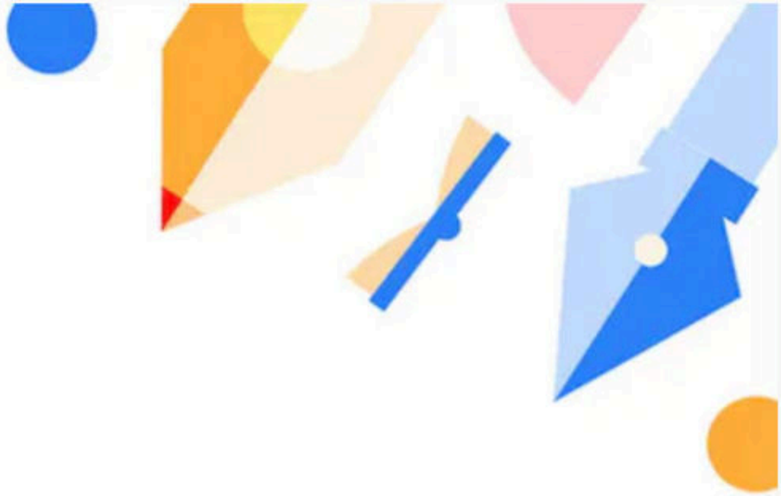
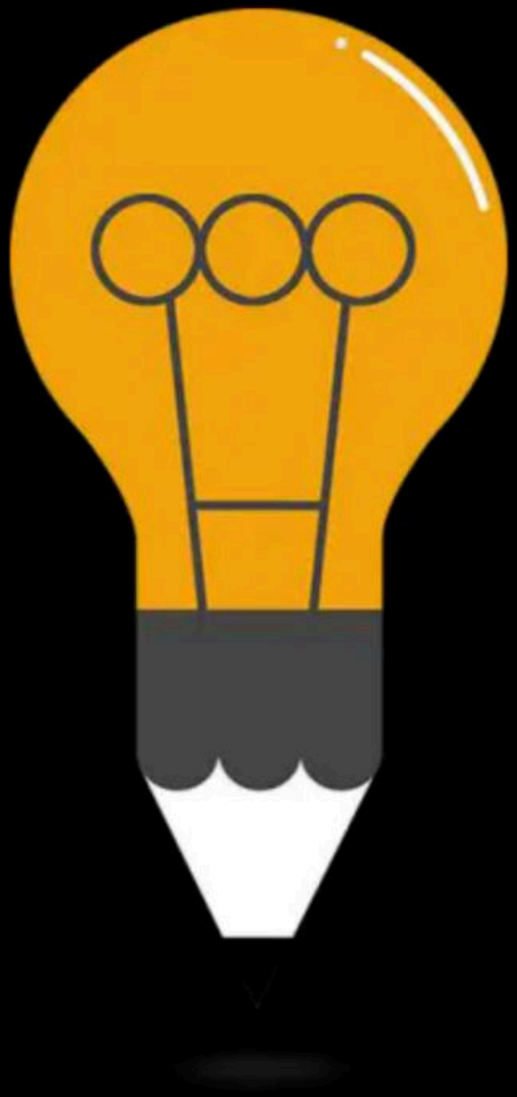


Array

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

Vishvadeep Gothi • Lesson 6 • Dec 1, 2022



Pointers & Array

By: Vishvadeep Gothi

floor() & ceil() functions => math.h

floor() → lower nearest int

ceil() → upper nearest int

floor(3.2) = 3

floor(3.9) = 3

floor(-3.6) = -4

ceil(4.6) = 5

ceil(4.2) = 5

ceil(4.1) = 5

ceil(4.01) = 5

ceil(-6.1) = -6

---

sqrt()

