



# CS & IT ENGINEERING

## Data Structure & Programming

1500 Series

Lecture No.- 04

By- Pankaj Sharma Sir



# Recap of Previous Lecture



Topic

Problem Practice Part-03





# Topics to be Covered



Topic

Problem Practice Part-04



## [MCQ]



$x_{static}$  1 2 3

#Q. The integer value printed by the ANSI-C program given below is:

```
#include <stdio.h>
```

```
int func(){
```

```
static int x = 1;
```

```
    ++x;
```

```
    return x;
```

```
}
```

```
int main () {
```

```
    int x, y;
```

```
    x = func ();
```

```
    if ((func () - 3) || (x--)) {
```

```
        printf ("%d", x);
```

```
    return 0;
```

```
}
```

$if((3-3) || (x--))$

$if(0 || (x--))$

$if(1)$

$x_{main}$

1

1

**A**

4

**B**

3

**C**

2

**D**

1



[NAT]



#Q. The integer value printed by ANSI-C program given below is\_\_\_\_\_.

```
#include <stdio.h>
```

```
int main () {
```

```
    char exam [] = "GATE 2\0 24";
```

```
    char organising [] = "IISc\0\0";
```

```
    int len, size;
```

⑩ len = strlen(exam) + strlen(organising);

⑪ size = sizeof(exam) + sizeof(organising);

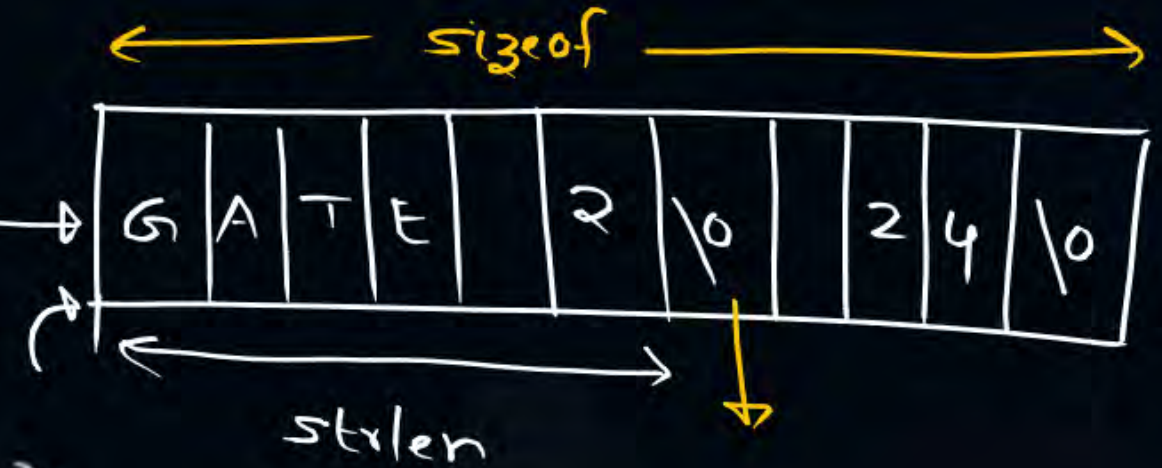
```
    printf("%d", size - len);
```

```
    return 0;
```

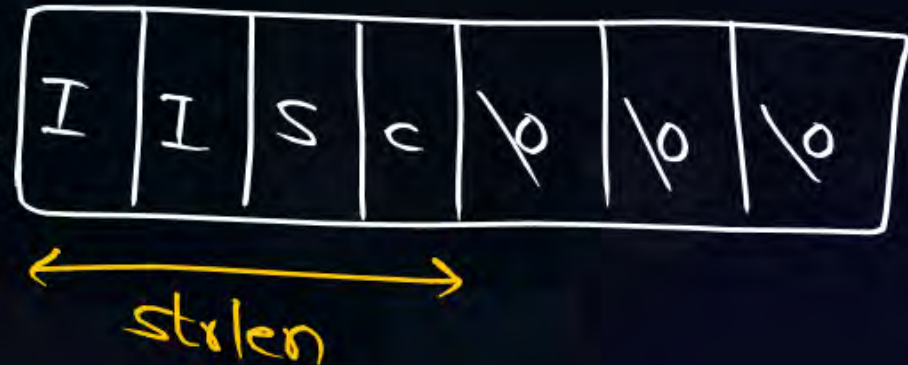
```
}
```

space space

exam



organising



8

10  
[strlen() sizeof()]  
Gate



[NAT]



