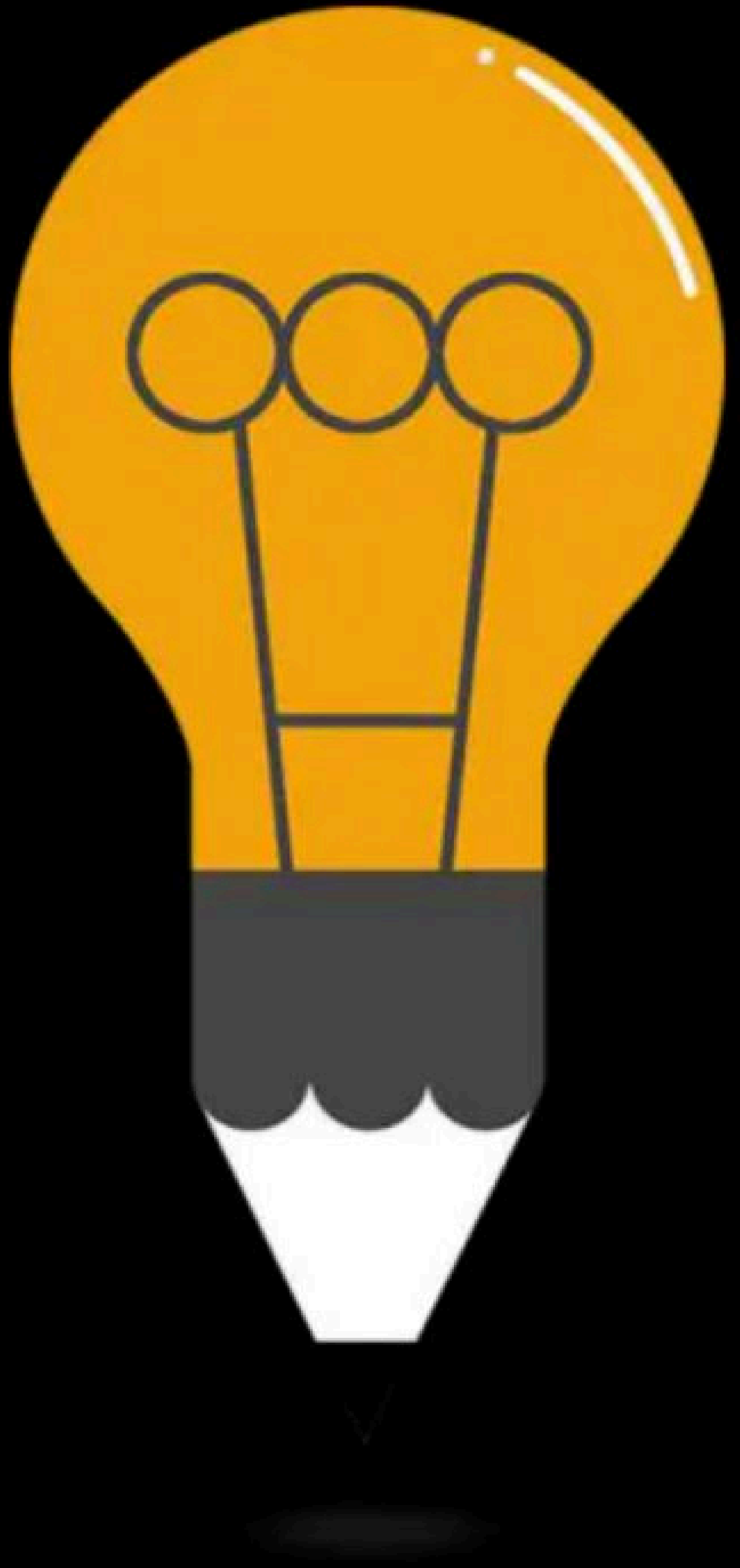
Doubt Solving Session

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

Vishvadeep Gothi • Lesson 17 • Dec 15, 2022



Doubts & Miscellaneous

By: Vishvadeep Gothi

GATE-2015

Question

Consider the following C function:

fun

```
int f(int n){
int x=1, k;
if(n==1) return x;
for(k=1; k<n; k++)
x=x+fun(k)*fun(n-k);
return x;
}
```

The return value of fun(5) is ____________?

Ans = 51

$n=5$

$k=1 \quad \overset{16}{x} = \overset{1}{x} + \overset{1}{fun(1)} * \overset{15}{f(4)}$

$k=2 \quad \overset{26}{x} = \overset{16}{x} + \overset{2}{fun(2)} * \overset{5}{f(3)}$

$k=3 \quad \overset{x}{x} = \overset{26}{x} + fun(\overset{5}{3}) * \overset{2}{f(2)}$