⇓

square root

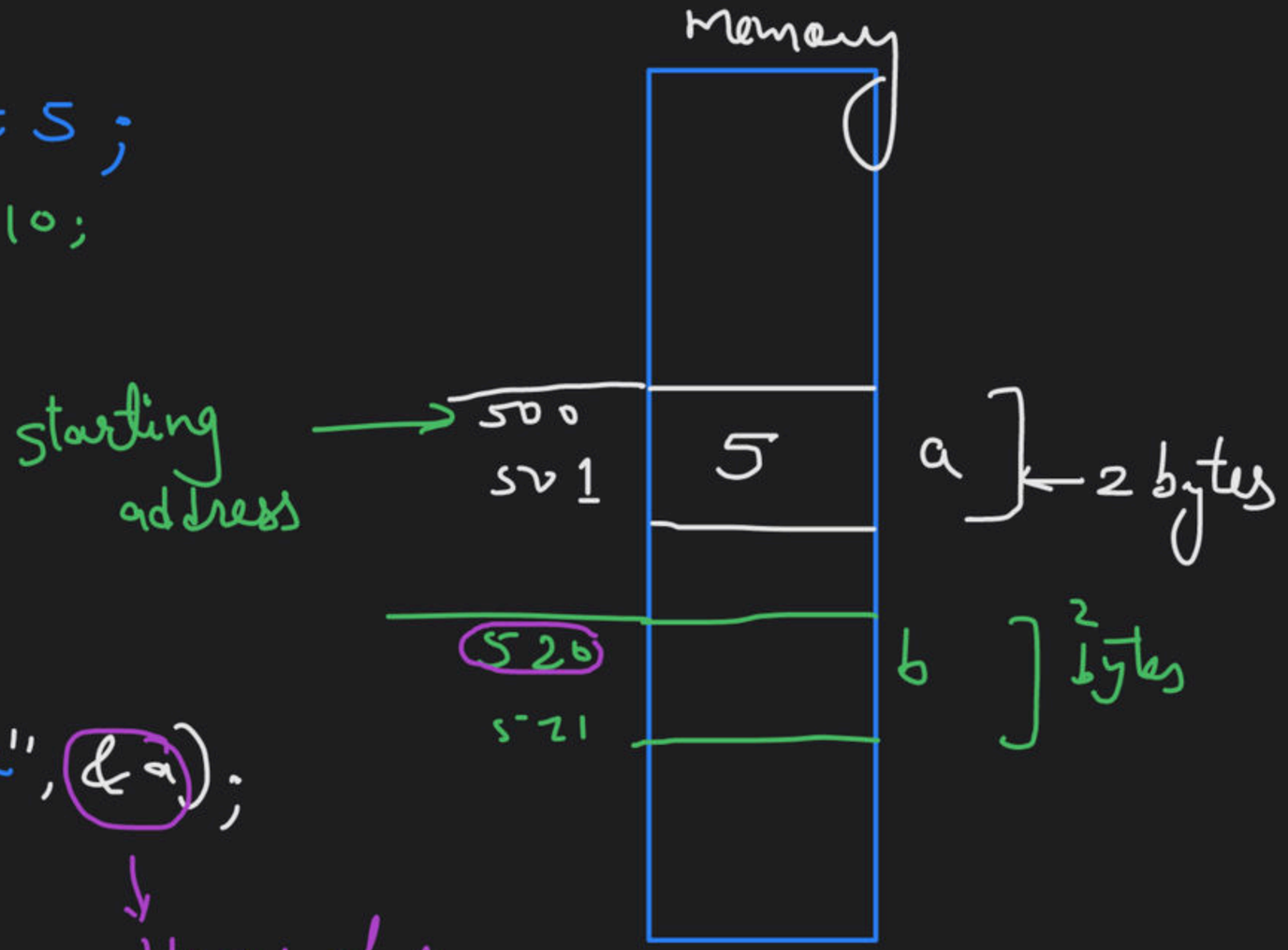
pow(x, n)