#Q. The output of the 'C' program snippet is \_\_\_\_.

```
int main () {
```

```
    char str [] = "GATE 2024";
```

```
    char *ptr = str;
```

```
    printf ("%d", (int) strlen (str + 1[ptr] - ptr [8] - 9 ));
```

```
    return 0;
```

```
}
```

str



ptr

ptr+1

ptr+8

str + 'A' - '4' - 9

Ascii

A → 65

0 → 48

1  
2  
3  
4  
5 → 52

↓  
\*(1 + ptr)

↓  
\*(ptr + 1)

↓  
ptr[1]

str + 65 - 52 - 9

str + 13 - 9

str + 4

[NAT]



#Q. The output of the 'C' program snippet is \_\_\_\_.

```
int main () {
```

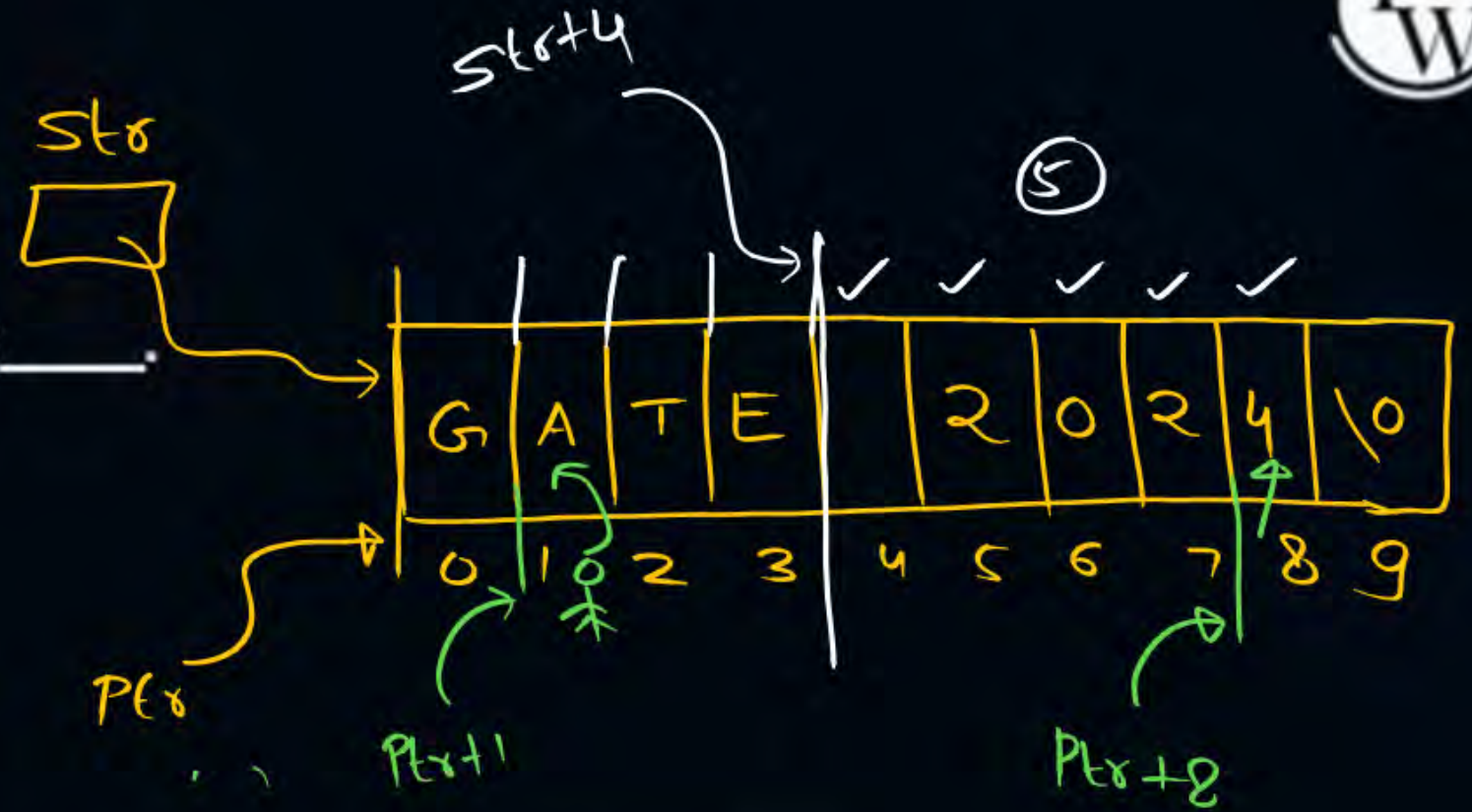
```
    char str [] = "GATE 2024";
```

```
    char *ptr = str;
```

```
    printf ("%d", (int) strlen (str + 1[ptr] - ptr [8] - 9 ));
```

```
    return 0;
```

```
}
```



