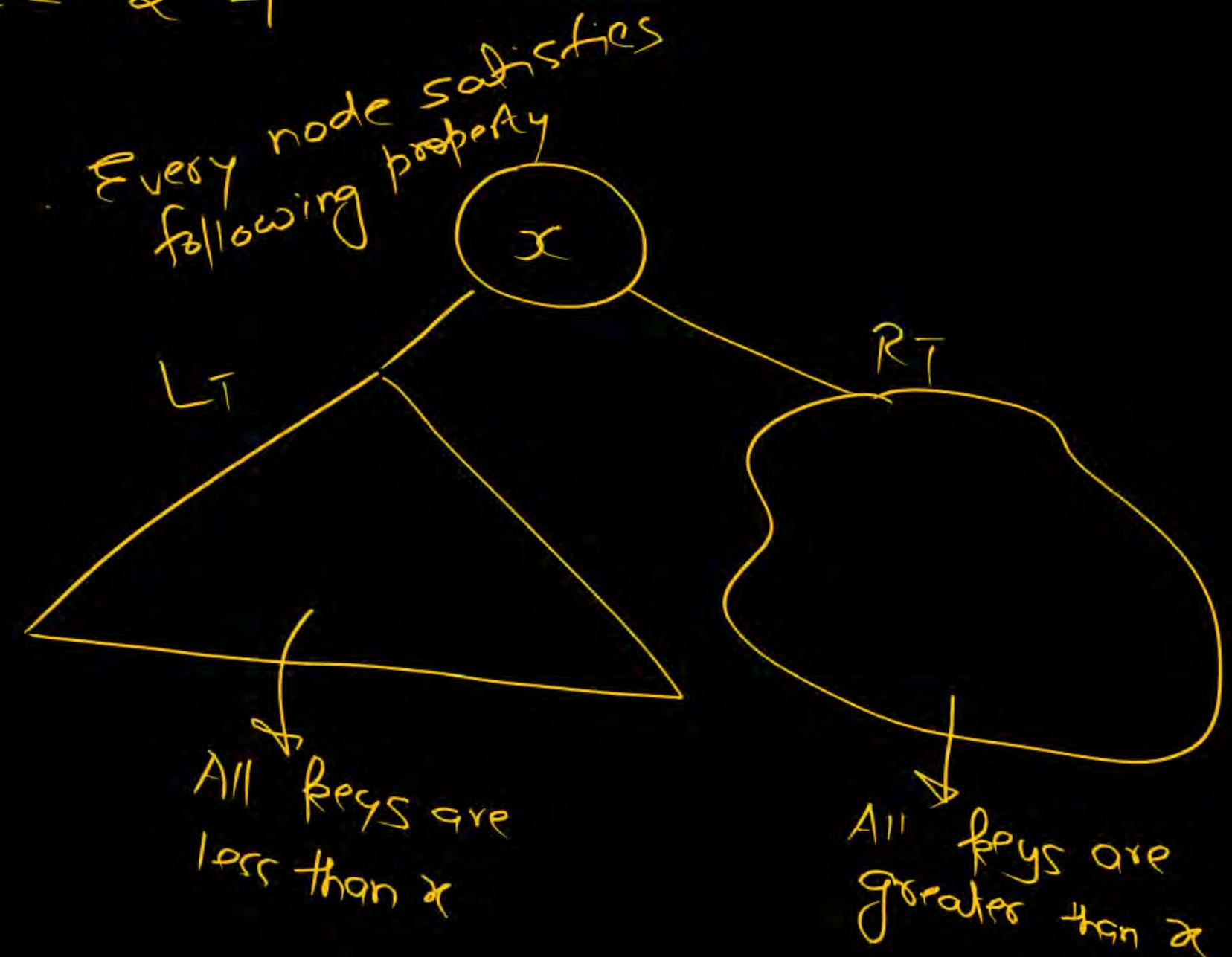
Root : No parent

$$n_{\min} = h+1 \quad (\text{1 node} \Rightarrow \text{height } 0)$$

$$n_{\max} = 2^{h+1} - 1$$

②

B.S.T



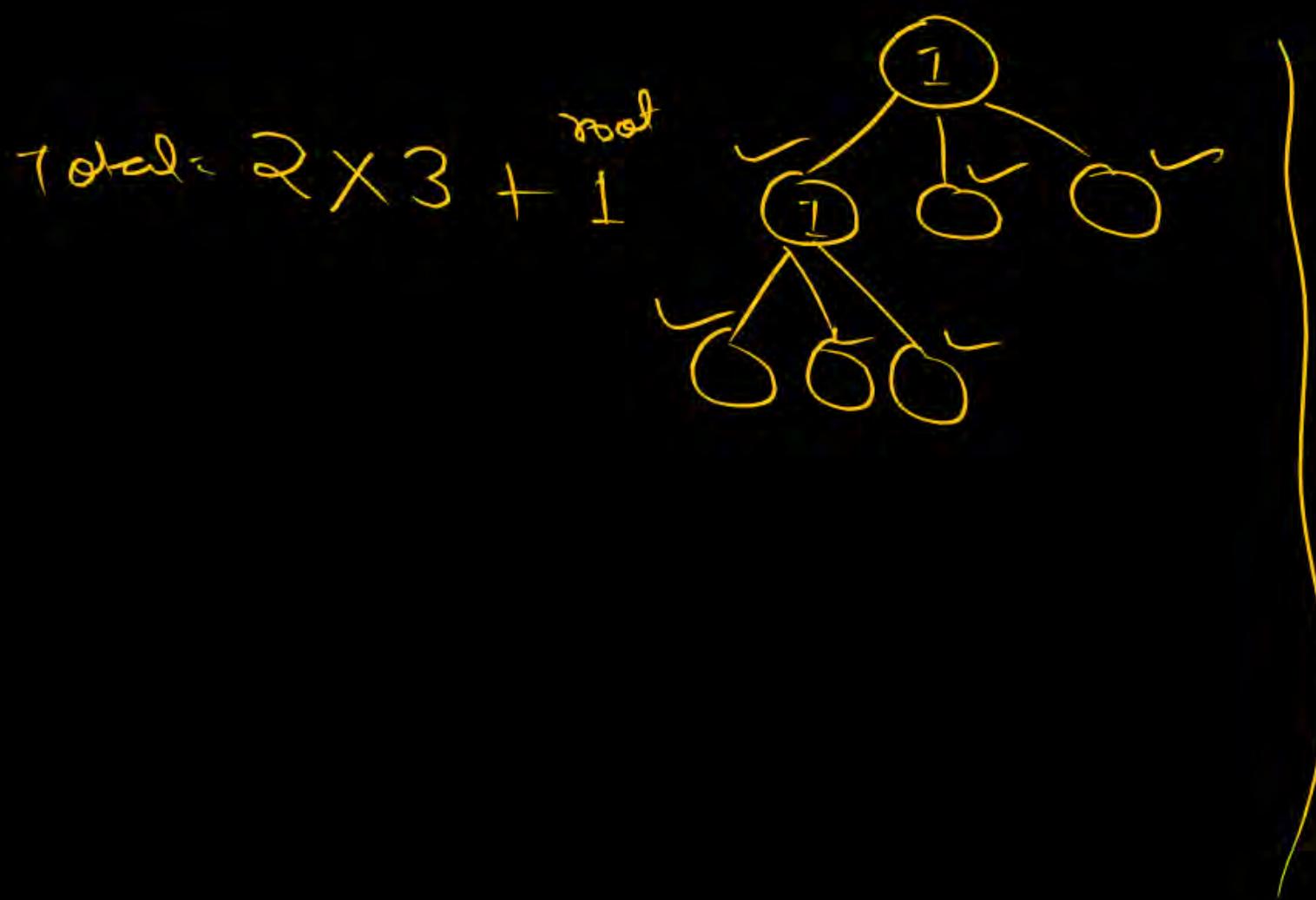
BST

a) Every node

b) Inorder traversal : increasing order of keys.

K-ary tree

Internal node  $\Rightarrow$  exactly  $K$  child



Total =  $I \times K + 1$

no. of internal node  
Each internal node has  $K$  children

$$N = KI + 1 \quad (1)$$

$$L + I = KI + 1$$

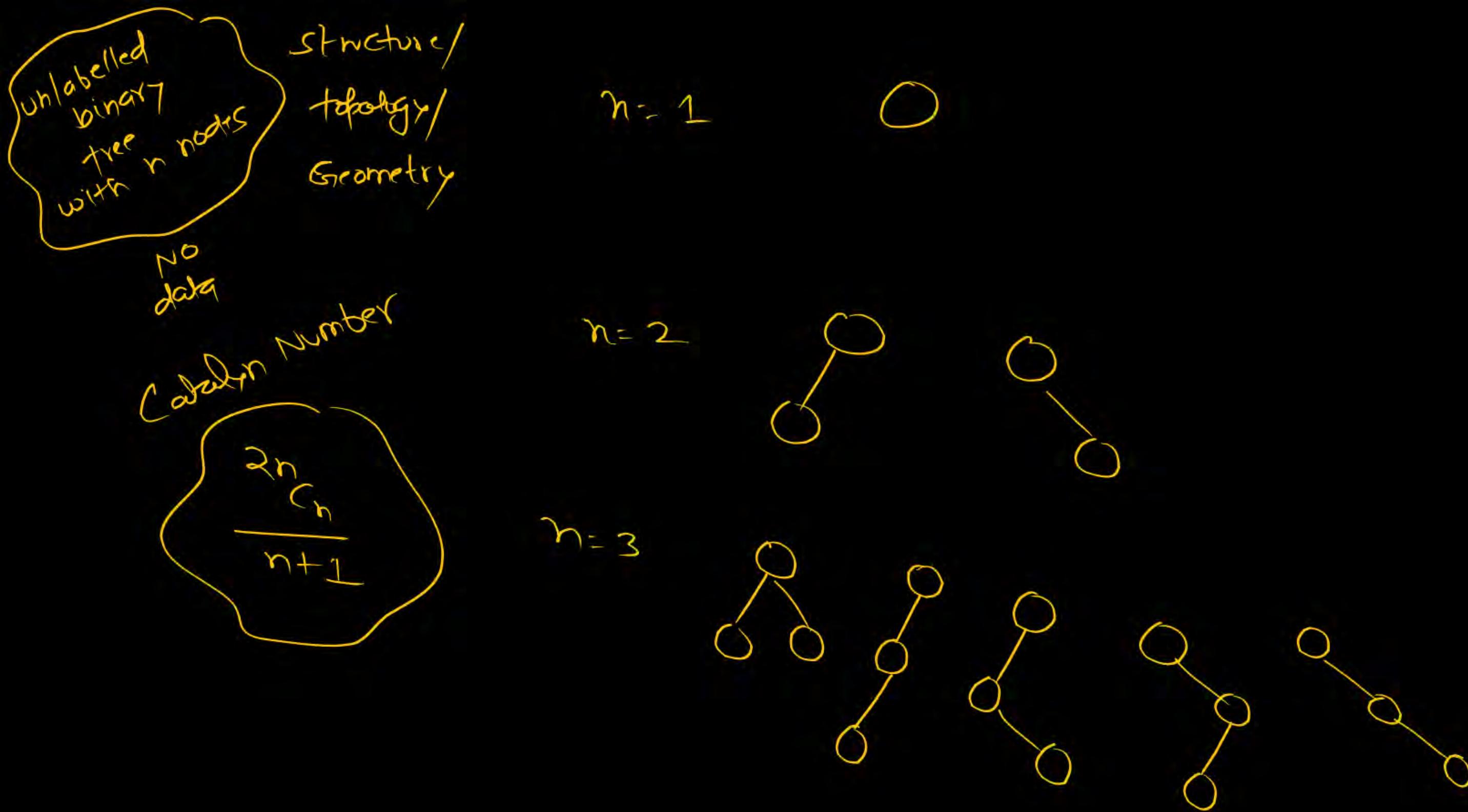
$$L = KI - I + 1$$

$$L = I(K-1) + 1 \quad (2)$$

$$n_0 = n_2 + 1$$

no. of  
leaf node  
(0-child)