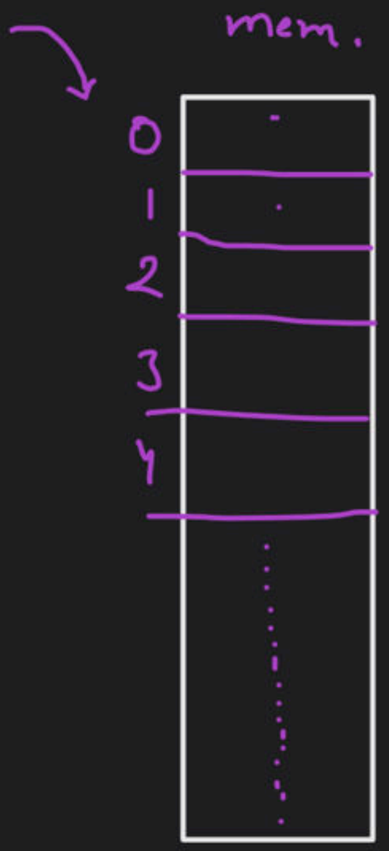
⇓

x^n

variable

```
int a = 5;
int b = 10;
```






```
printf("%x", &a);
```

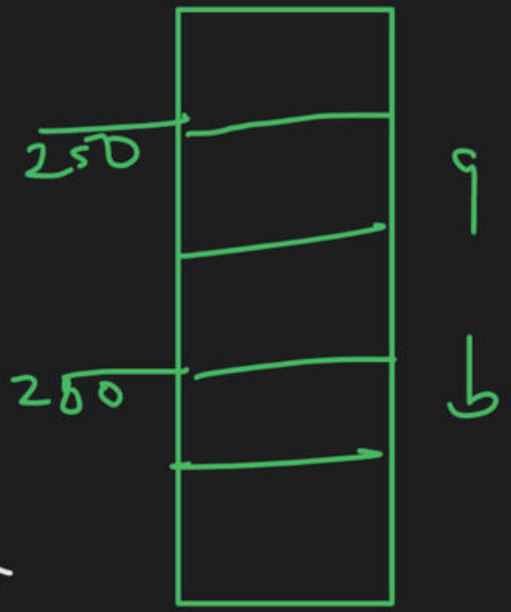
or

↓ address of a

%U for unsigned

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

error



b is an 'int variable

and can not be used

to store addresses.

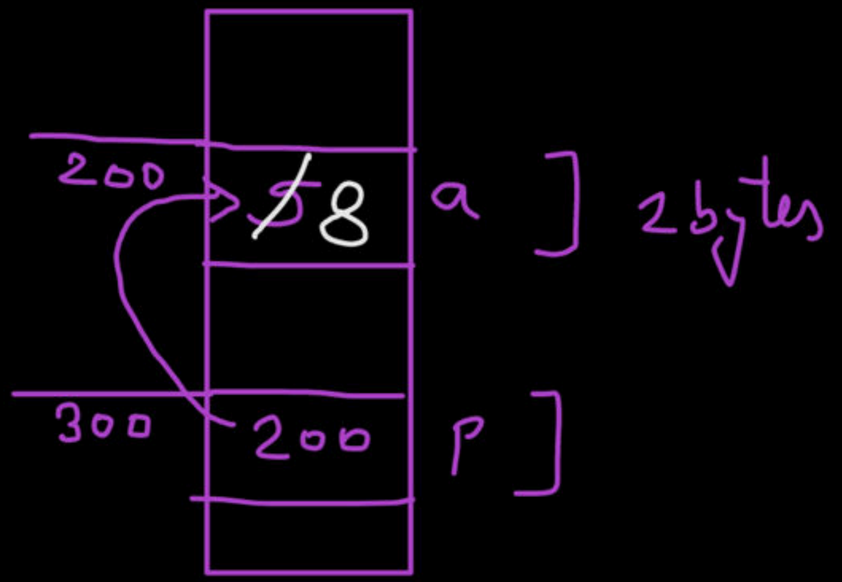
Pointer

pointer is a variable which is used to store address.

pointer declaration:-

int *p; => pointer p which can store add. of int

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



```
printf("%d", a);      5

printf("%d", &a);     200

printf("%d", p);      200

printf("%d", *p);     *(200) => 5

printf("%d", (*p)*4);  20

*p = 8;

printf("%d", (*p)+2);  10
printf("%d", a);       8
```

* => value at address

```
char *cp;
float *fp;
int *ip;
```

cp, fp, ip

↑ store addresses

```
printf("%d", sizeof(cp));  ⇒ size of integer 2
printf("%d", sizeof(fp));  ⇒ 2
printf("%d", sizeof(ip));  ⇒ 2
```

pointer stores add. ⇒ which is int ⇒ pointer = int
value size size

pointer arithmetic

increment or decrement in pointer is done by size of data item it points.

```
int *ip;
char *cp;
float *fp;
```

ip = ~~200~~ 202

cp = 300

fp = 400

} assuming

```
ip++;
ip = ip+3;
fp = fp+5;
```

ip = 202 + (3 * 2) = 208

fp = 400 + (5 * 4) = 420

allowed operations :-

1. Increment ++
2. Decrement --
3. Addition of any integer constant p + 3 , q + 8
4. subtraction −1, p − 4, q − 3
5. subtraction of 2 pointers p1 − p2
6. Comparison of pointers

```
int a = 5, b = 2;
int *p = &a;
int *q = &b;
printf("%d", ++(*p) * (*q));
printf("%d", (*p) + 4);
```