$k=4 \quad \overset{51}{x} = \overset{36}{x} + fun(\overset{15}{4}) * \overset{1}{f(1)}$

n	1	2	3	4	5
fun(n)	1	2	5	15	51

$n = 1$

$fun(1) = 1$

$n=2$

$$x = \underset{1}{x} + \underset{1}{fun(1)} * \underset{1}{fun(1)}$$

$x = 2$

$n=3$

$$\overset{3}{x} = \overset{1}{x} + \overset{1}{fun(1)} * \overset{2}{fun(2)}$$

$$\overset{5}{x} = \overset{3}{x} + \overset{2}{fun(2)} * \overset{1}{fun(1)}$$

$x = 5$

$n = 4$

$$\overset{6}{x} = \overset{1}{x} + \overset{1}{fun(1)} * \overset{5}{fun(3)}$$

$$\overset{10}{x} = \overset{6}{x} + \overset{2}{fun(2)} * \overset{2}{fun(2)}$$

$$\overset{15}{x} = \overset{10}{x} + \overset{5}{fun(3)} * \overset{1}{fun(1)}$$

$x = 15$

$$fun(n) = \begin{cases} 1 & \text{if } n = 1 \\ 1 + \sum_{k=1}^{n-1} fun(k) * fun(n-k) & \text{if } n > 1 \end{cases}$$

$$fun(2) = 1 + fun(1) * fun(1)$$

$$f(3) = 1 + f(1) * f(2) + f(2) * f(1)$$

Characteristics of Variables

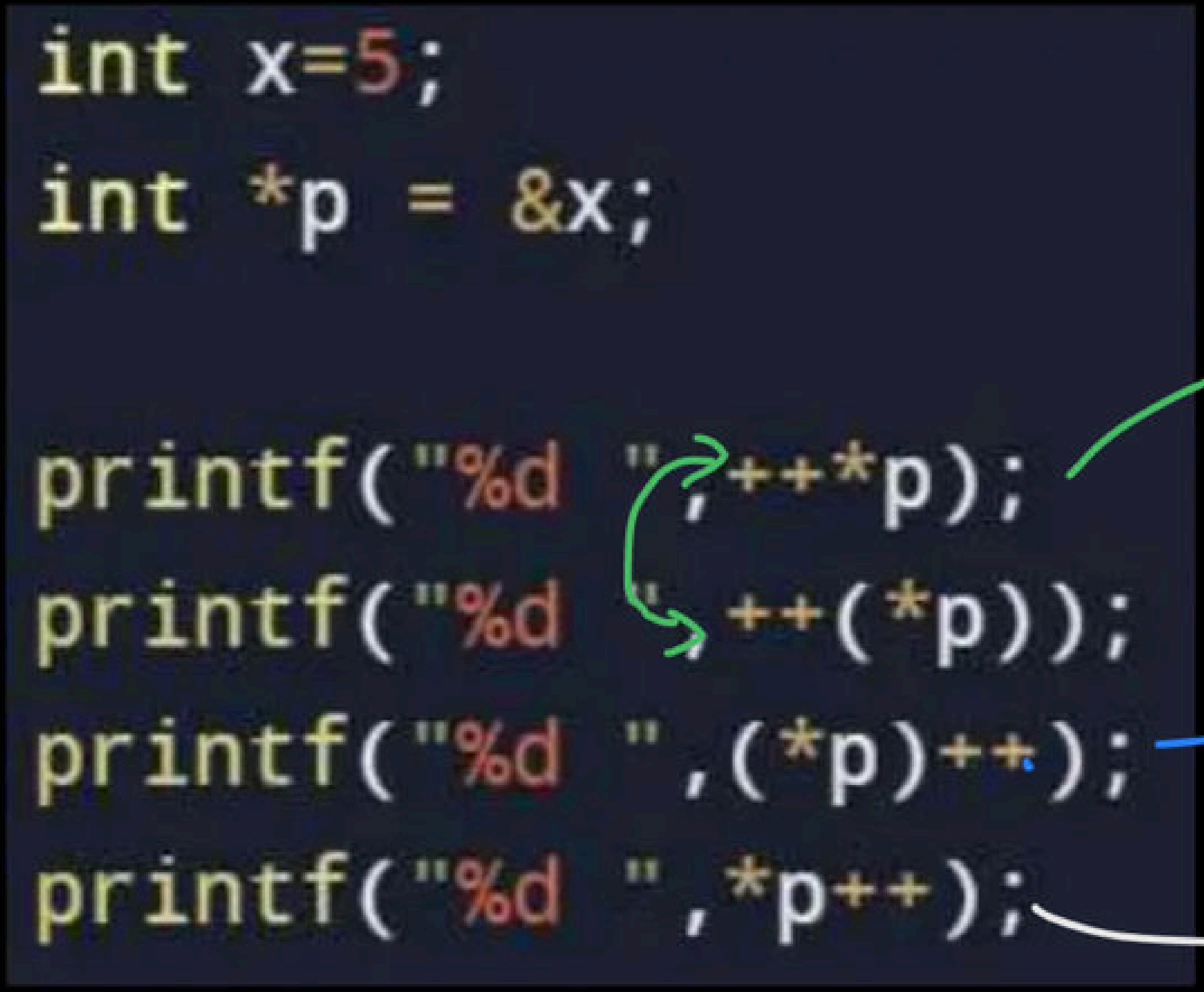
1. Lifetime
2. Scope
3. Initialization
4. Location