(5)

$str + 65 - 52 - 9$

$str + 13 - 9$

$str + 4$

strlen( )



```
#Q. #include <stdio.h>
int fun (static int x) {
    static int y;
    x ++;
    y ++;
    return x + y ;
}
int main ( ) {
    printf ("%d%d", fun (5), fun(5));
    return 0;
}
output of the program is -
```

- A** Garbage value
- B** 7 8
- C** 7 9
- ☒ **D** Compilation error.



#Q. #include <stdio.h>

int main () {

register int x;

x = 5;

switch(size of (x)) {

Case 1 : printf ("1");

Case 2 : printf ("2");

Case 3 : printf ("3");

l: Case 4 : printf ("4");

Case 8 : printf ("8");

default : printf ("%d", size of (x));

Case 5 : if (x == 5) goto l;

}

}

int  $\Rightarrow$  4 byte

output of the program is \_\_\_\_.

(If the code has compilation error answer is 1, if segmentation fault then answer is 0.)

if (5 == 5)

$\rightarrow$  x 54

484484

## [MCQ]



#Q. #include <stdio.h>

```
int fun1() {  
    static int x = 5;  
    printf ("%d", x);  
    x--;  
    return x && fun1();  
}  
main () {  
    fun1 ();  
}
```

Output of program is-

short  
circuit



Compilation error



5 4 3 2 1



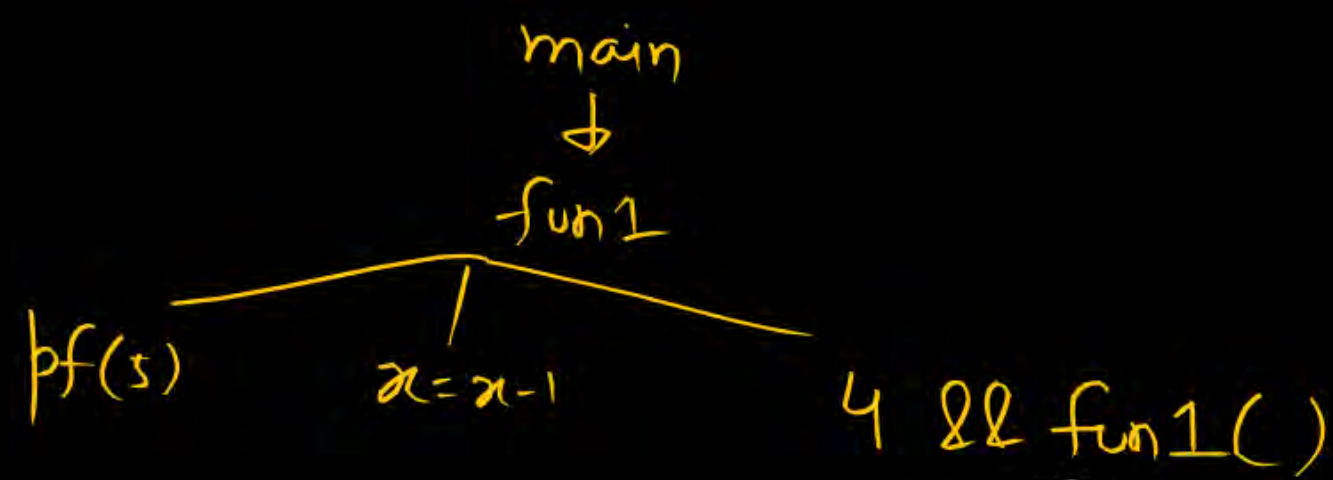
Stack overflow (since no base case)