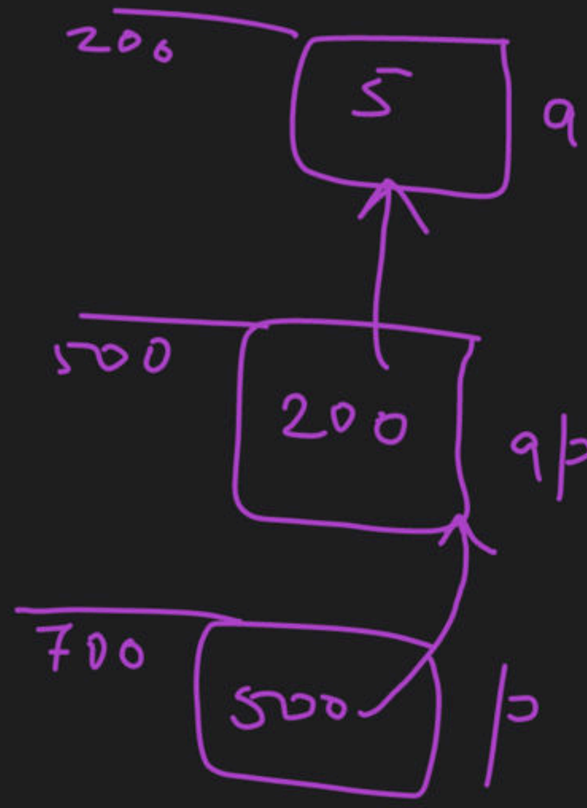
12

10

1210

```
int **p;

int a = 5;

int *ap;

ap = &a;

p = &ap;

printf("%d", p);   500
printf("%d", *p);  200
printf("%d", **p); 5
```

**p ⇒ p is a pointer which holds add. of another pointer to int.



Array

It is a collectn of similar datatype elements.

datatype name[size];

int A[5];

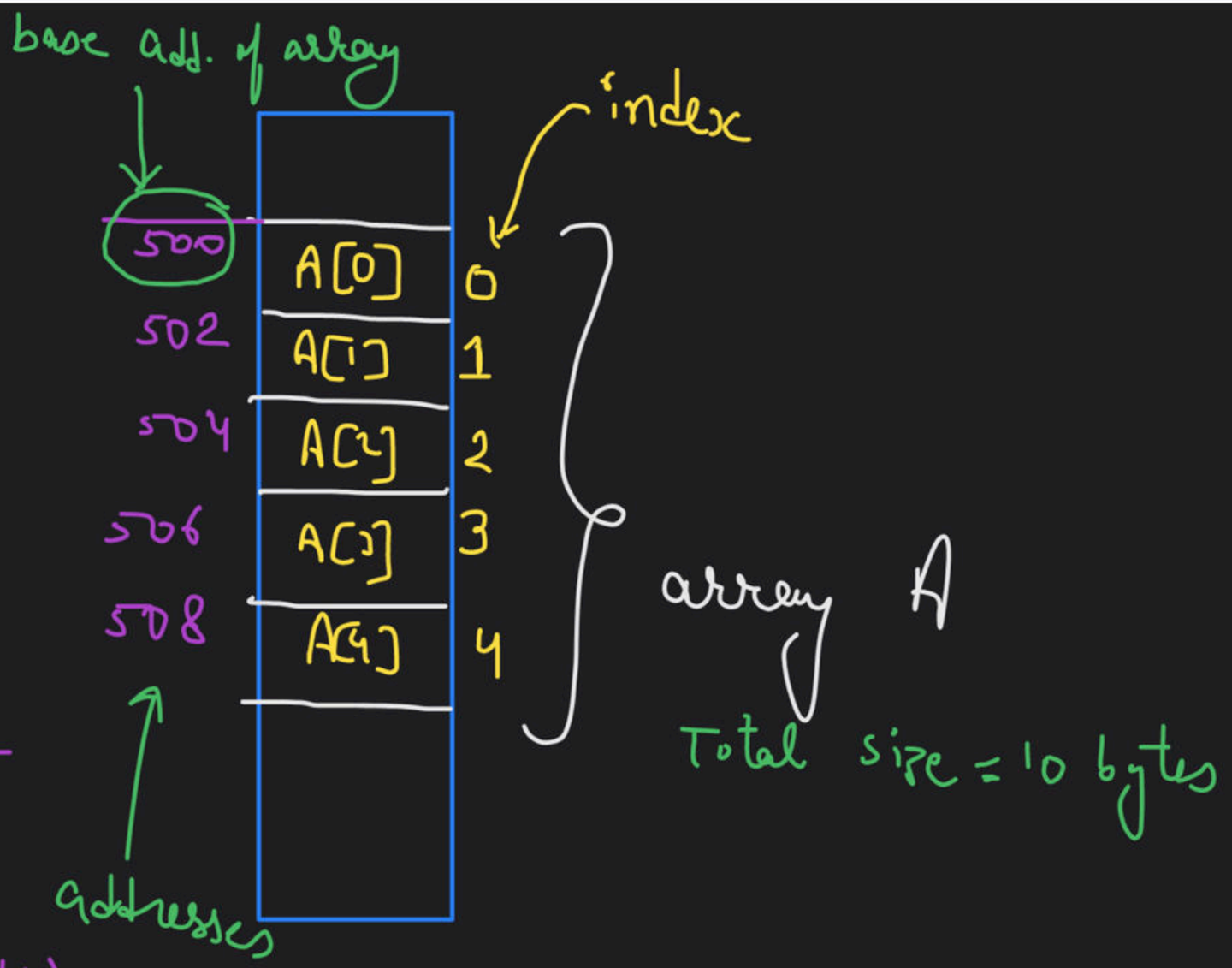
char B[500];