Storage Classes

1. auto
2. register
3. static
4. extern

Pointers

Right to left



```
int x=5;
int *p = &x;

printf("%d ",++*p);
printf("%d ",++(*p));
printf("%d ",(*p)++);
printf("%d ",*p++);
```

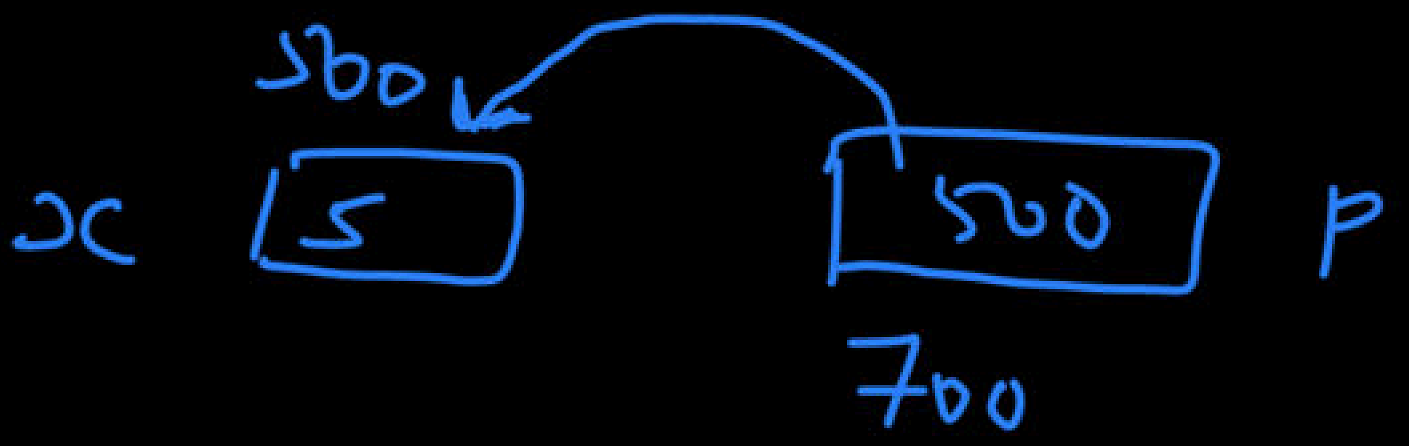
x => 6
*p => 6
p => 500

x => 6
*p => 6
p => 500

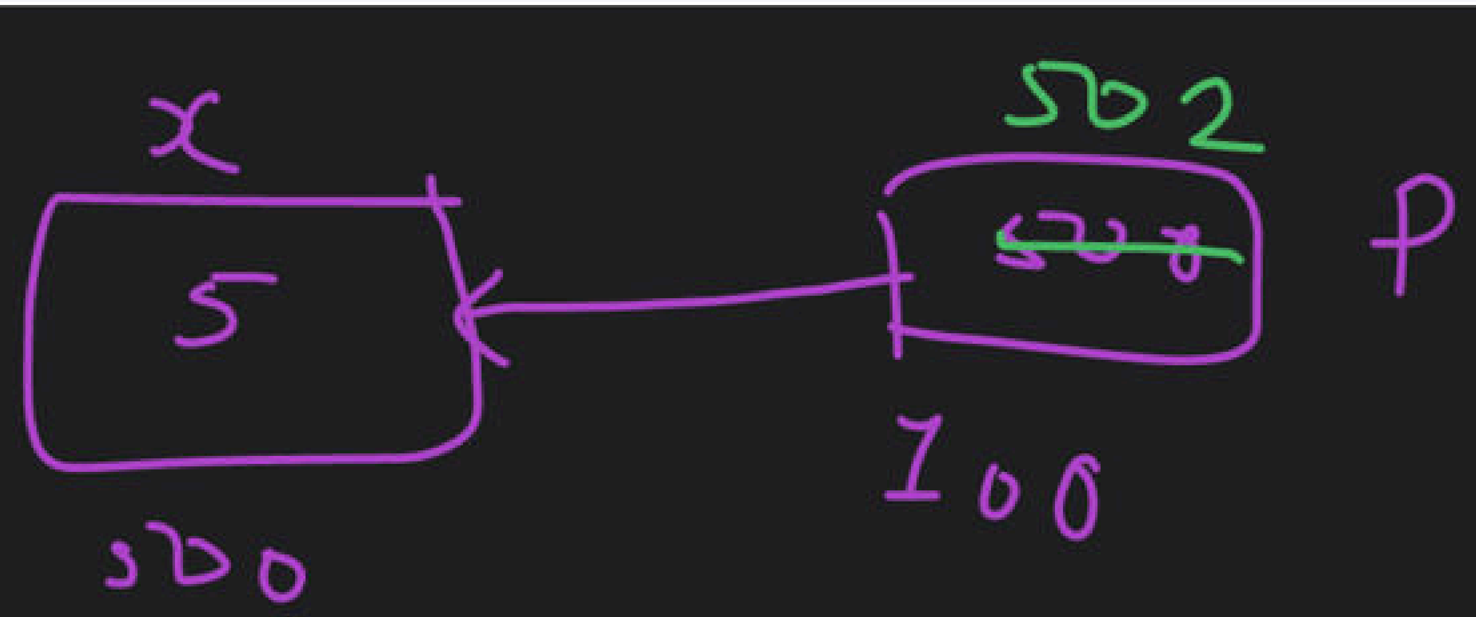
increment in value of p

++ *p
++(*p)
*p = *p + 1

all same



```
int x = 5;
int *p = &x;
```



```
printf("%d", *p++);   5
```

prints value pointed by p.