5 4 3 2 1 0

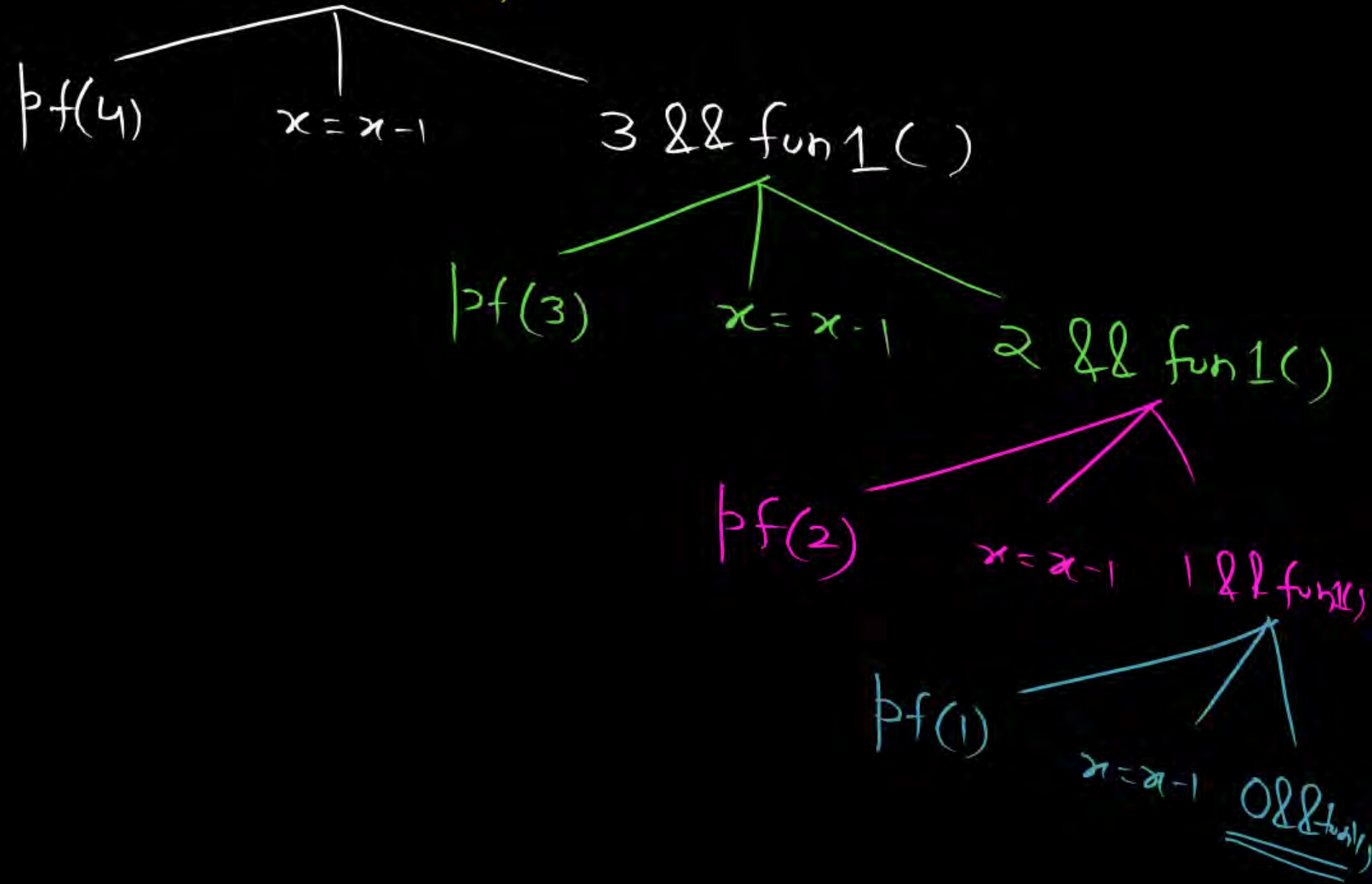


x 8 4 3 2 1 0



5

5 4 3 2 1



## [MCQ]



#Q. #include <stdio.h>

```
int fun (int x);
```

```
int f1 (int x) {
```

```
    return x && f1(x-1);
```

```
}
```

```
int fun (int x) {
```

```
    if (x == 0) return 1 ;
```

```
    return fun (x >> f1 (x))
```

```
}
```

What is the output of fun (5)?

$f1(5) \Rightarrow 0$

$fun(5)$