no. of  
nodes  
with exactly  
2 childs

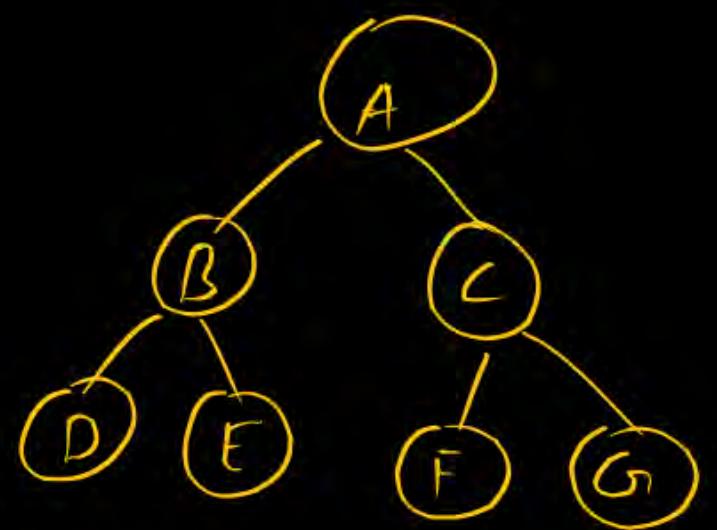


$$\text{No. of BST with } n \text{ keys} = \frac{2^n C_n}{n+1}$$

3, 2, 1

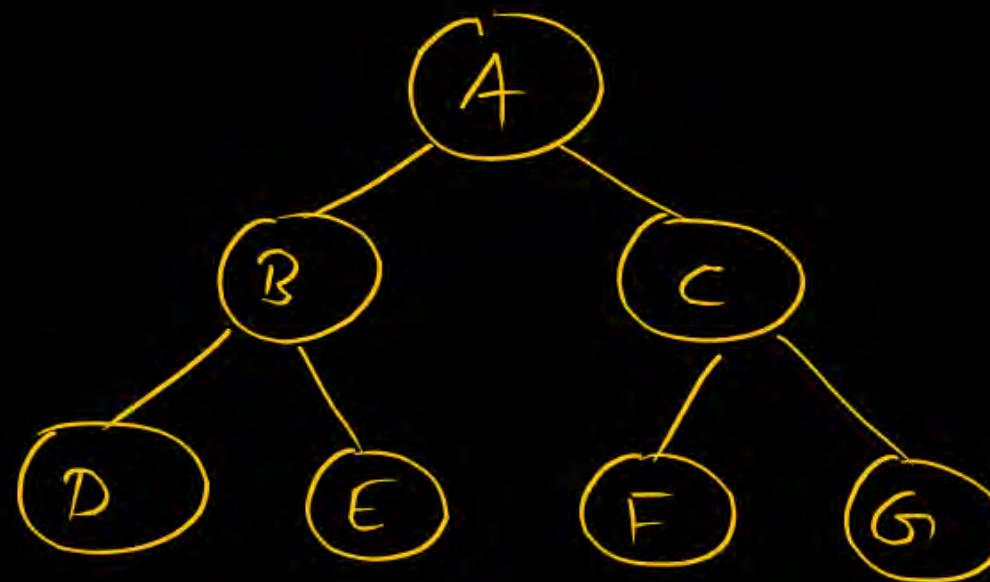
Preorder : AB DE CFG

Inorder : DBEAFCG



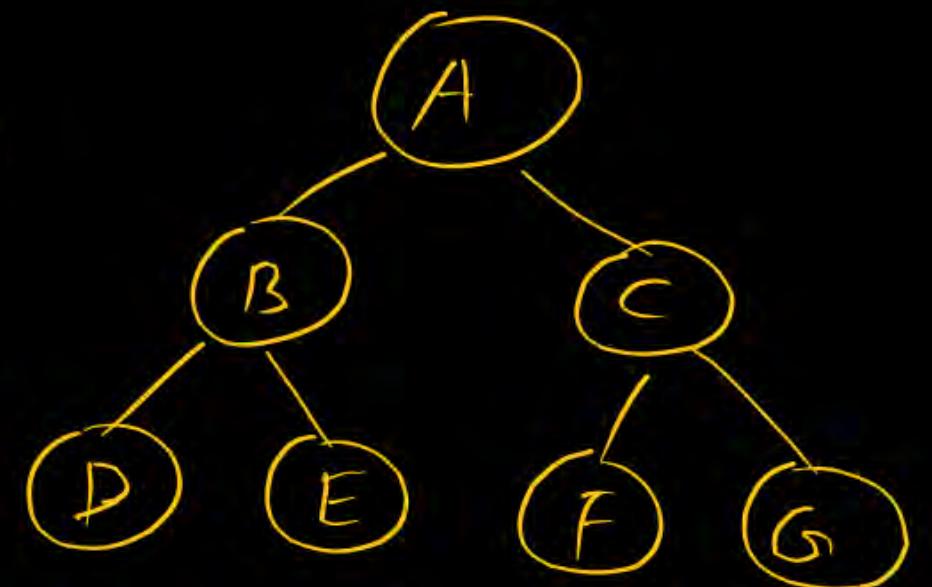
Preorder :  $\overbrace{AB}^{\text{insert}} \overbrace{DE}^{\text{insert}} \overbrace{CFG}^{\text{insert}}$

Inorder :  $D \boxed{B} E A \boxed{F} C G$



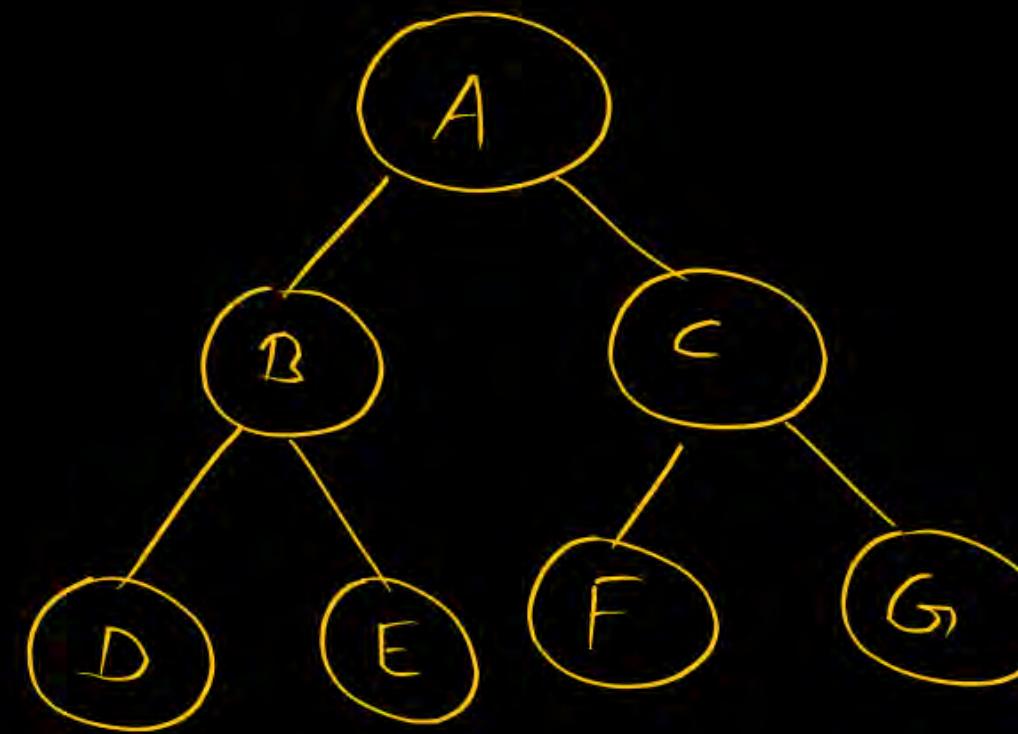
Post : D E B F G C A

In : D B E A F C G



Post : D E B F G C A

In : D [B] E A F C G





## 2 mins Summary

ISRO PYQ SERIES

P  
W

**Topic**

One

Array → insert, delete, traversal, search, merge

**Topic**

Two

LL → insert, traverse, delete, Node count, search

**Topic**

Three

Stack & Queue → application

**Topic**

Four

Trees → binary tree, BST, pre, inorder, postorder

**Topic**

Five

**THANK - YOU**



# CS & IT ENGINEERING



## PYQ SERIES

C Programming & Data  
Structures

Lecture No.- 05

By- Pankaj Sharma Sir



# Recap of Previous Lecture



Topic

Programming & Data Structures Practice  
Problems & Revision

# Topics to be Covered



Topic

Programming & Data Structures

[AVL tree  
Questions]



# Programming & Data Structures

ISRO PYQ SERIES

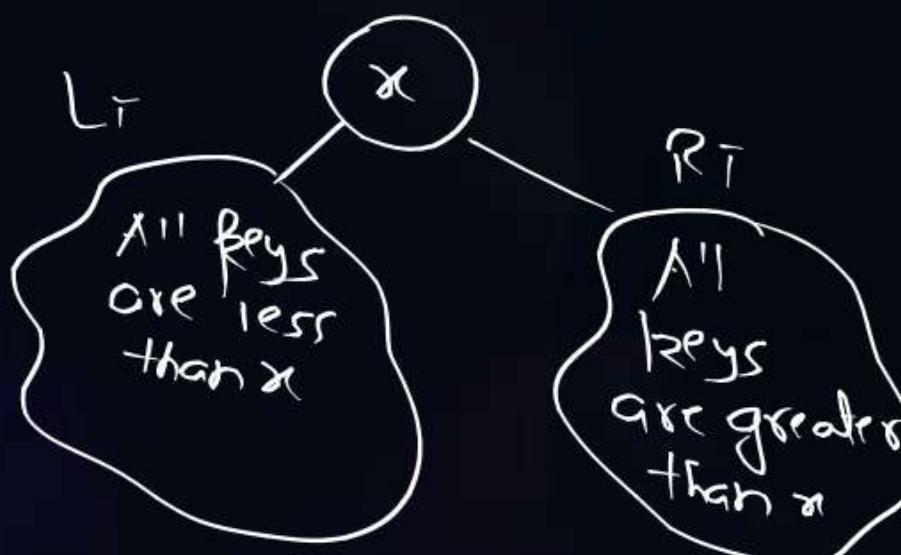
P  
W

AVL tree

height balanced search tree

A BT in which every node satisfies 2 properties:

① BST prop.



② AVL tree property

Bal. factor of each node  
must be -1, -1 or 0

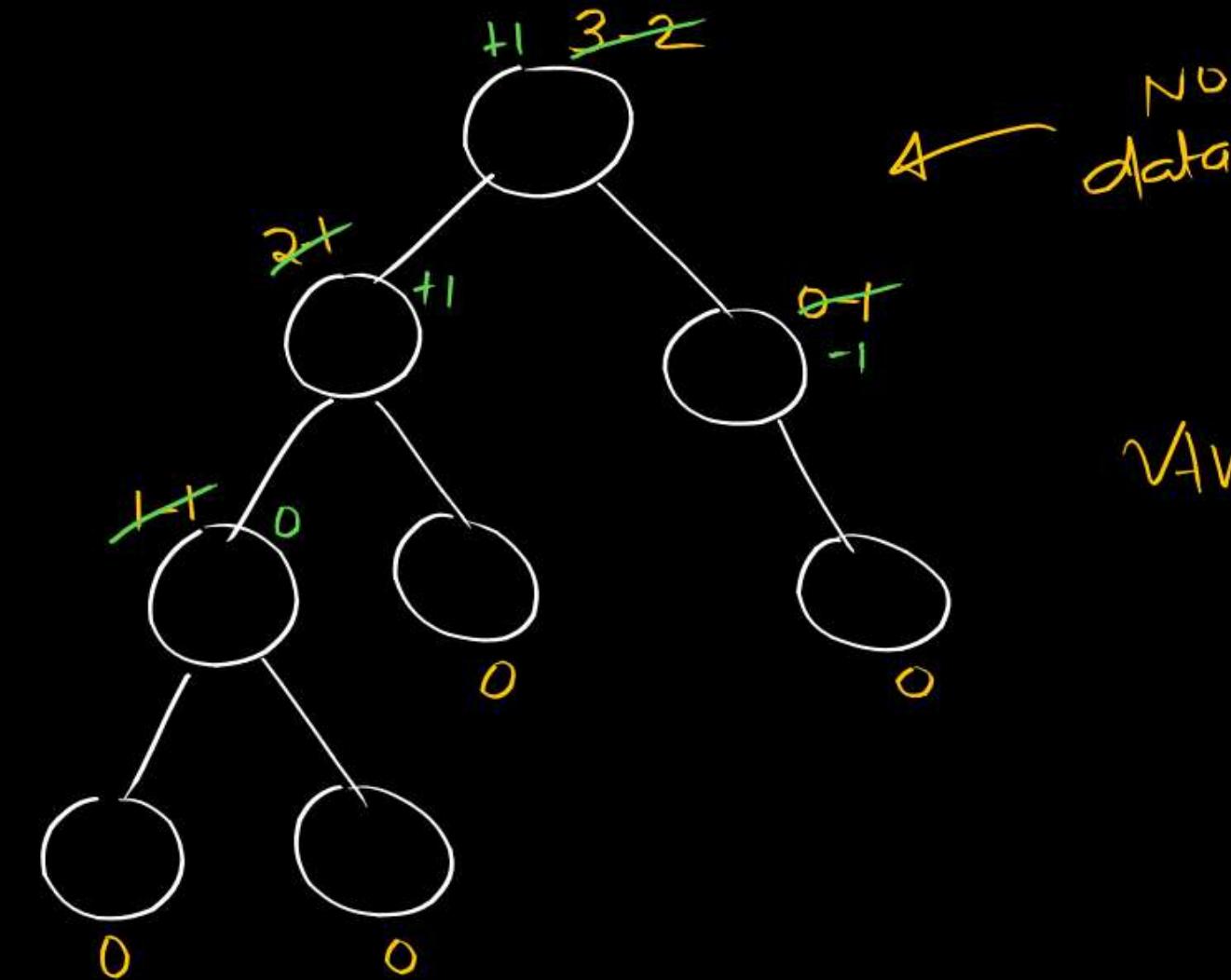
## Bal. factor

(i) leaf  $\rightarrow 0$

(ii) left  $\rightarrow$

-

Right  $\rightarrow$



AVL tree

$\hookrightarrow$  structure related

Is it a BST? (we need data to answer)