after that p++ happens

*p => garbage

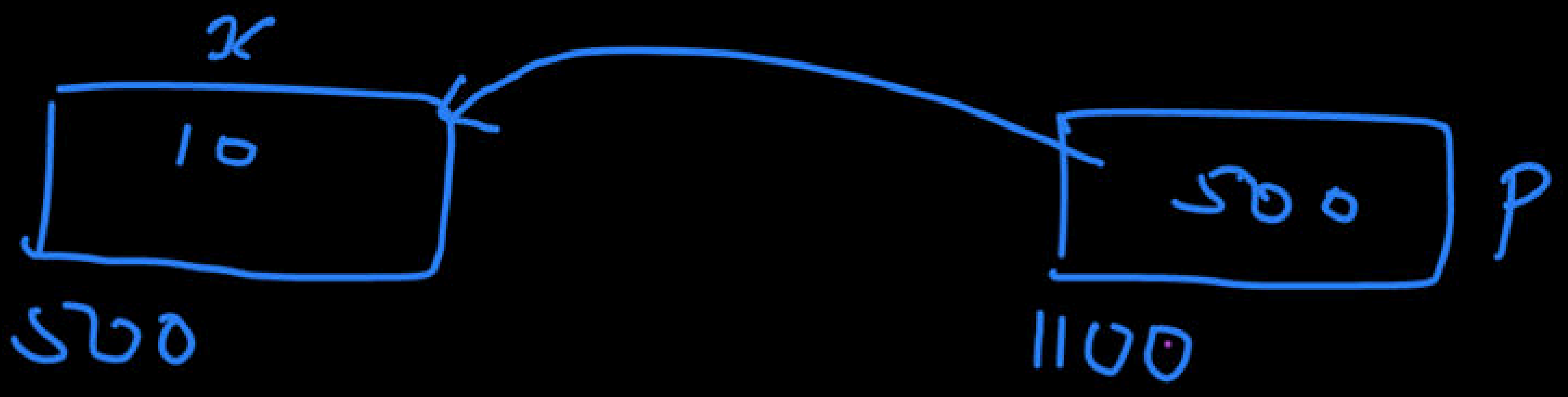
x => 5

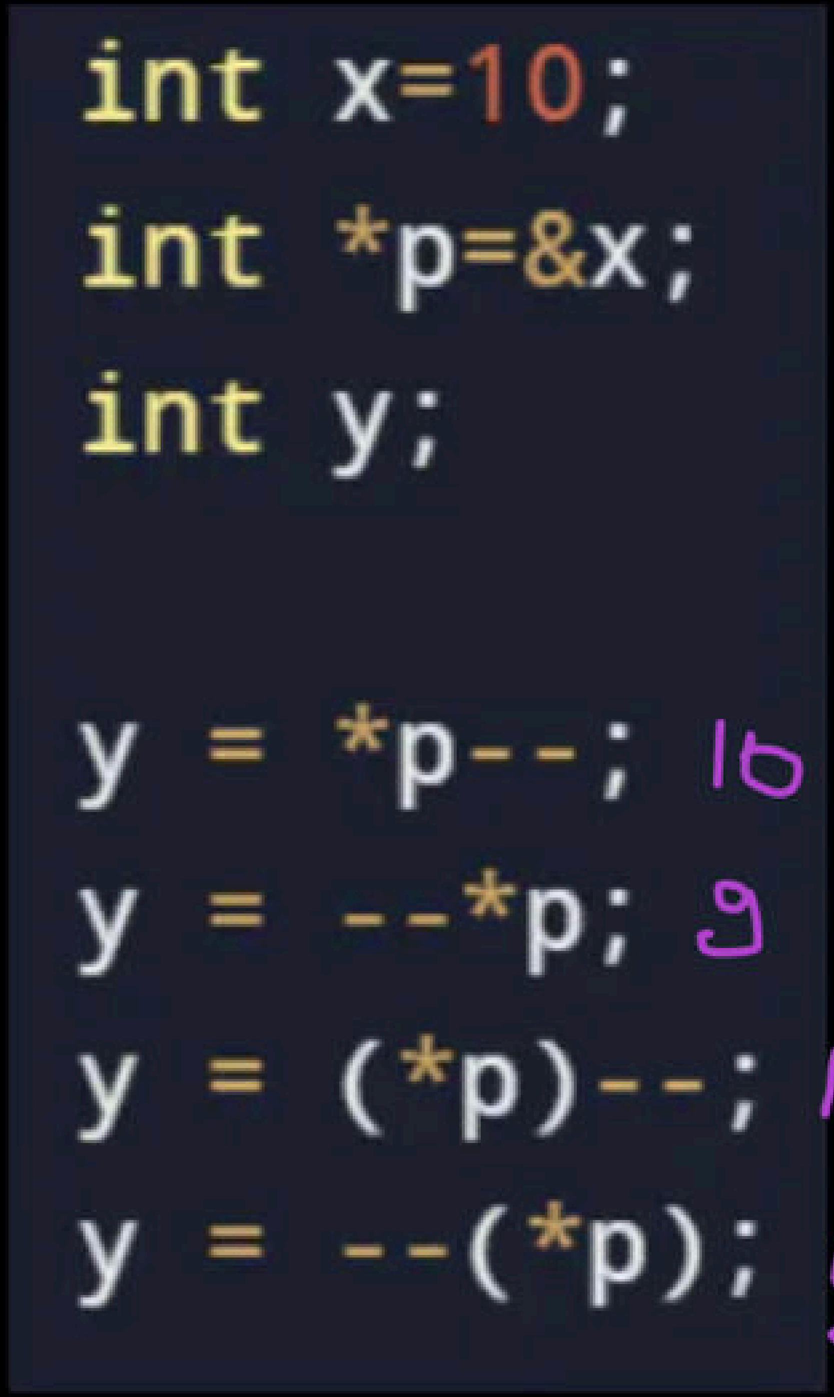
p => 502

Question

Calculate value of y, for each of the following individual case.

Assume address of x is 500 and address of p is 1100.





```
int x=10;
int *p=&x;
int y;
```

`y = *p--;` 10 and then p becomes 498

`y = --*p;` 9 and x = 9

`y = (*p)--;` 10 and x = 9

`y = --(*p);` 9 and x = 9

NULL Pointer

`int *p;` ⟹ p points to a garbage value

```
