

CS & IT ENGINEERING

Programming in C

Arrays and Pointer – 3


DPP 03 Discussion Notes



By- Pankaj Sharma sir



TOPICS TO BE COVERED



01 Question

02 Discussion

Q.1

Consider the following program:

[NAT]

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

```
int main()
```

```
{
```

```
    int p=10, *q;
```

```
    q=&p;
```

```
    *q=p+++*q;
```

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

```
    return 0;
```

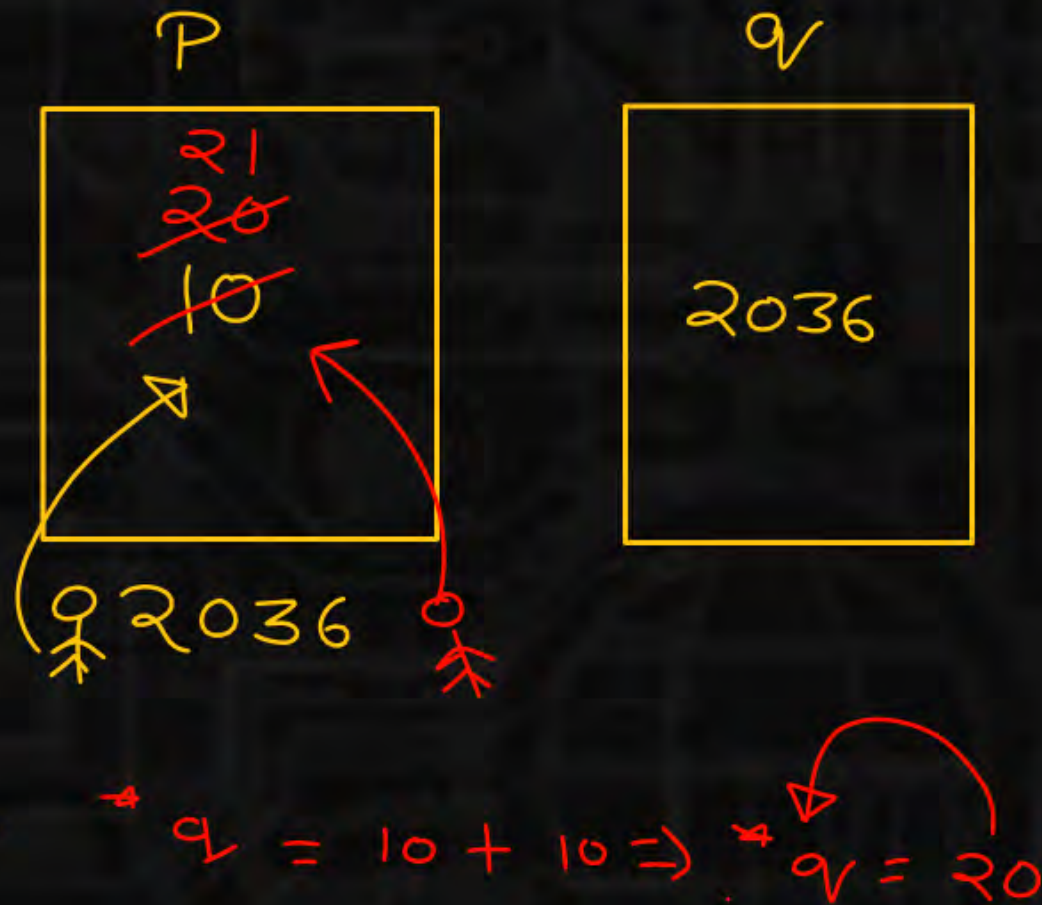
```
}
```

The output is 21.

$$*q = (p++) + (*q)$$

$$(i) *q = p + *q$$

$$(ii) p = p + 1$$



Q.2

Consider the following program:

[MCQ]

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

```
int * f(){
```

```
static int a[4]={1, 2, 3, 4};
```

```
return a;
```

```
}
```

```
int main(){
```

```
int *p, i;
```

```
p=f();
```

```
for(i=0;i<3;i++){
```

```
printf("%d\t", p[i]+p[i+1]);
```

```
}
```

```
return 0;
```

```
}
```

A.

Compilation Error

B.

Runtime Error

C.