$fun(5 \gg f1(5))$

**A**

Infinite loop

**B**

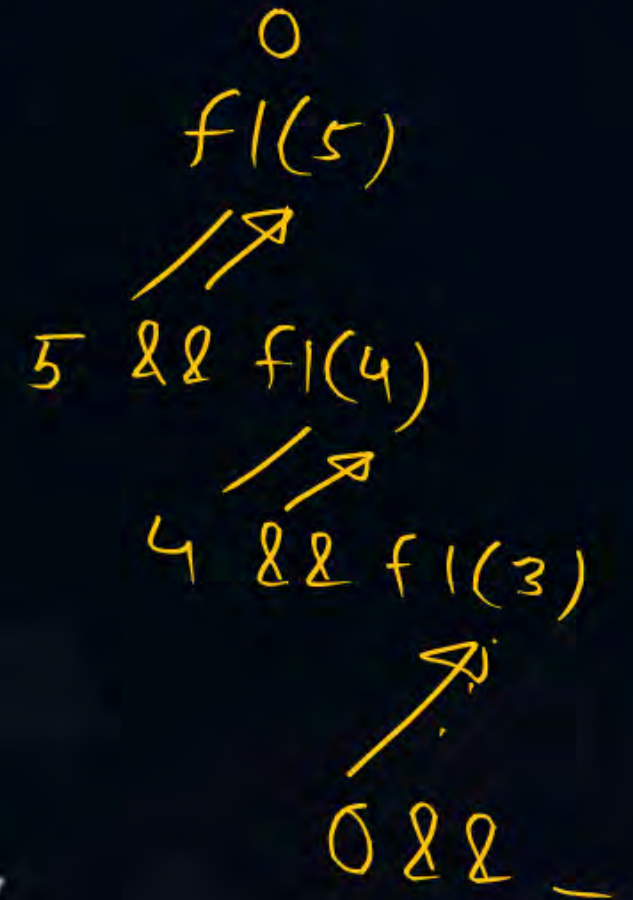
Stack over flow

**C**

0

**D**

2





## [MCQ]

#Q. #include <stdio.h>

```
int fun (int x);
```

```
int f1 (int x) {
```

```
    return x && f1(x-1);
```

```
}
```

```
int fun (int x) {
```

```
    if (x == 0) return 1 ;
```

```
    return fun (x >> f1 (x))
```

```
}
```

What is the output of fun (5)?