printf("%d", *p);
```

if we want that p does not hold any add.

```
int *p = NULL;
```

p

String

string

```
char ch[] = {'H', 'E', 'L', 'L', 'O'};

char str[] = "HELLO";   ⇐ string

char *p = "HELLO";
```

ch □

0	1	2	3	4
'H'	'E'	'L'	'L'	'O'

str ↳

0	1	2	3	4	5
'H'	'E'	'L'	'L'	'O'	'\0'

*p ↑

↑ NULL Character

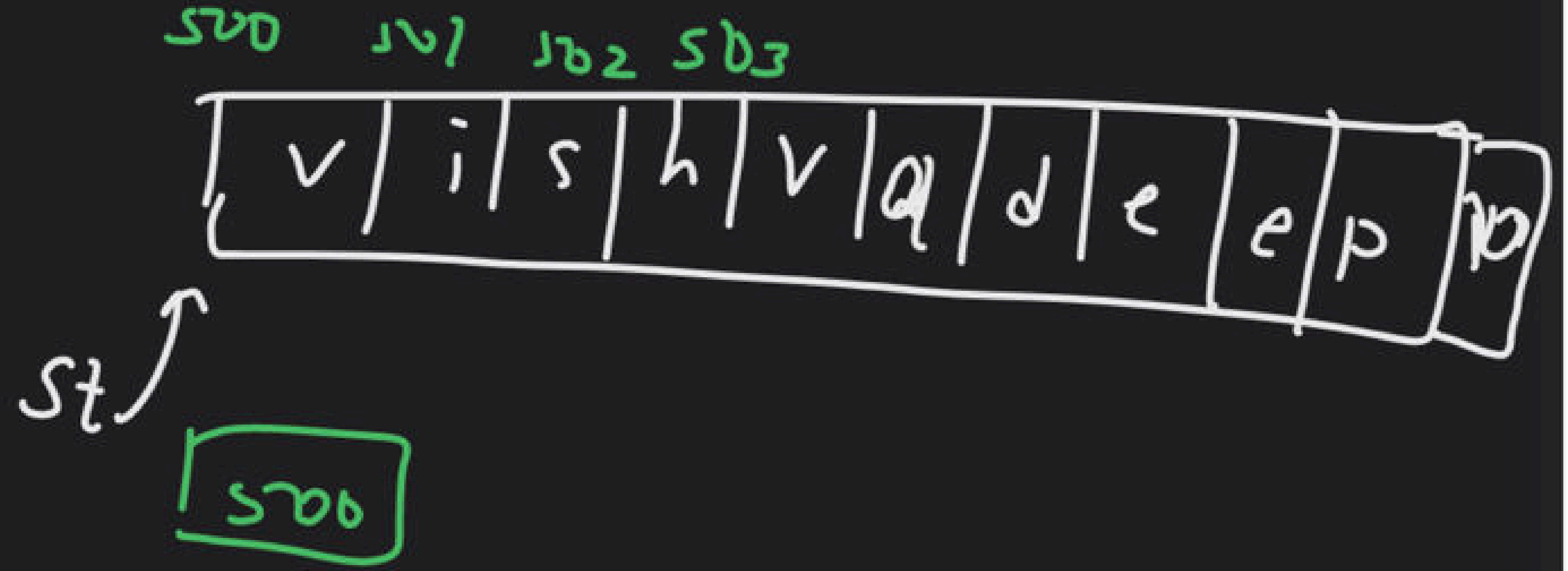
```
for(i=0; i<5; i++)
{ printf("%c", ch[i]);
}
                ↓
             *(ch+i)
```

```
printf("%s", str);

printf("%s", p);
```