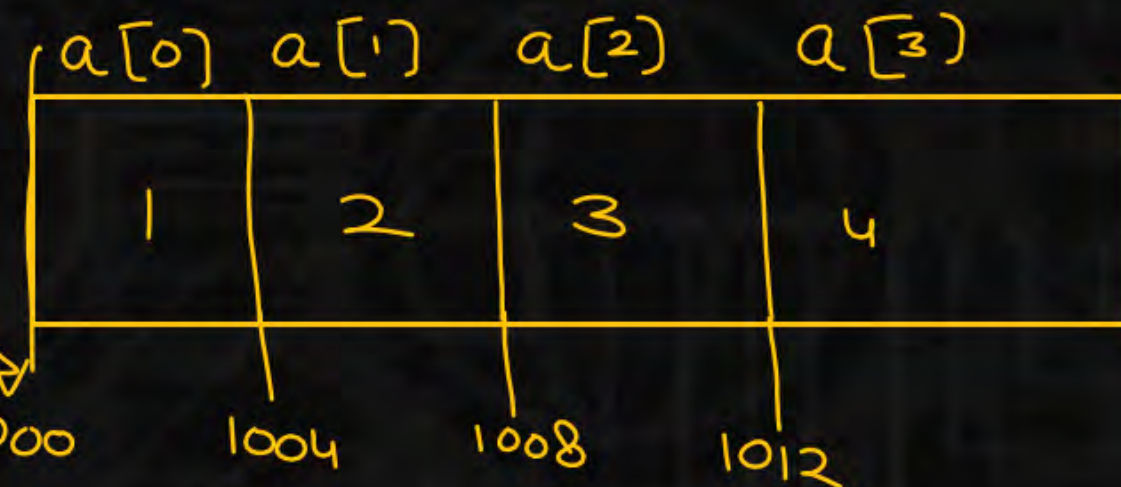
3 5 7

D.

None

Handwritten note: $\text{return a;} \rightarrow \&a[0]$

Handwritten note: K



The output is-

Q.2**[MCQ]**

Consider the following program:

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

```
int * f(){
```

```
static int a[4]={1, 2, 3, 4};
```

```
return a;
```

```
}
```

```
int main(){
```

```
int *p, i;
```

```
p=f();
```

```
for(i=0;i<3;i++){
```

```
printf("%d\t", p[i]+p[i+1]);
```

```
}
```

```
return 0;
```

```
}
```

The output is-

A.

Compilation Error

B.