Insert  $\rightarrow$  AVL tree

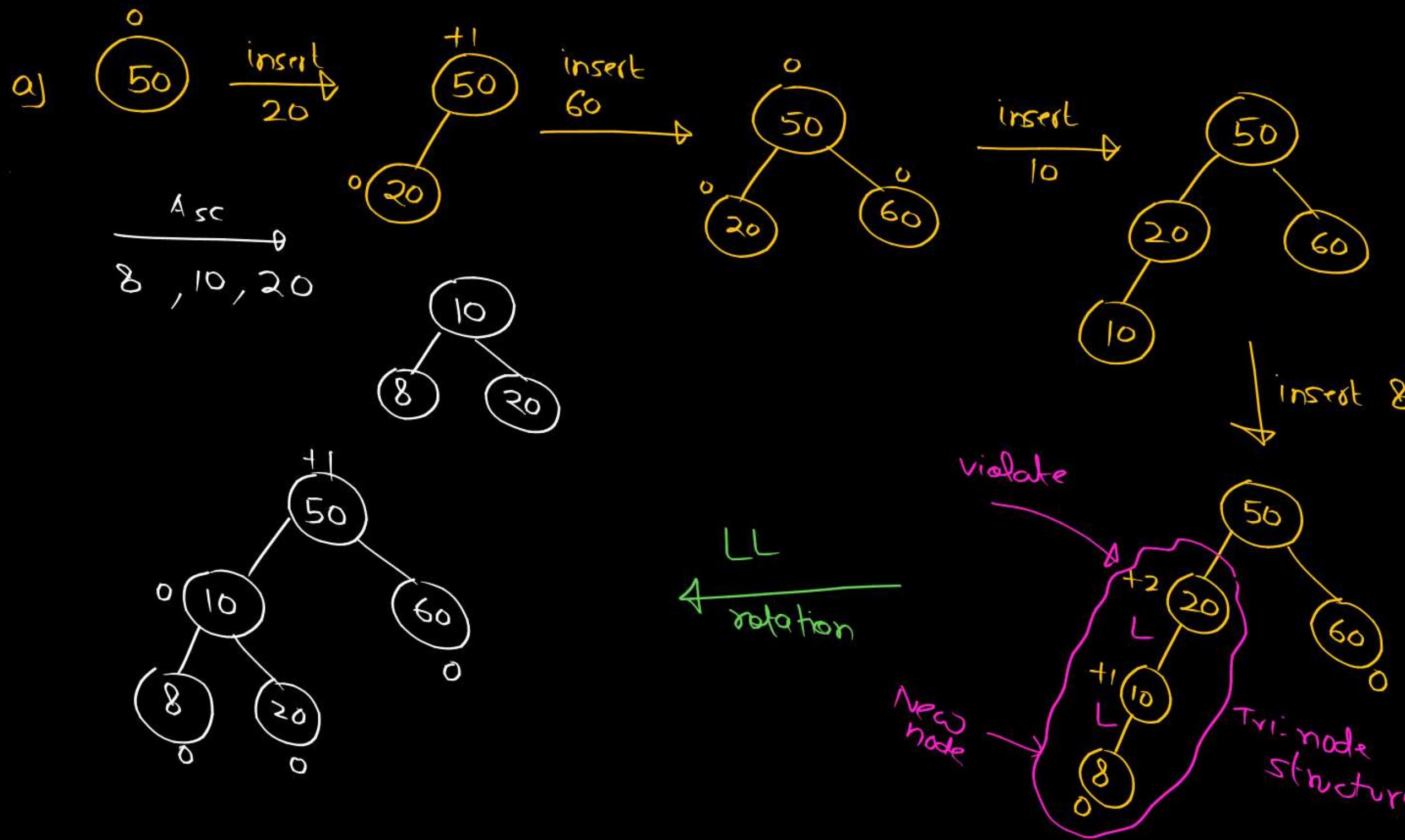
Same as BST

but  $\rightarrow$  Bal. factors  $\rightarrow$  other than  $-1, 0, +1$



Unbalanced  
 $\downarrow$   
rotations

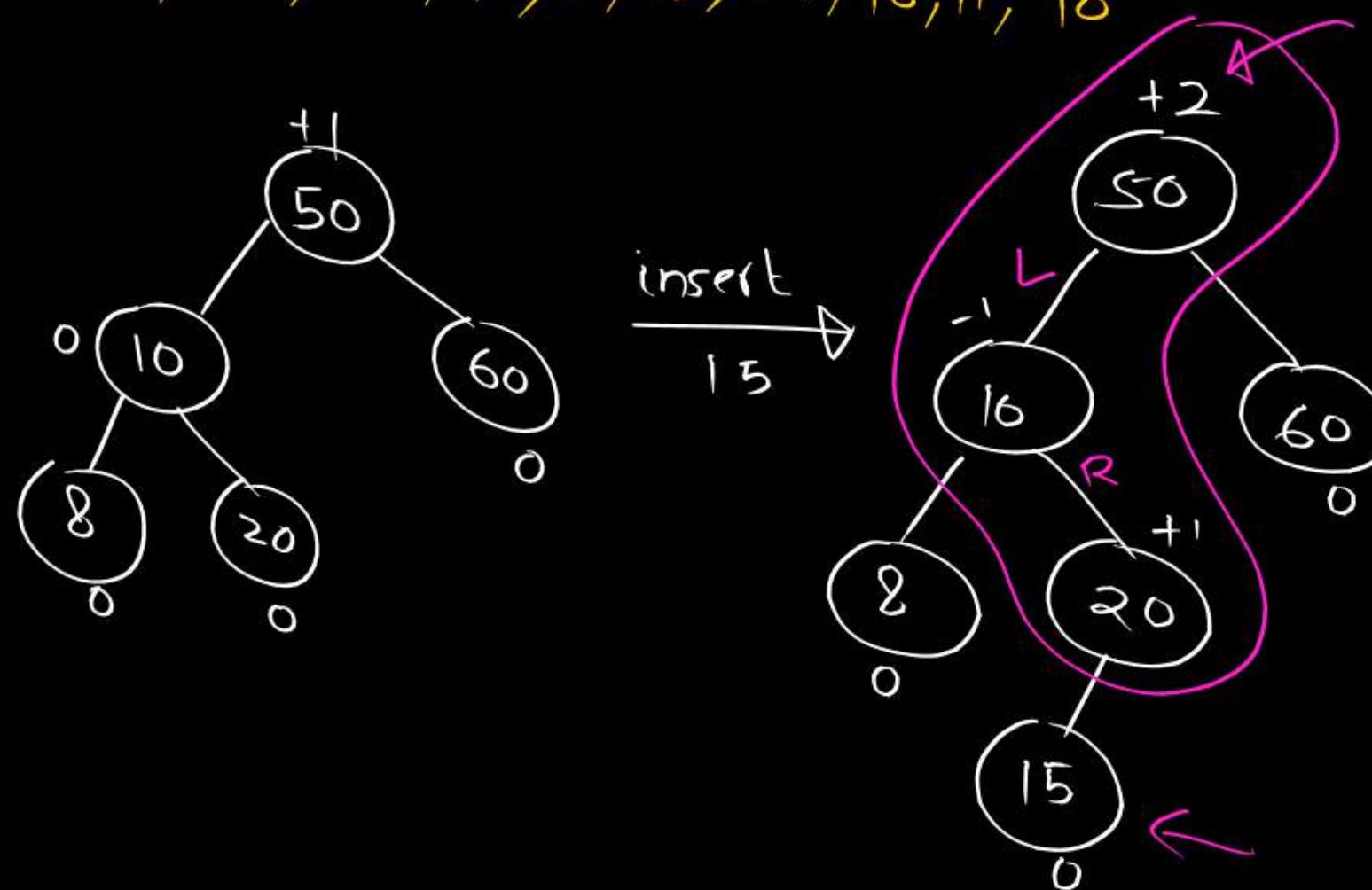
50, 20, 60, 10, 8, 15, 32, 46, 11, 48



10 20 50

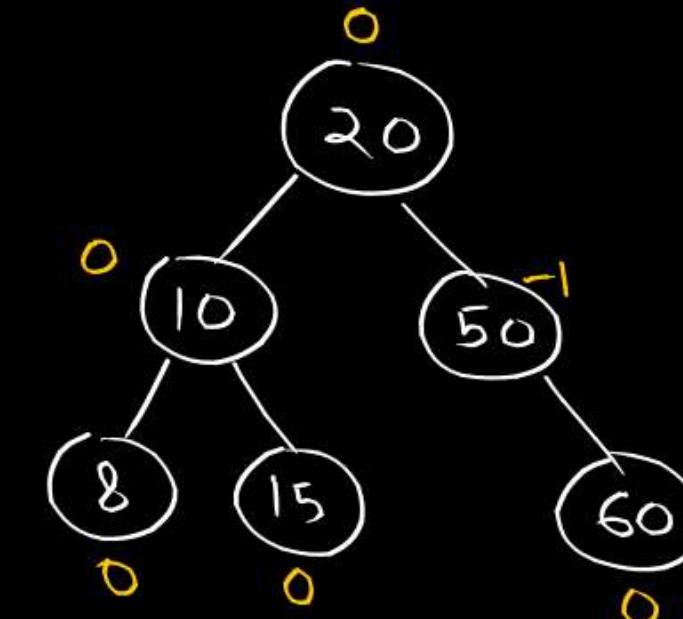
↓

50, 20, 60, 10, 8, 15, 32, 46, 11, 48