with *s ⇒ pass starting address of string

```
char *st = "Vishvadeep";
printf("%s", st); Vishvadeep
printf("%s", st+2]; shvadeep
```



%s Prints string starting from given address, till first NULL character

```
char *st = "GATE 2024 Exam is \0 going to be \0 fun";
printf("%s", st);
```

output:-

⇓

GATE2024 Exam is

```
printf("%c", st+2);   => garbage value
```

add in (st+2) => ascII value

⇓ correct

```
printf("%c", *(st+2));   => T
```

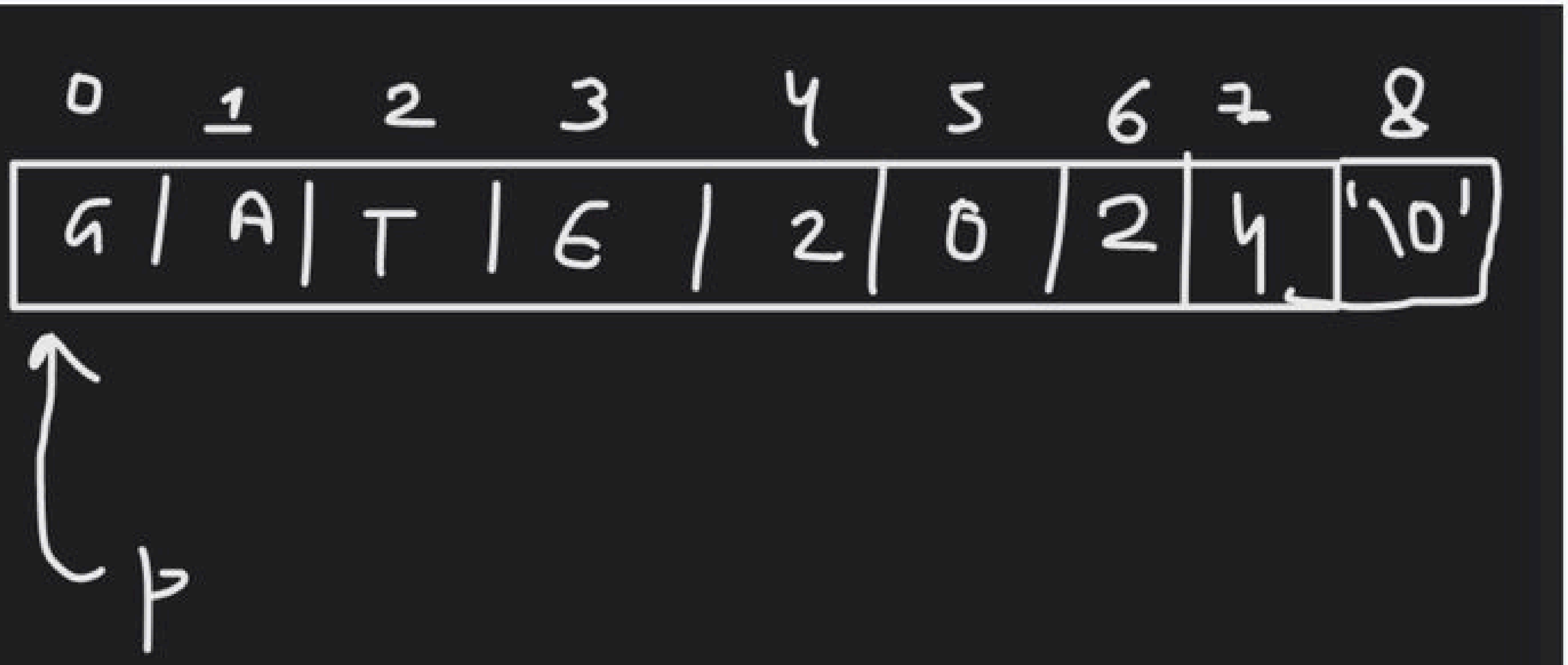
```
char A[] = "GATE2024";
char *p;
p = A;
printf("%s", p + p[3]-p[1]); => 2024
```

(E − A)

4

⇓

```
printf("%s", p+4);
```



0	1	2	3	4	5	6	7	8
G	A	T	E	2	0	2	4	'\0'

Literals and Constants

using const keyward

⇓

```
const type name = value;

const float pie = 3.14;
```

using macro

```
#define PIE 3.14

printf("%f", 2*PIE*r);
```

ex:-

```
#include <stdio.h>
#define PIE 3.14
void main()
{
  float r;
  scanf("%f",&r);
printf("area of circle is = %f", PIE * r*r);
}
```