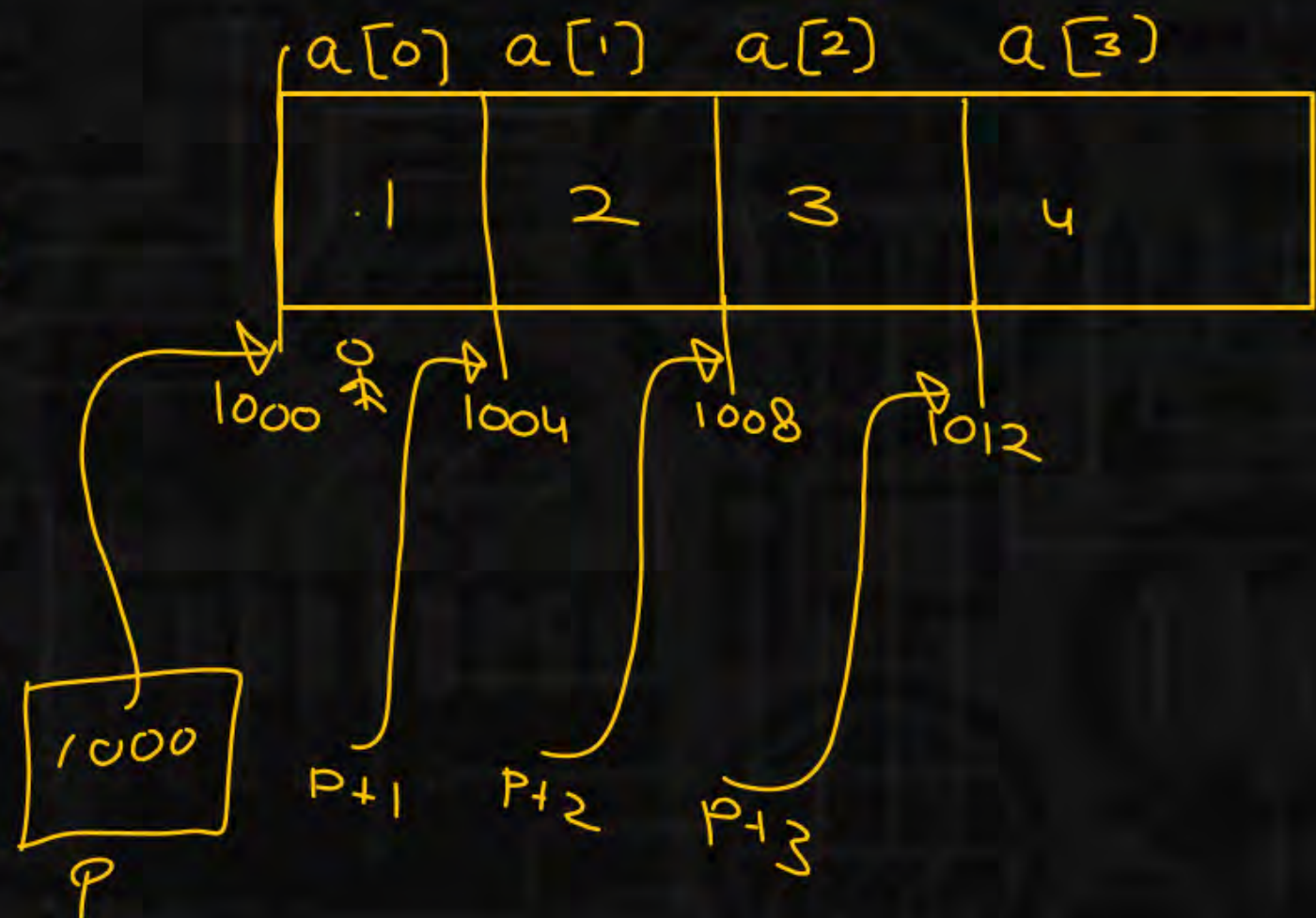
Runtime Error

C.

3 5 7

D.

None

(i) $i=0$ $P[0] + P[1]$ $*(P+0) + *(P+1)$ $1 + 2 \Rightarrow 3$ (ii) $i=1$ $P[1] + P[2]$
 $*(P+1) + *(P+2)$ $2 + 3 \Rightarrow 5$ (iii) $i=2$ $P[2] + P[3]$
 $3 + *(P+3) = 7$ $3 + 4$ 

Q.3

Consider the following program:

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

```
int main()
```

```
{
```

```
int p=10, s=20, *q, **r;
```

```
q=&p;
```

```
*q=p+++*q;
```

```
q=&s;
```

```
r=&q;
```

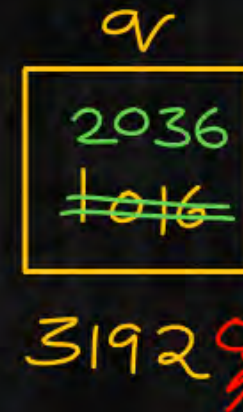
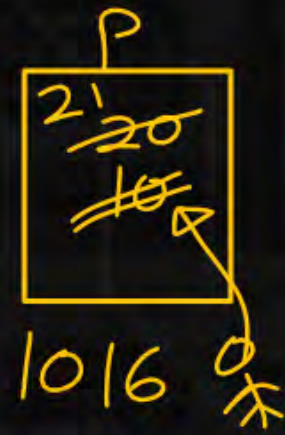
```
**r=--*q***r;
```

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

```
return 0;
```

```
}
```

The output is 382.



[NAT]



$$*q = \{p++\} + *q$$

$$\begin{aligned} (i) \quad &*q = p + *q \\ (ii) \quad &p = p + 1 \end{aligned}$$

$$*q = 10 + 10 = 20$$

$$21 + 361 = 382$$

$$\begin{aligned} r &= 3192 \\ *r &= 2036 \\ **r &= \text{val.at}(m, 2036) \end{aligned}$$

$$**r = --(*q) * (**r)$$

$$\begin{aligned} (i) \quad &*q = *q - 1 \Rightarrow *q = 20 - 1 = 19 \\ (ii) \quad &**r = *q \times (**r) \end{aligned}$$

$$\begin{aligned} **r &= 19 \times 19 \\ **r &= 361 \end{aligned}$$

Q.4

Consider the following program:

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

```
int * f()
```

```
{
```

```
    int a[4]={1, 2, 3, 4};
```

```
    return a;
```

```
}
```

```
int main()
```

```
{
```

```
    int *p, i;
```

```
    p=f();
```

```
    for(i=0;i<3;i++){
```

```
        printf("%d\t", p[i]+p[i+1]);
```

```
    }
```

```
    return 0;
```

```
}
```

The output is-

A.

Compilation Error

B.