$f_1(5) \Rightarrow 0$

$fun(5)$   
|  
 $fun(5 >> f_1(5))$

$fun(5)$   
|  
 $fun(5)$   
|  
 $fun(5)$



**A**

Infinite loop

☒ **B**

Stack over flow

**C**

0

**D**

2



[NAT]



#Q. # include <stdio.h>

```
int main () {
```

```
    int x = 0617;
```

```
    char *p = &x;
```

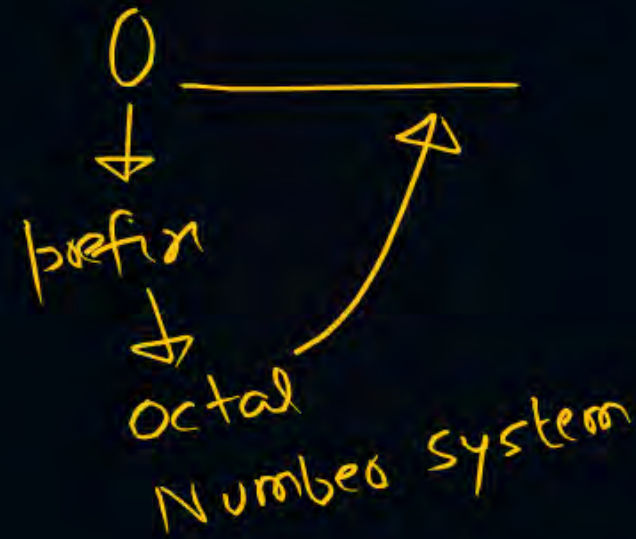
```
    int ans = size of (p) + x;
```

```
    printf("%d", ans);
```

```
    return 0;
```

```
}
```

The output of above code is \_\_\_\_\_ (consider 64 bit system).



$$x = 0617$$

$$x = (617)_8$$

Pointer size = 8 byte

$$617 \\ 8^2 8^1 8^0$$

$$(399)$$

$$\Rightarrow 6 \times 8^2 + 1 \times 8^1 + 7 \times 8^0 \\ \Rightarrow 6 \times 64 + 8 + 7 \\ 384 + 15$$



[NAT]



#Q. # include <stdio.h>

```
int main ( ) {
```

```
    int x = 0617;
```

```
    char *p = &x;
```

```
    int ans = size of (p) + x;
```

```
    printf("%d", ans);
```

```
    return 0;
```

```
}
```



$$ans = 8 + 399$$

407

The output of above code is \_\_\_\_\_(consider 64 bit system).

## [MCQ]



sizeof op  $\rightarrow$  compile time  
 $\rightarrow$  No Eval.

#Q. #include<stdio.h>

```
int main () {
```

```
int a = 5 ;
```

```
char* b=&a;
```

```
printf ("%d%d, size of (++*b), a);
```

```
return 0;
```

```
}
```

Which of the following is the output of above code.

++\*b  
 $\downarrow$   
sizeof(\*b)  
sizeof(char)  
 $\downarrow$   
1

~~A~~ 45

C 15

~~B~~ 46

D 16



[NAT]



Gate Prg

-5

#Q. #include <stdio.h>

```
int main ( ) {
```

```
int arr [ ] = {5,6, 7, 8, 9, 11, 12, 13,};
```

```
int sum = 0, *p = arr + 5;
```

```
for (int i = 0; i < 6; i++) {
```

```
    sum = sum + * (p-i) - (*p -i) ;
```

```
}
```

```
printf ("%d," sum);
```

```
    return 0;
```

```
}
```

The output is\_\_\_\_\_.

[NAT]



#Q. # include <stdio.h>

int main () {

int arr [ ] = {1, 2, 3, 4, 5, 10, 11, 12, 13, 14, 21, 22, 23, 24, 25};

int \* p = &arr [1] + 9;

printf ("%d", p [1]);

return 0;

}

The output is 22.

~~21~~  
22

↑ ↑  
P (P+1)

P+1 ⇒ add. of 22

\* (P+1) ⇒ 22

P[1]



Tomm.



Saturday / Sunday / Monday



①



①



①



## 2 mins Summary



Topic

One -

Topic

Two -

Topic

Three

Topic

Four

Topic

Five





**THANK - YOU**