Characteristics :-

1. All elements of array are stored on consecutive memory locations.
2. All elements of array can be accessed using a set of indexes.

indexes starting from zero.

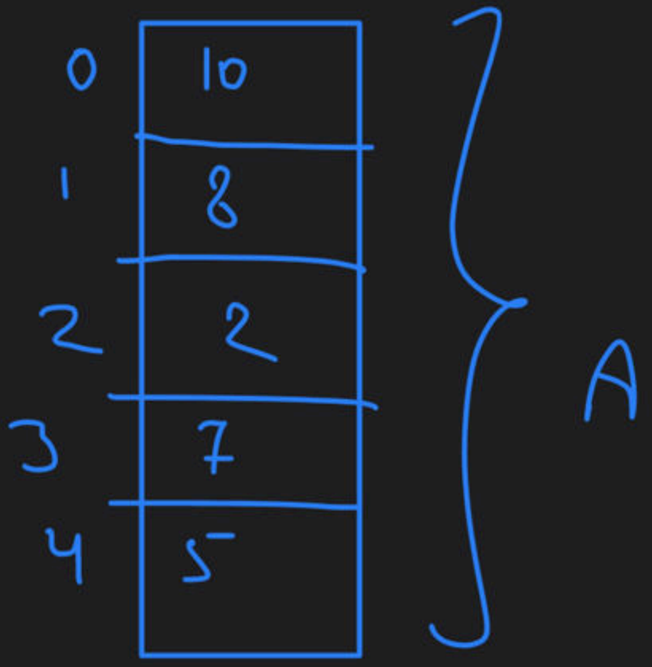
```
int A[5];

scanf("%d", &A[0]);
scanf("%d", &A[1]);
        :
scanf("%d", &A[4]);
```



```
int i;
for(i=0; i<=4; i++)
{
 scanf("%d", &A[i]);
}
```

```
int A[5] = {10, 8, 2, 7, 5};
printf("%d", A[3]);
```

7

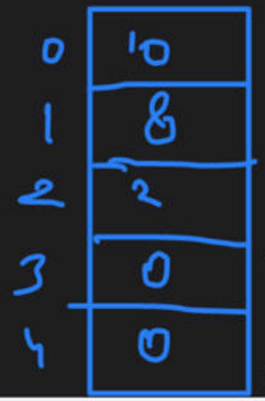



```
int A[5];
printf("%d", A[2]);
```

⇓

unpredicted

```
int A[5] = {10, 8, 2};
```



initialized to 0.