Runtime Error

C.

3 5 7

D.

None

[MCQ]



Q.5

Consider the following statements:

[MCQ]



Correct

P: $\text{int}^* \text{p}(\text{int}^*)$ - p is a function that takes an integer pointer as argument and returns an integer pointer.

Correct

Q: $\text{int} (*\text{p}(\text{int}^*))[]$ - p is a function that takes an integer pointer as argument and returns a pointer to an array of integers.

Which of the following is INCORRECT?

$\text{int}^* \text{p}(\text{int}^*)$

P is a function that takes a pointer to integer as argument and returns pointer to integer

A.

P only

B.

Q only

C.

Both P and Q

☒ D.

Neither P nor Q

~~$\text{int} (*\text{p}(\text{int}^*))[]$~~

P is a function that takes integer pointer as argument and returns pointer to array of integers

Q.6

Consider the following program:

```

#include<stdio.h>
void f(int (*q)[2]){
    printf("%d\t",(*q)[1]);
    q+=2;
    printf("%d",(*q)[1]);
}
int main()
{
    int a[][2]={2,4,6,8,10,12};
    int (*ptr)[2]=a;
    f(ptr);
    return 0;
}

```

ptr is a pointer
to an array of
2 integers.

A.

4 12

C.

2 10

B.

4 8

D.

2 6

[MCQ]

$$q = \&a[0]$$

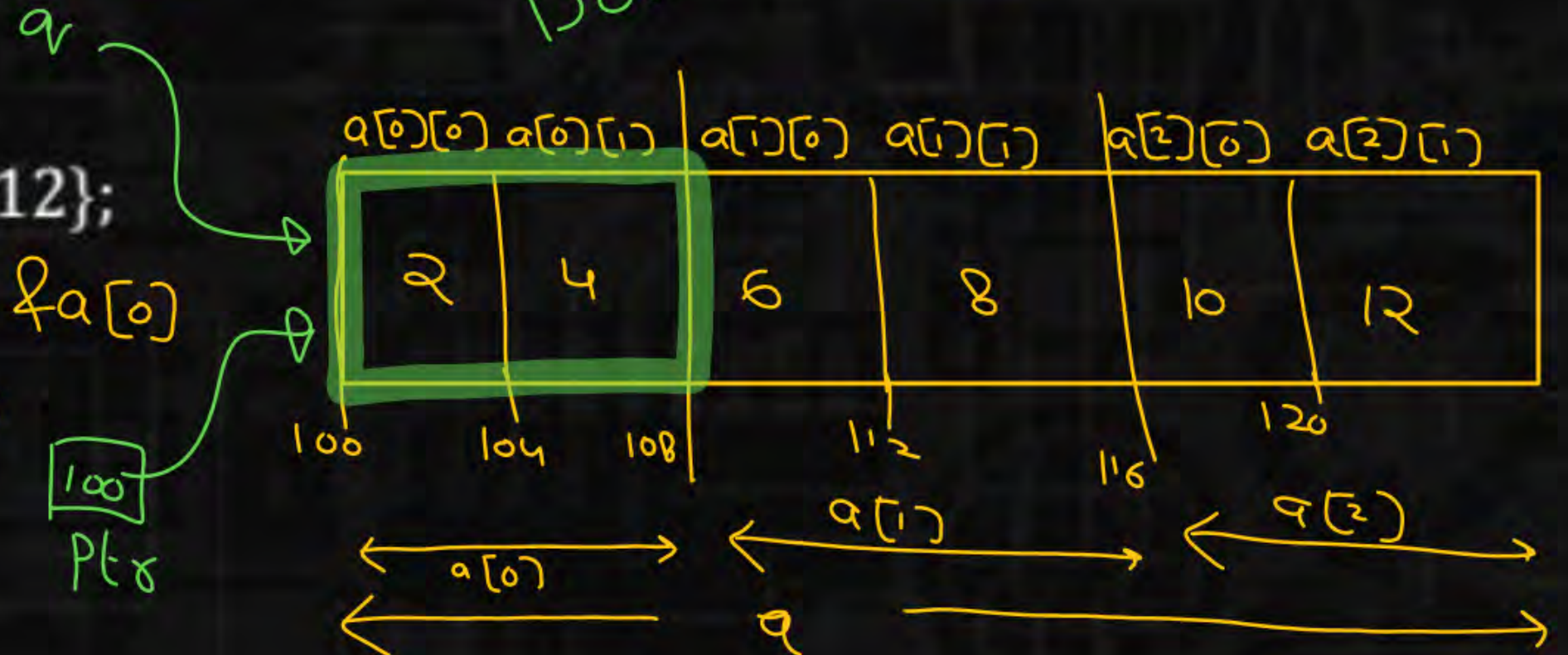
$$q+1 \Rightarrow \&a[0]+1 = \&a[1]$$

$$*q[1] \quad *(q+1) \Rightarrow *\&a[1] = a[1] = \&a[1][0]$$

$$q[1] = \&a[1][0]$$

$$*q[1] = *\&a[1][0] = a[1][0] = 6$$

Don't do this



The output is:

Q.6

Consider the following program:

```
#include<stdio.h>
void f(int (*q)[2]){
    printf("%d\t",(*q)[1]);
    q+=2;
    printf("%d",(*q)[1]);
}
int main()
{
    x * 2 = 6 ⇒ x = 3
    int a[][2]={2,4,6,8,10,12};
    int (*ptr)[2]=a;
    f(ptr);
    return 0;
}
```

4

A.

4 12

B.

4 8

C.

2 10

D.

2 6

[MCQ]



$$q = \&a[0]$$

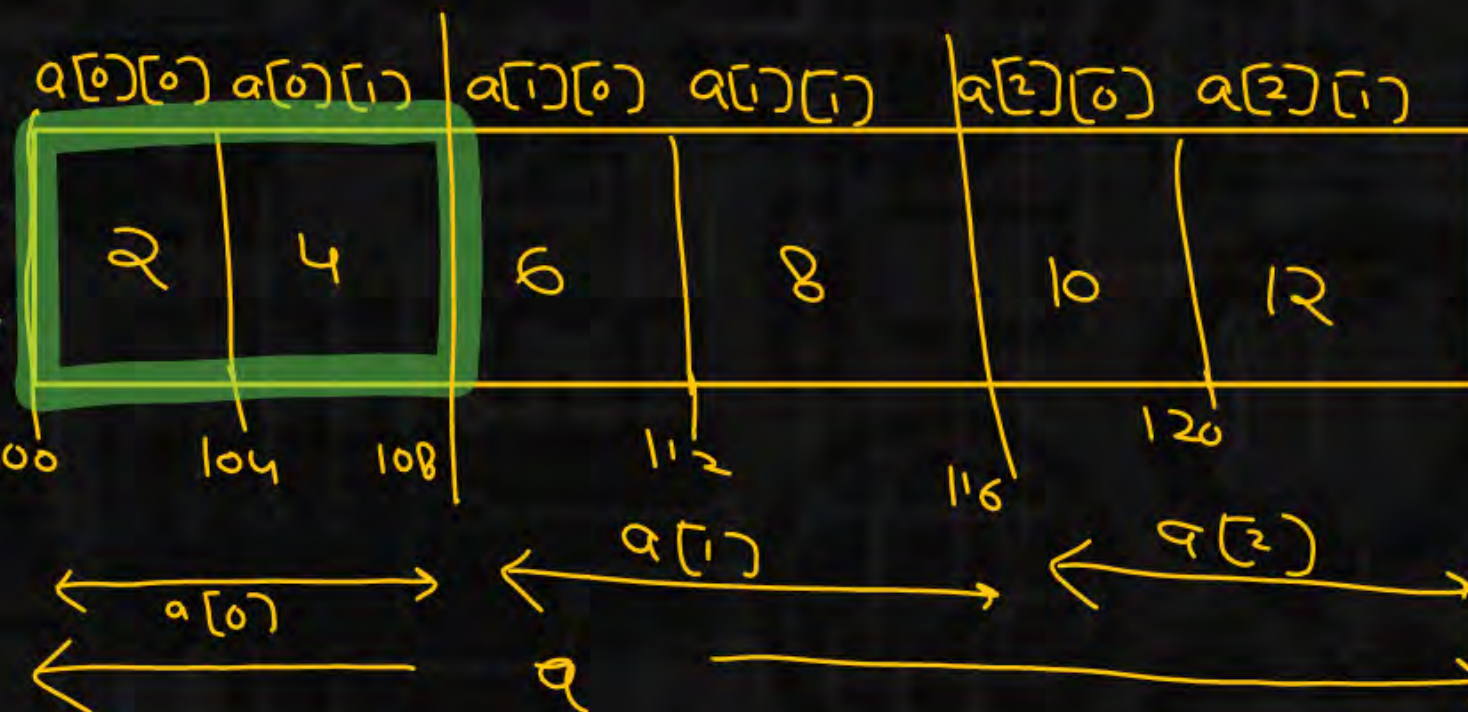
$$(*q) = *\&a[0] = a[0] = \&a[0][0]$$

$$(*q)[1] = *(&a[0][0] + 1) = *(&a[0][1]) = a[0][1]$$

q

&a[0]