=> preprocessor directive

```
#include <stdio.h>
#define f(a,b)  a*b
void main()
{ int x=5, y=3;
printf("%d", f(x,y));
printf("%d", f(x+y, x+y));
}
```

#define f(a,b) a*b → wrong way ⇒ correct way ⇓

#define f(a,b) (a)*(b)

⇓

(x+y)*(x+y)

f(x,y) → 5*3 ⇒ 15

f(x+y, x+y) → x+y * x+y ⇒ 23

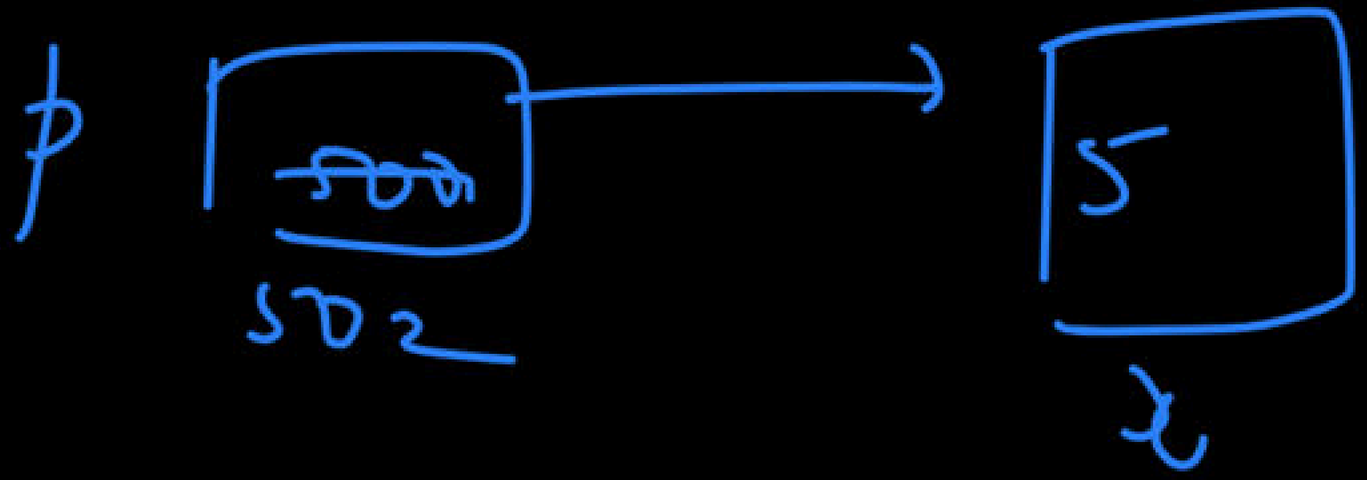
Happy Learning.!