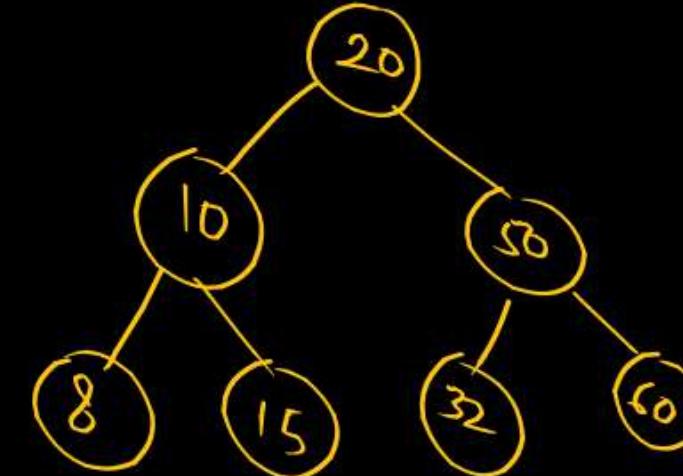


insert  
15

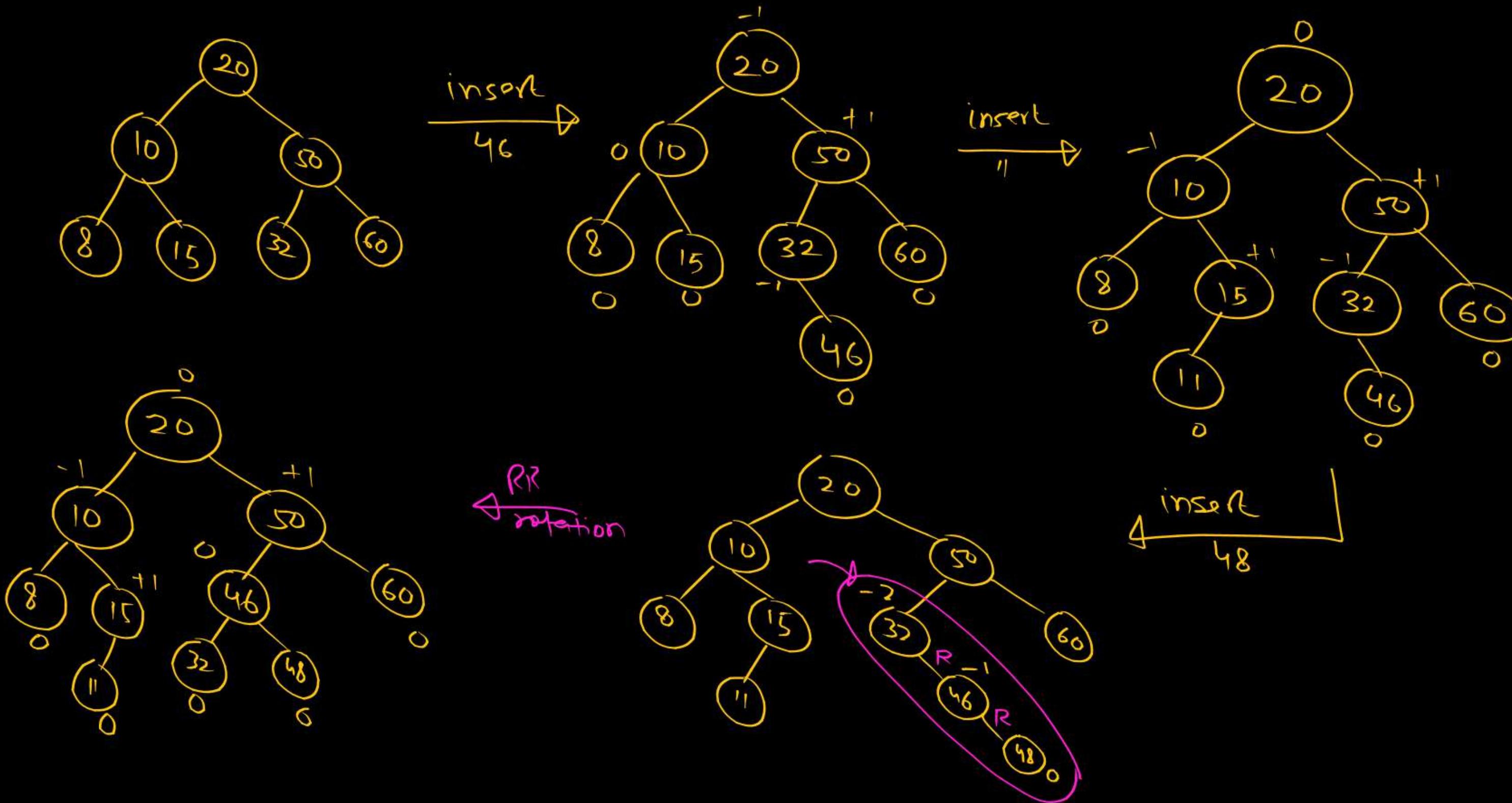
LR  
rotation  
(double rotation)



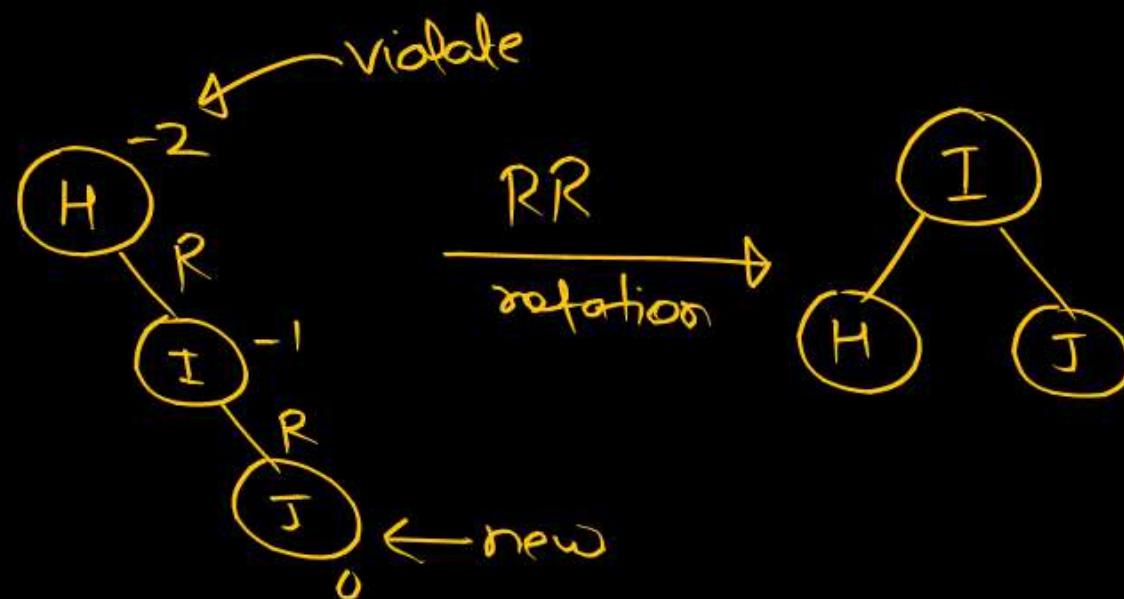
insert  
32



50, 20, 60, 10, 8, 15, 32, 46, 11, 48



2. H, I, J, B, A, E, C, F, D, G, K, L

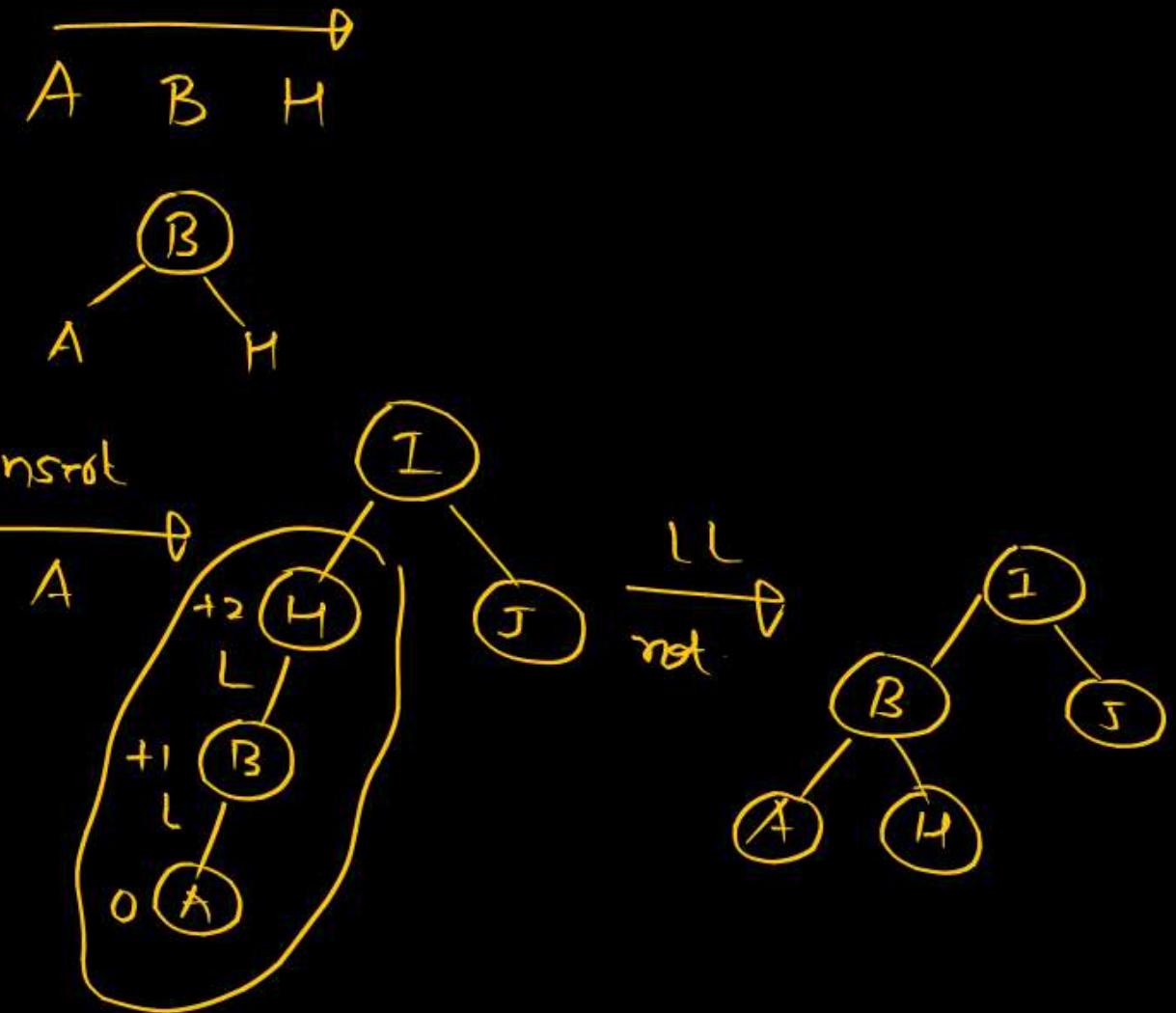
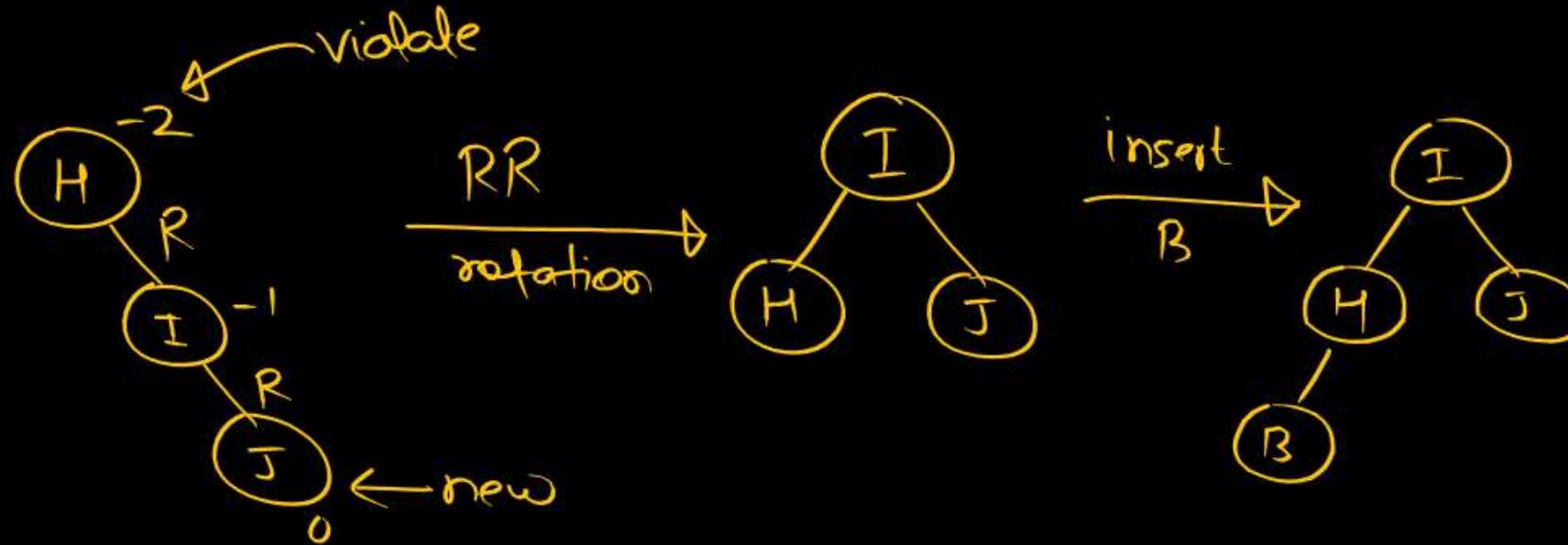