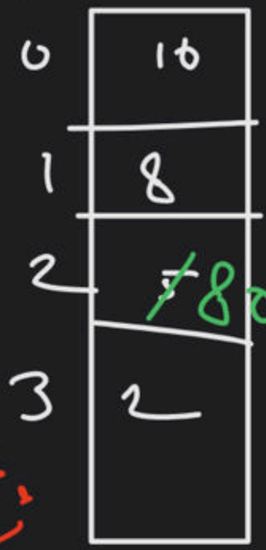
all value zero

```
int A[100] = {0};
```

```
int A[] = {10, 8, 5, 2};
```

A is of size 4.

```
A[2] = 80;
```




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

```
int A[] = {'m', 'y', 'c', 'o', 'u', 'r', 's', 'e'};
printf("%d", A[4]);
```

⇒ 117 ascii value of 'u'

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

⇒ base address of array

---

array name is a constant pointer which stores base address of array

```
int A[5];
```

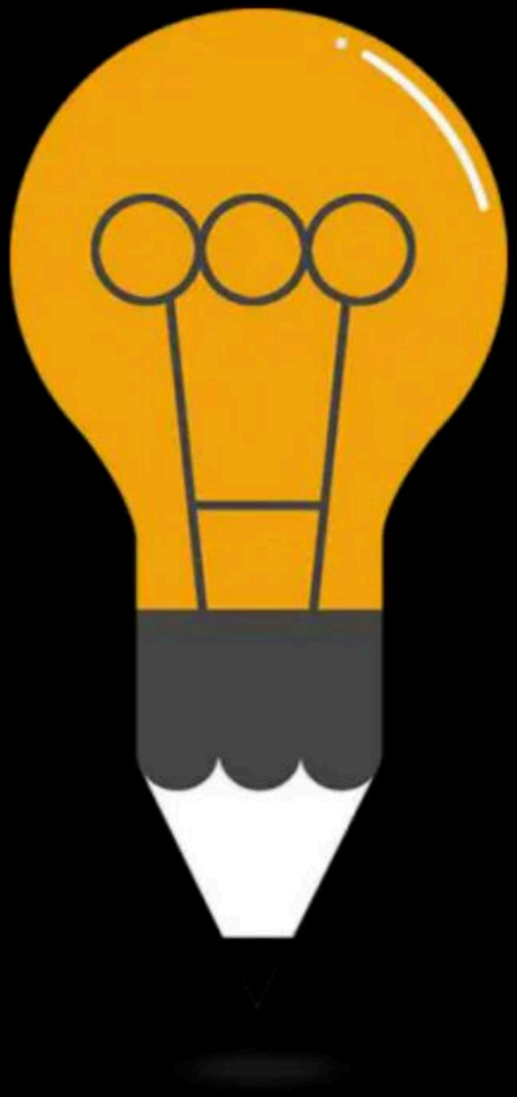

```
A++;
```

⇐ error

A is a constant pointer

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

200



DPP 3

By: Vishvadeep Gothi

Question

What is the output of the following programs-

```
void main(){
int m, count=0;
for(int i=1;i<=m; i*=2)
{
 count+=1;
}
}
```

(a) count=ceil(log m) - 1
(b) count=floor(log m) + 1
(c) count=ceil(log m)
(d) count=floor(log m) - 1
(e) None of the above

Question

What is the output of the following programs-

```
void main(){
int i,j,k,count=0,n;
for(i=0;i<=n;i++){
 for(j=0;j<=n;j++){
  for(k=n/3;k<=n;k+=n/3)
   count++;
}
}
printf("%d,%d,%d,%d",i, j, k, c);
}
```

Question

What is the output of the following programs-

```
for(int i=k, j=m; k<=n && j>=t; k++, j--)
{
}
```

Assume initially k<n and m>t.
When will the loop terminate?
(a) k<=n || j>=t
(b) k<=n && j>=t
(c) k>n || j<t
(d) k>n && j<t

Question

What is the output of the following programs-

```
void main(){
int i,j=1,count=0,n;
for(i=n;i>0;i/=2)
count=count+1;
while(j<n)
{
count–;
j*=2;
}
printf(“%d”,count);
}
```

Question

What is the output of the following programs-

```
void main(){
int i, j, count=0, n;
for(int i=1; i<n; i*=2){
 for(j=1; j<n; j*=2){
count++;
Break;
}
}
do
{
Count--;
} while(0);
printf("%d,%d,%d",i, j, count);
}
```

Happy Learning.!