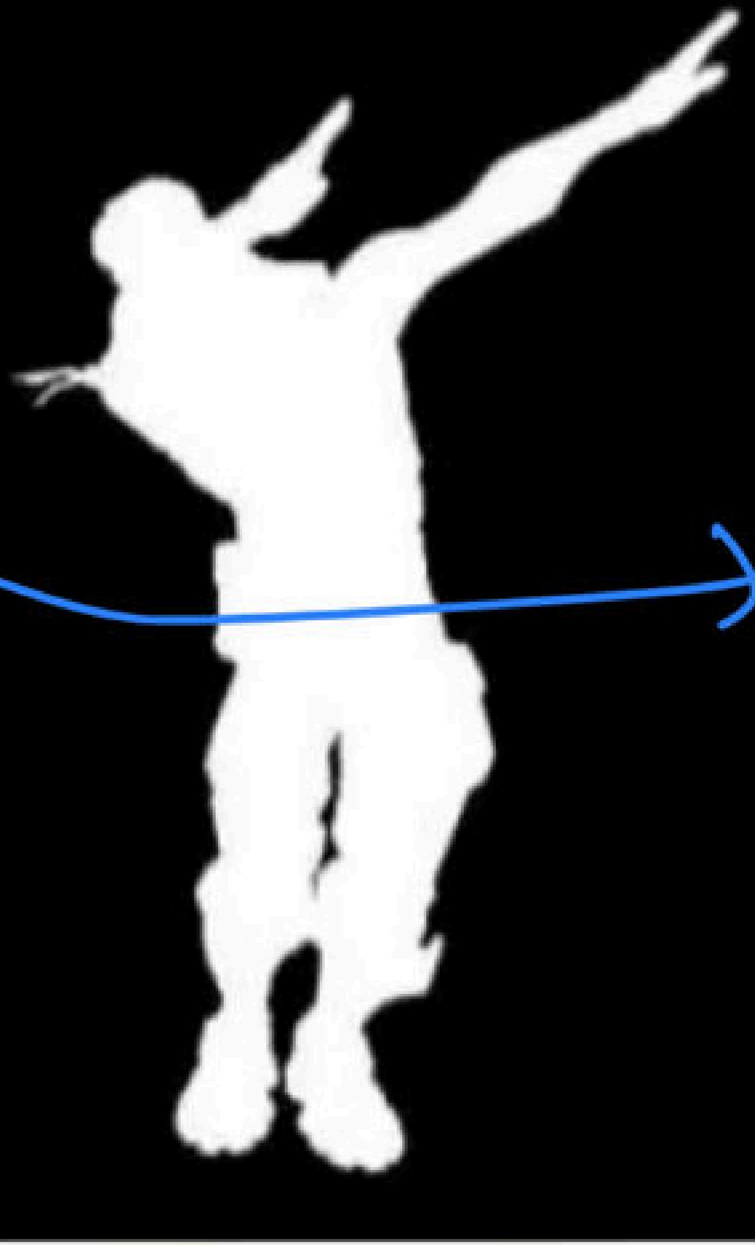
*++p

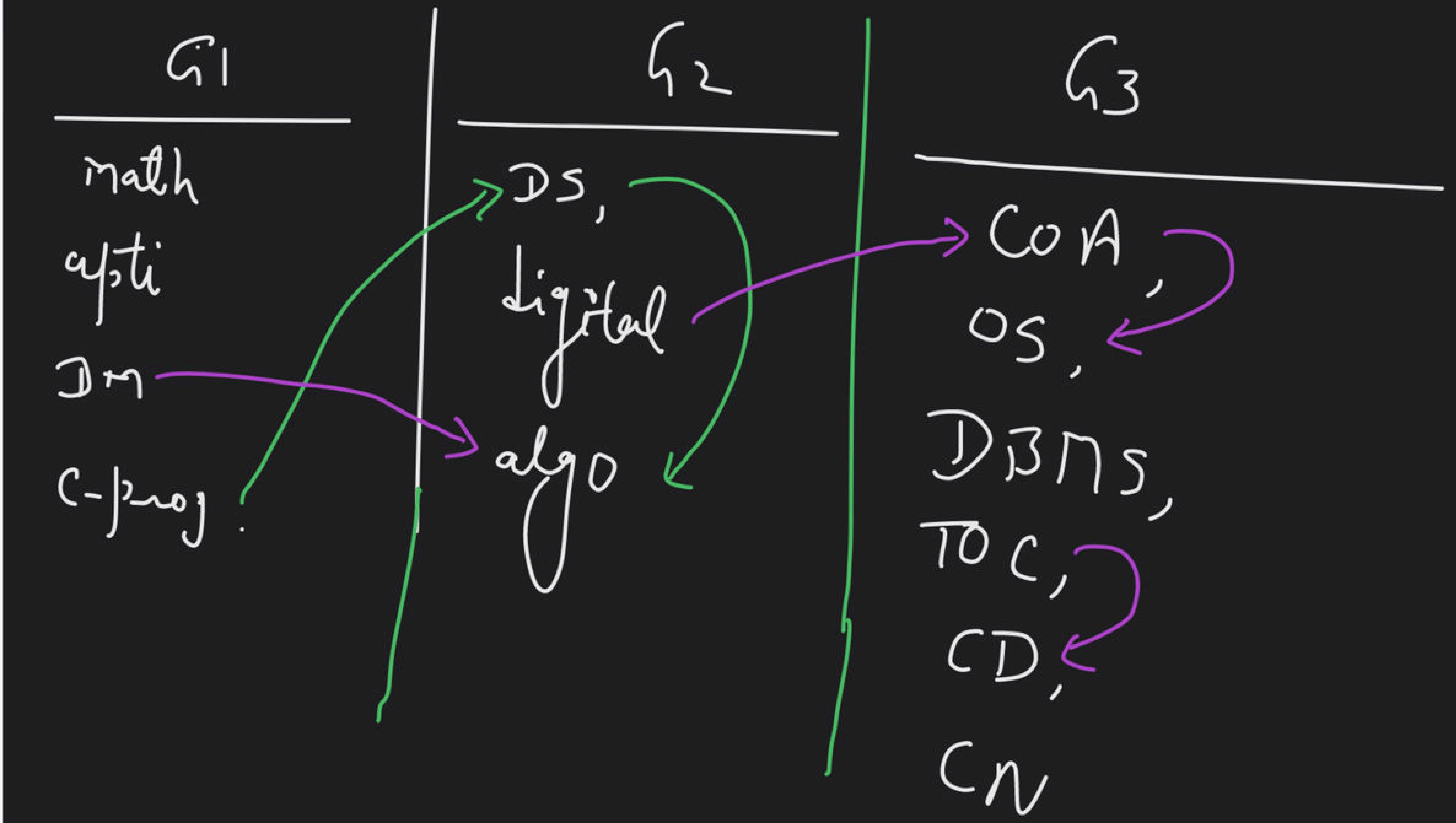
⇓

*(++p)

(*502) → garbage value







G1	G2	G3
math	DS,	COA,
apti	digital	OS,
DM	algo	DBMS,
C-prog.		TOC,
		CD,
		CN