single rotation

double rotation

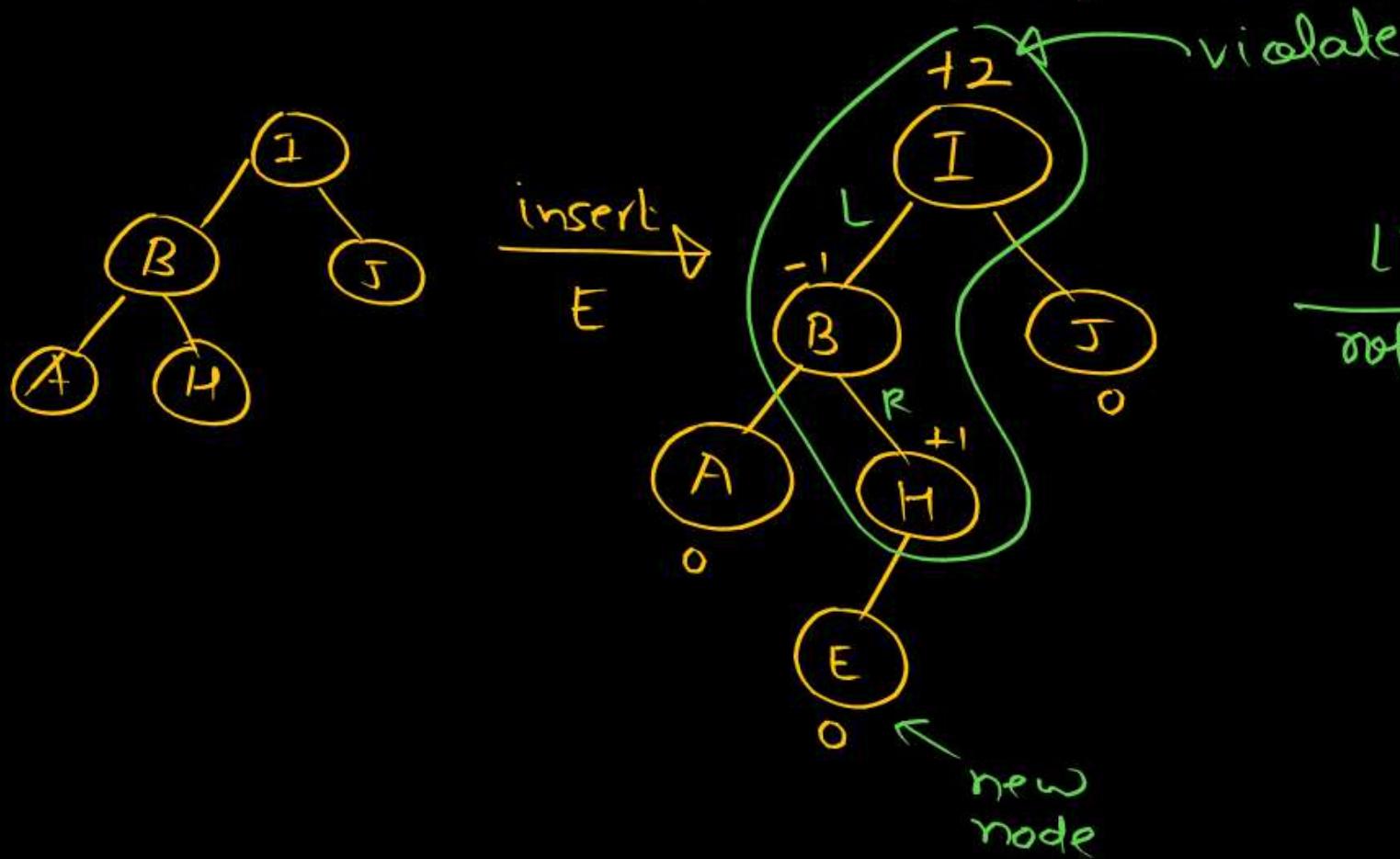
LL  
LR  
RL  
RR

2. H, I, J, B, A, E, C, F, D, G, K, L

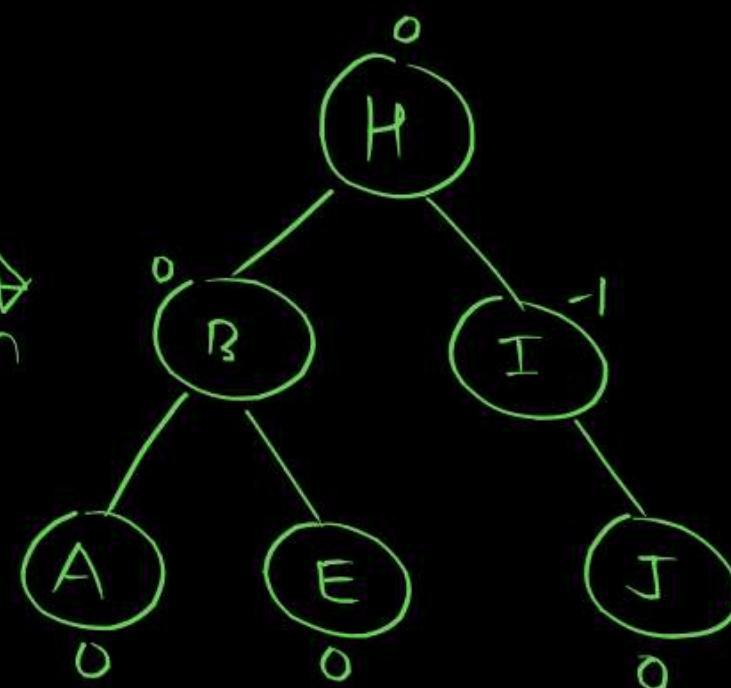


2.

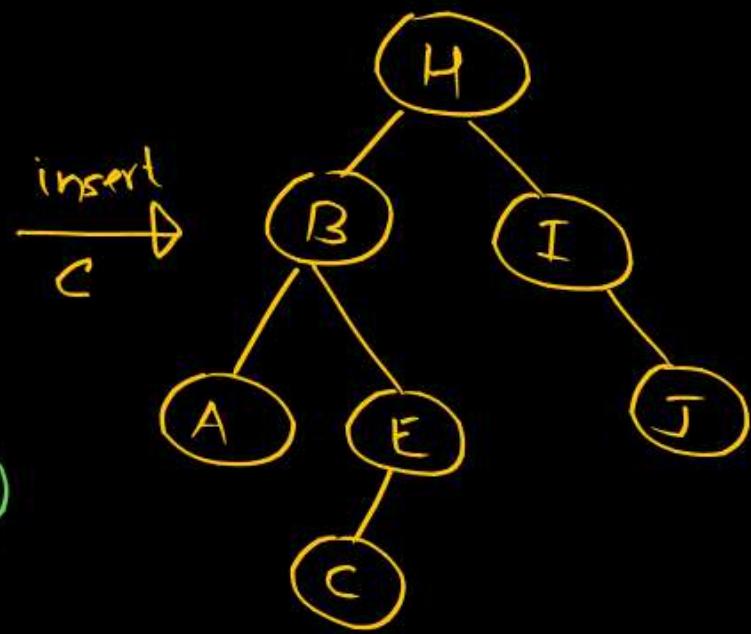
H, I, J, B, A, E, C, F, D, G, K, L



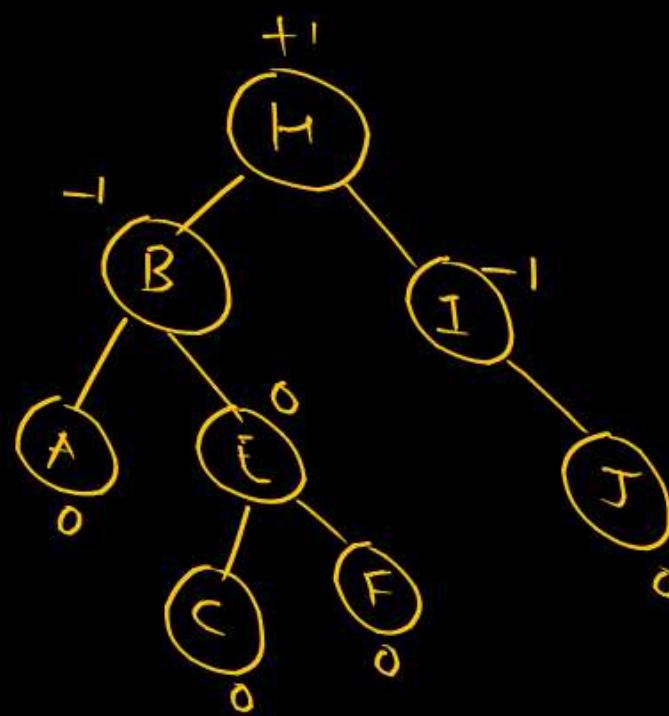
LR  
rotation



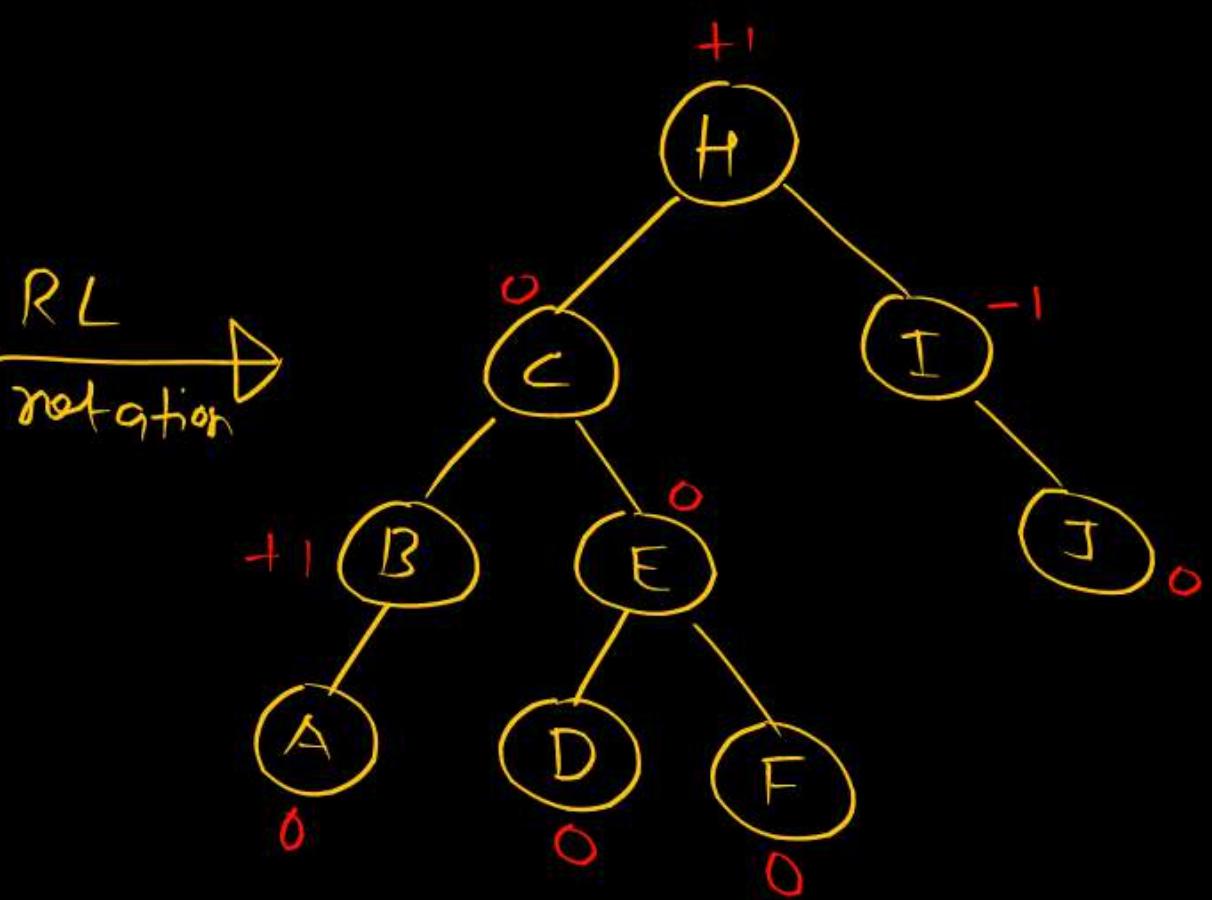
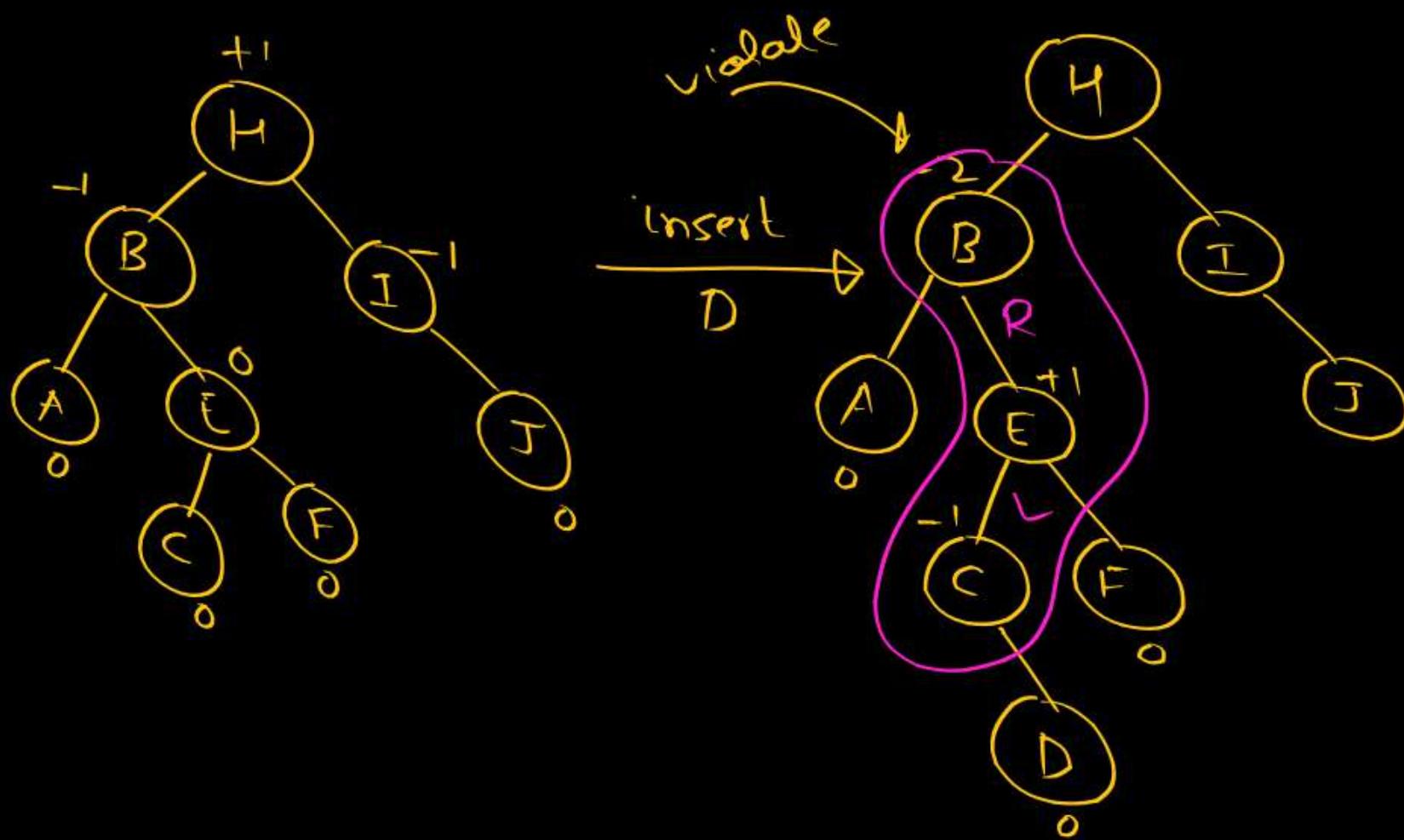
insert  
C



insert  
F

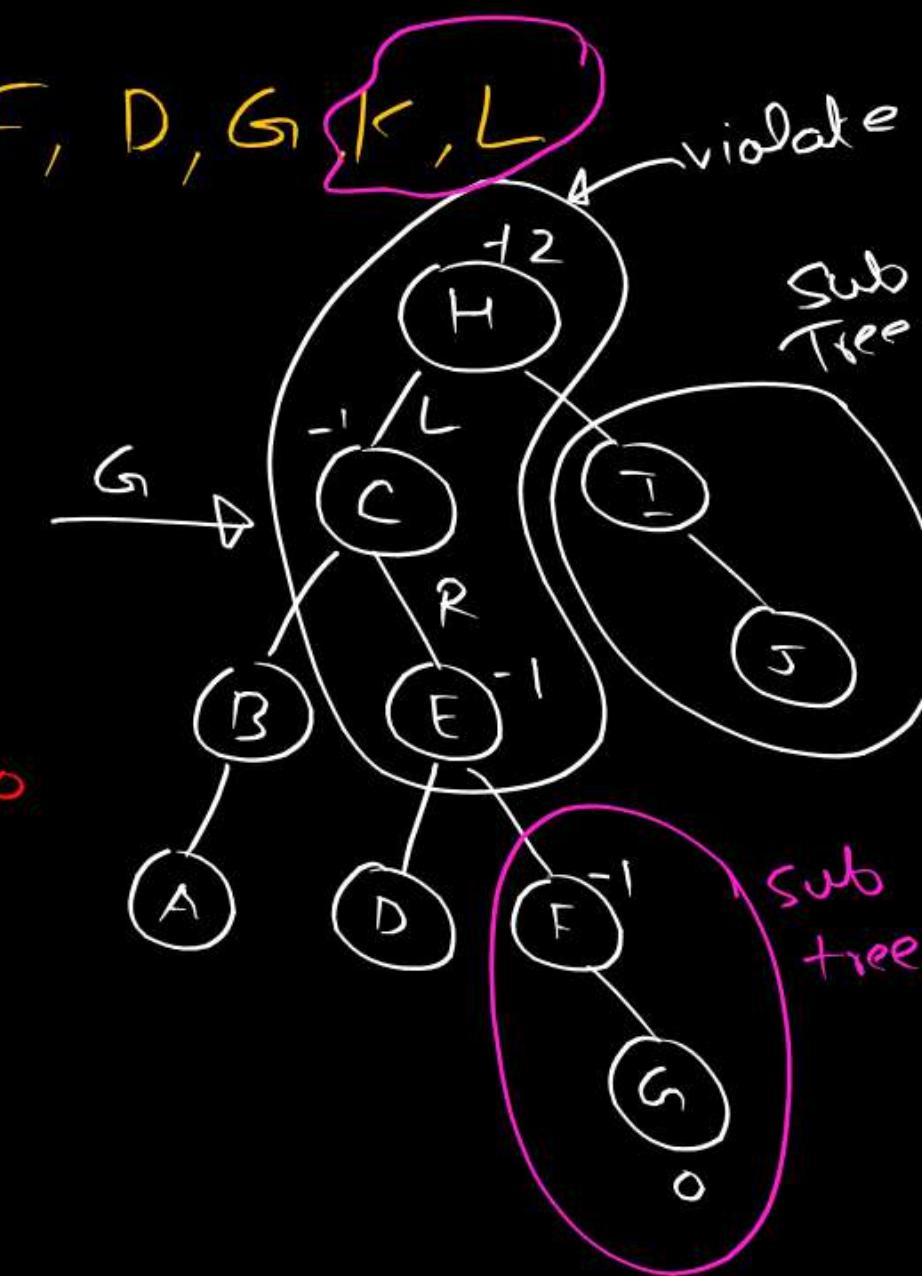
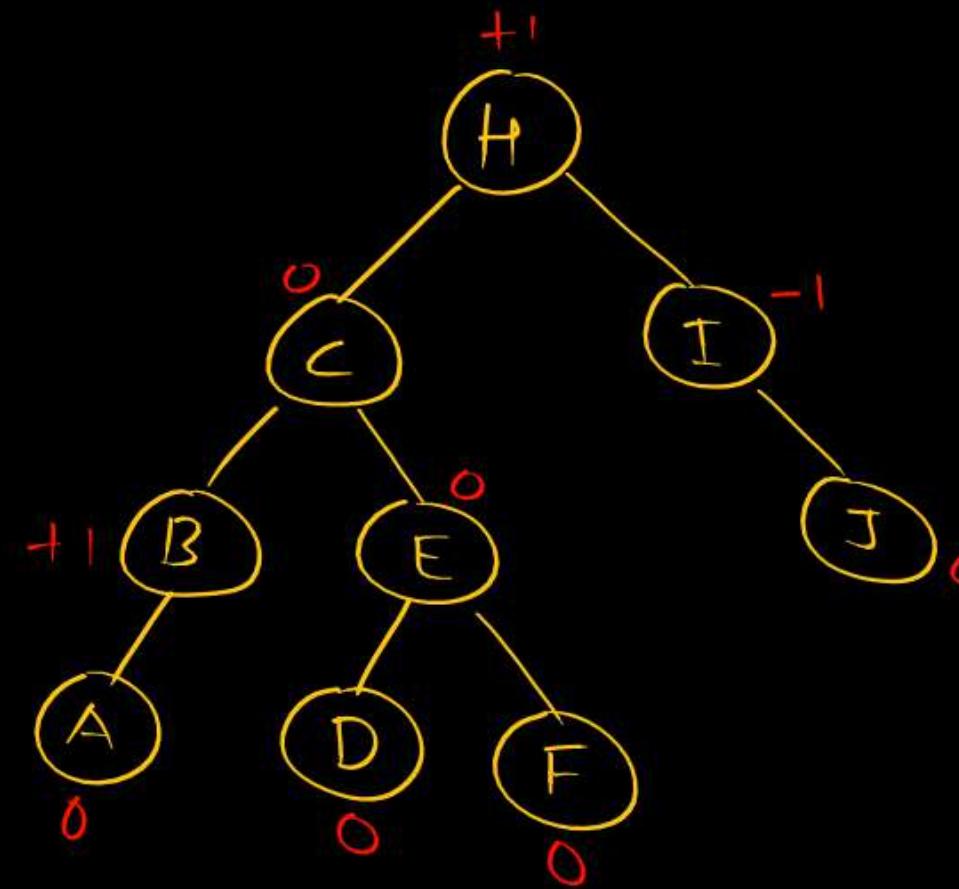


2. H, I, J, B, A, E, C, F, D, G, K, L

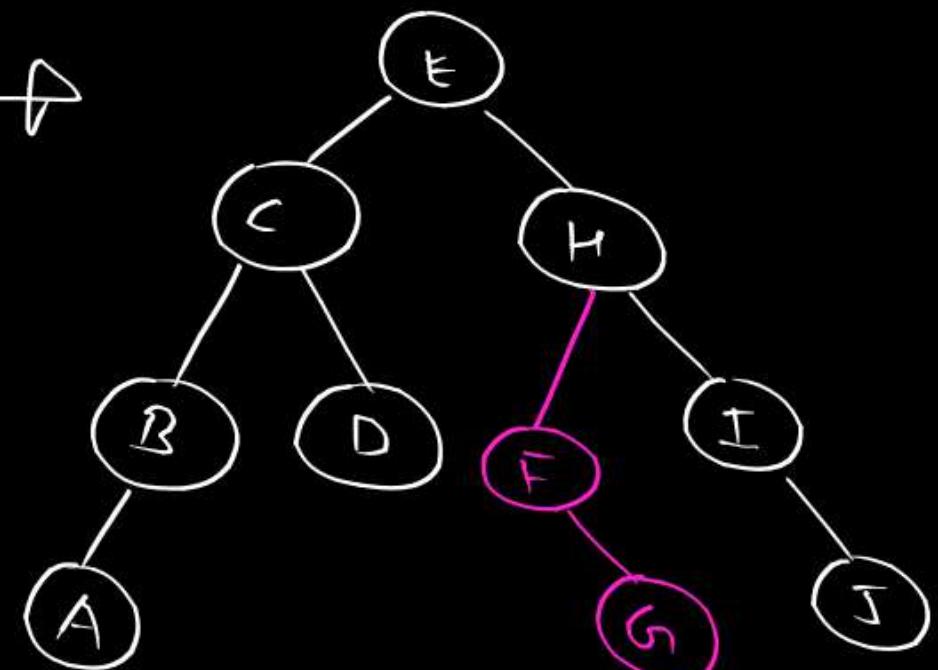


2.

$H, I, J, B, A, E, C, F, D, G, \{K, L\}$



$B \quad C \quad E$   
 $C \quad E \quad H$



⑨

$$10 \quad 5 \quad + \quad 60 \quad 6 \quad / \quad \times \quad 8 \quad -$$

a) 284

b) 213

c) 142

d) 71

$$\begin{array}{r} \overbrace{10 \quad 5 \quad +}^{\rightarrow} \quad 60 \quad 6 \quad / \quad \times \quad 8 \quad - \\ 15 \quad \overbrace{60 \quad 6 \quad /} \quad \times \quad 8 \quad - \\ 15 \quad 10 \quad \times \quad 8 \quad - \end{array}$$

$$150 \quad 8 \quad -$$

$$\Rightarrow \underline{162}$$

Q which of the foll. permutations can be obtained in the O/P (in same order) using a stack assuming I/P seq 1, 2, 3, 4, 5 in order.

a) 3 4 5 1 2 X

b) 3 4 5 2 1 ✓

c) 1 5 2 3 4

d) 5 4 3 1 2

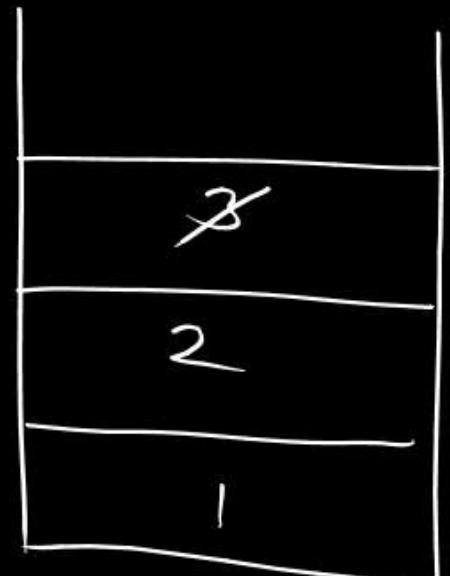
a) Push 1

Push 2

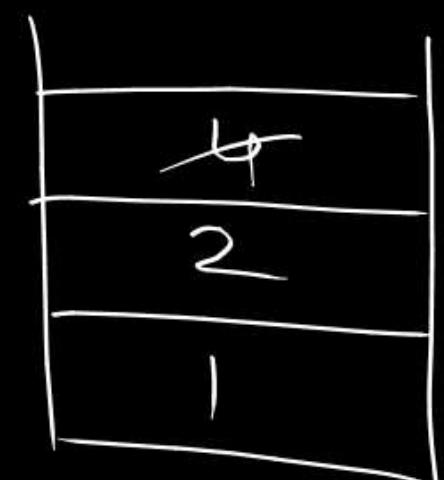
Push 3

c) Push 4

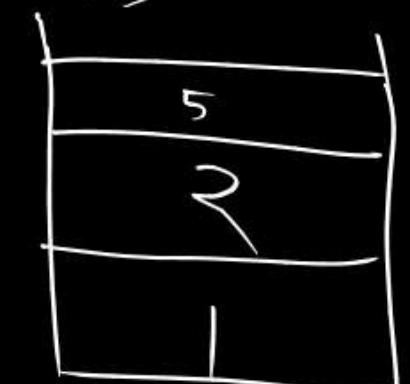
f) Pop()



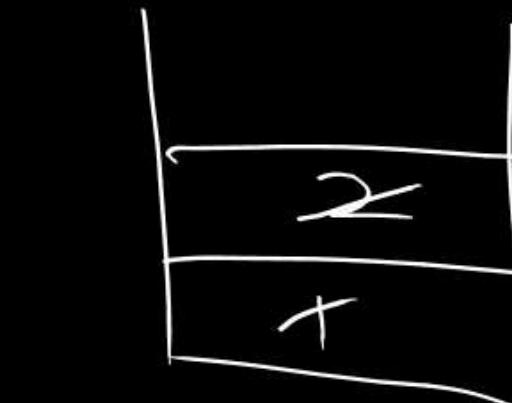
b) Pop()



d) Pop()



e) Push 5

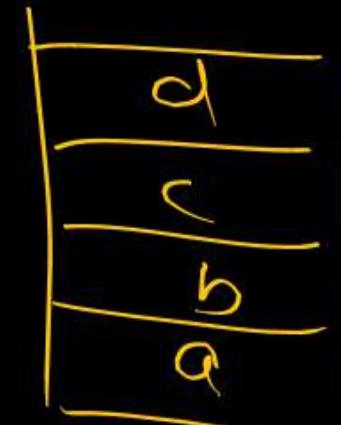


Pop()

Q A frog attempts to generate as many permutations as possible of the string "abcd" by pushing the character  $\xrightarrow{a,b,c,d}$  in same order onto a stack but it may pop off top char. at any time. Which one of the following string can not be generated using this frog?

- ~~a) abcd~~  $\rightarrow$  Possible  
~~b) dcba~~  
c) cbad  
d) cabd
- Push a  
Pop  
Push b  
Pop  
Push c  
Pop  
Push d  
Pop

- b) Possible  
Push a  
Push b  
Push c  
Push d



Pop  
Pop  
Pop  
Pop

Q A prog. attempts to generate as many permutations as possible of the string "abcd" by pushing the character  $\xrightarrow{a,b,c,d}$  in same order onto a stack but it may pop off top char. at any time. Which one of the following string can not be generated using this prog.?

~~a) abcd~~

~~b) dcba~~

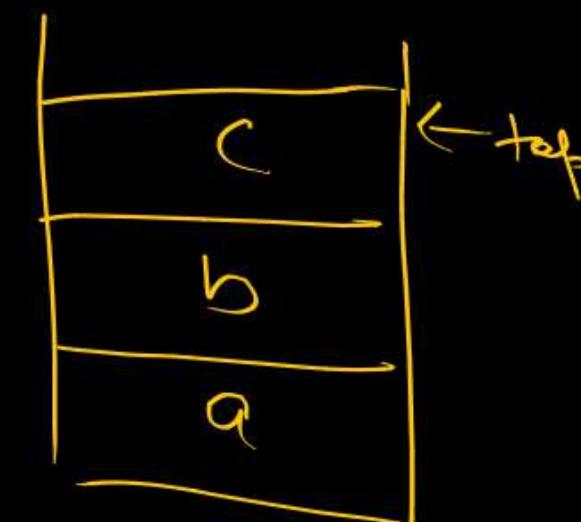
~~c) cbad~~ <sup>Possible</sup>

~~d) cabd~~

Push a

Push b

Push c



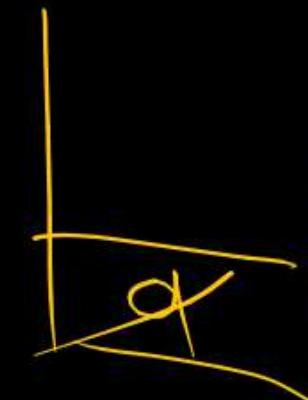
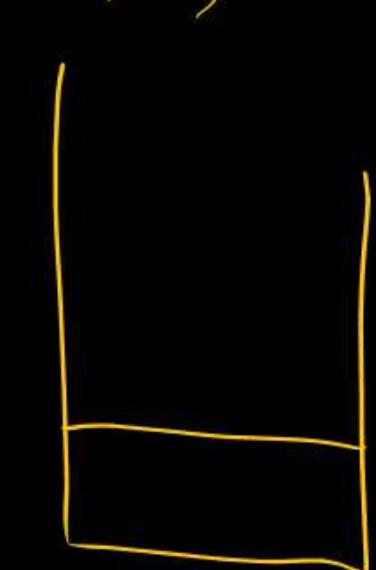
Pop()

Pop()

Pop()

Push d

Pop



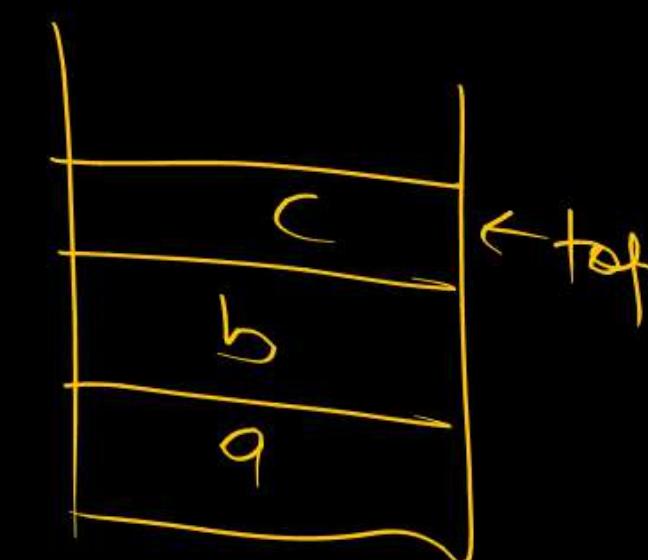
Q A prog. attempts to generate as many permutations as possible of the string "abcd" by pushing the character  $\xrightarrow{a,b,c,d}$  in same order onto a stack but it may pop off top char. at any time. Which one of the following string can not be generated using this prog.?

- ~~a) abcd~~
- ~~b) dcba~~
- ~~c) cbad~~
- ~~d) cabd~~
- e) abdc

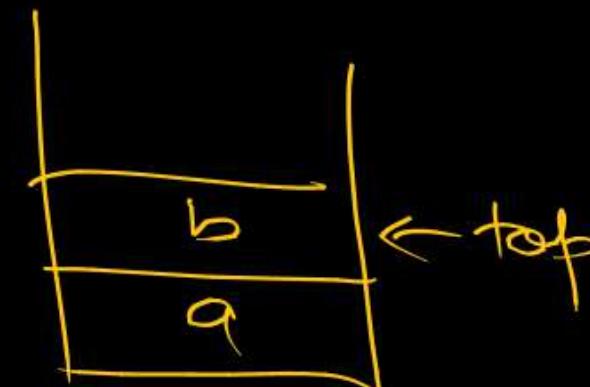
Push q

Push b

Push c



Pop()



Q The foll. postfix exp with single digit operand is eval using stack.

Ans<sup>exp0</sup> 8 2 3 ^ / 2 3 X + 5 1 X -

top 2 ele. of the stack after first X is eval.

- a) 6 1 ✓
- b) 5 7
- c) 3 2
- d) 1 5

→ [ t me / PW|pankajsirP ]

Hashing → 3-4  
videos

All the best



Q The foll. postfix exp with single digit operand is eval. using stack.

8 2 3 ^ / 2 3 x + 5 1 x -

^

order of Pop

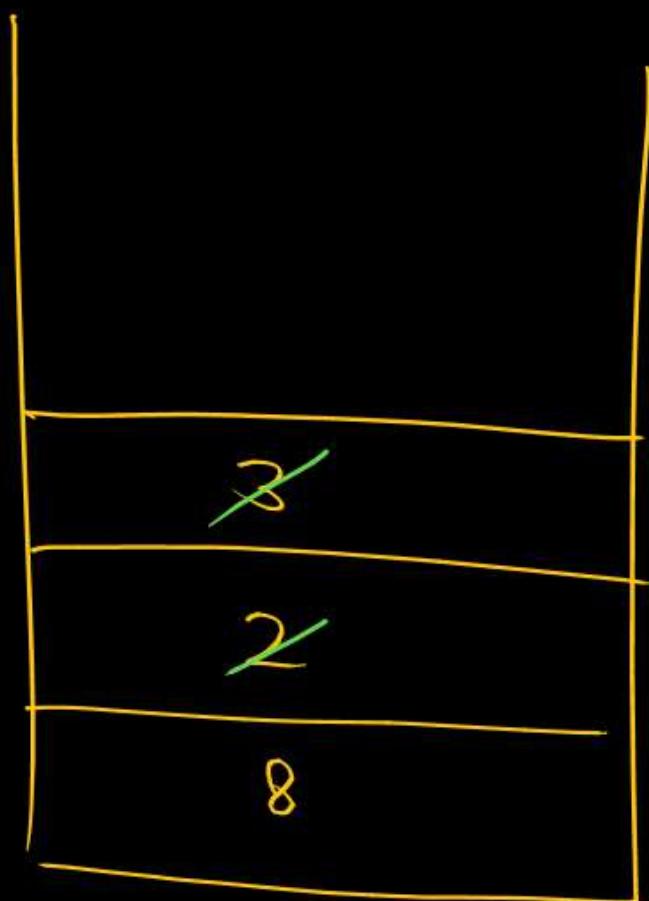
3 2



$2^3$

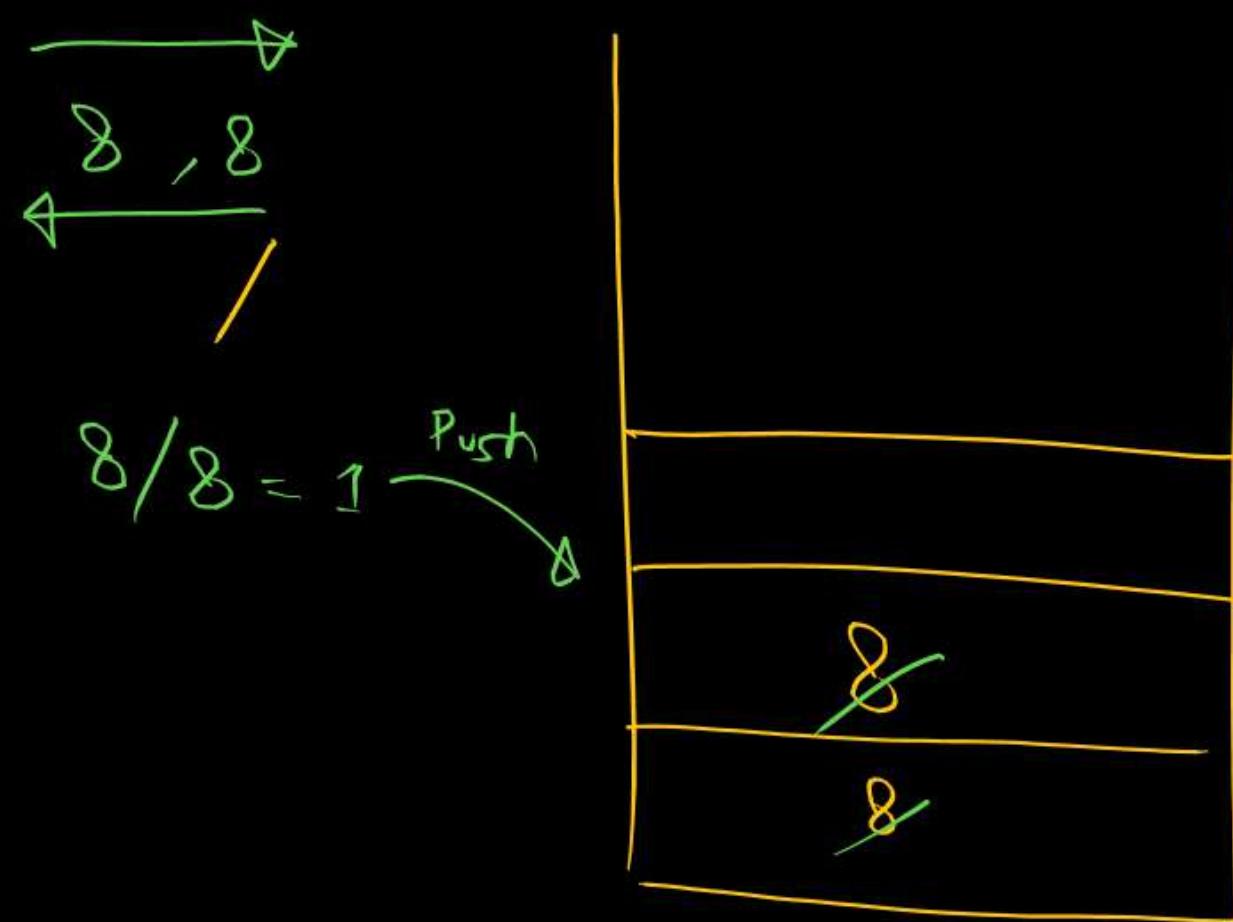
$\Rightarrow 8$

Push



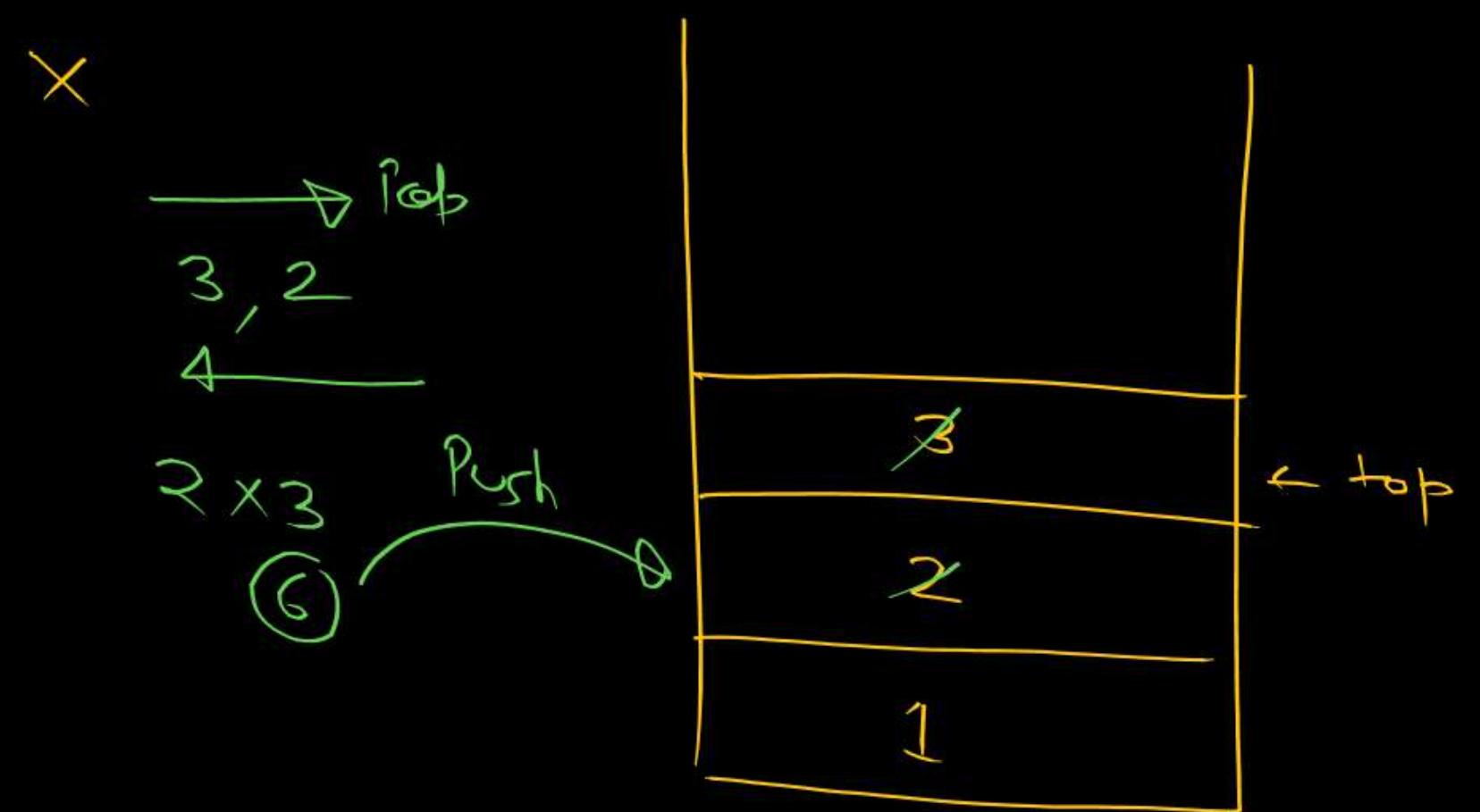
Q The foll. postfix exp with single digit operand is eval. using stack.

8 2 3 ^ / 2 3 X + 5 1 X -



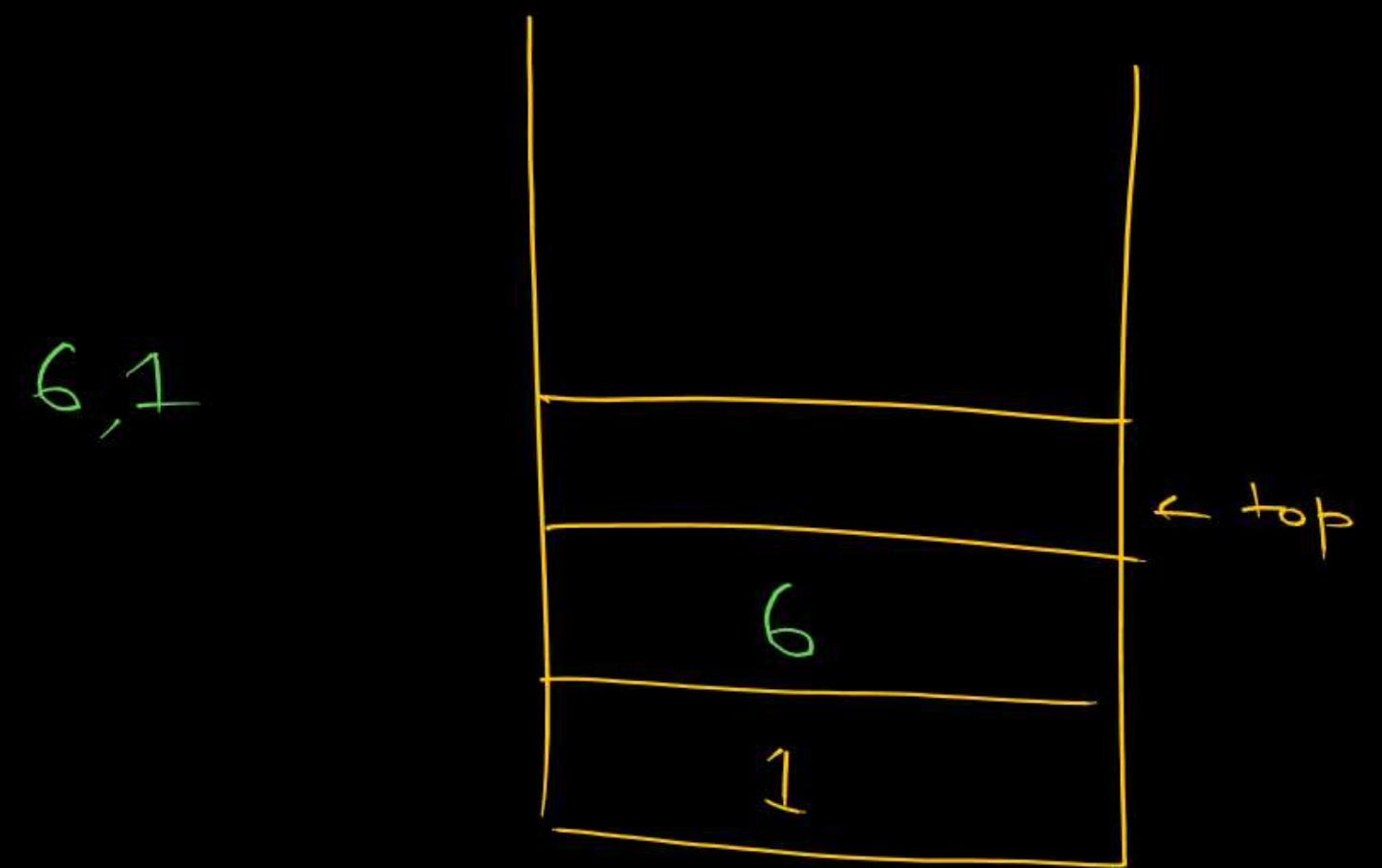
Q The foll. postfix exp with single digit operand is eval. using stack.

8 2 3 ^ / 2 3 X + 5 1 X -



Q The foll. postfix exp with single digit operand is eval. using stack.

8 2 3 ^ / 2 3 x + 5 1 x -





## 2 mins Summary

ISRO PYQ SERIES

P  
W

**Topic** One

**Topic** Two

**Topic** Three

**Topic** Four

**Topic** Five

# THANK - YOU