100
ptr



The output is:

Q.6



[MCQ]

Consider the following program:

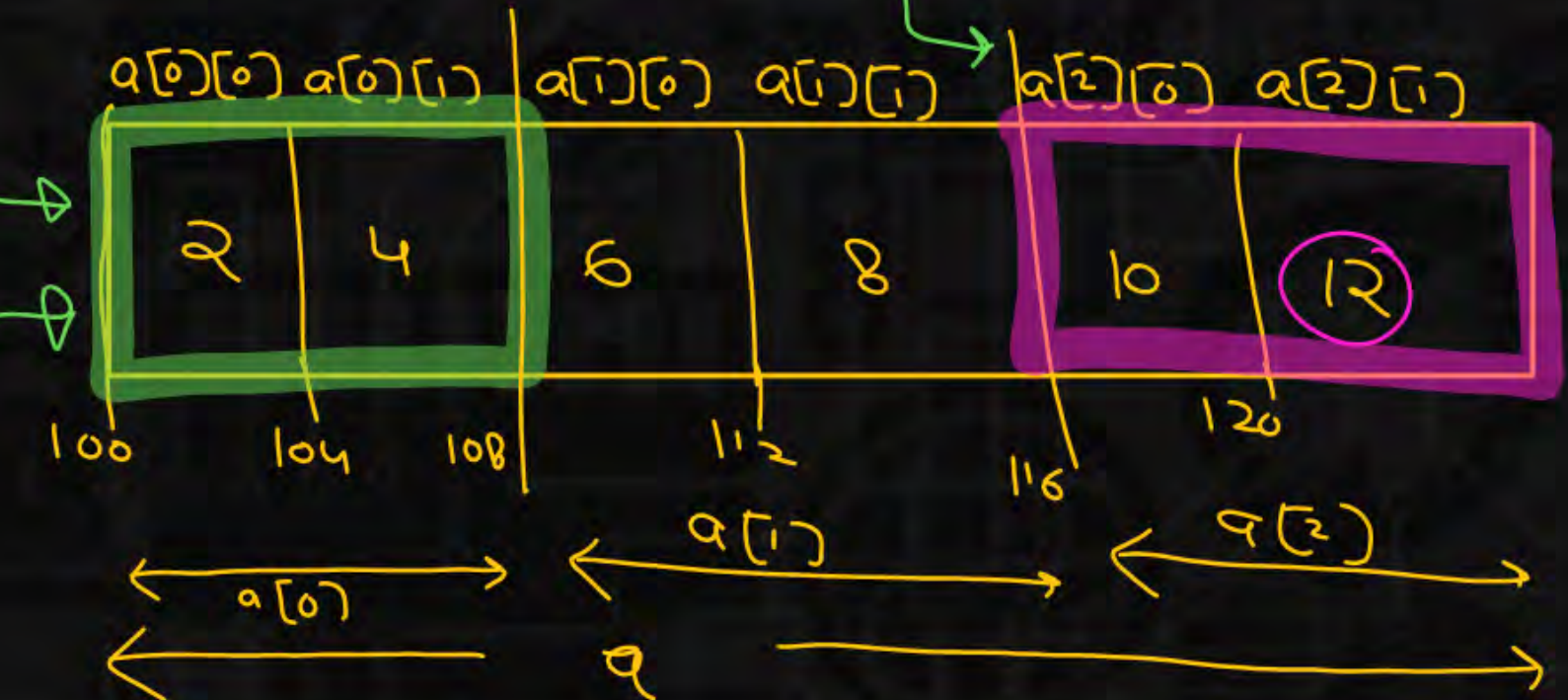
```
#include<stdio.h>
void f(int (*q)[2]){
printf("%d\t",(*q)[1]);
q+=2;
printf("%d",(*q)[1]);
}
int main()
{
int a[][2]={2,4,6,8,10,12};
int (*ptr)[2]=a;
f(ptr);
return 0;
}
```

$$x \times 2 = 6 \Rightarrow x = 3$$

ptr is a pointer
to an array of
2 integers.

 $\&a[0]$

100
ptr



$$q = \&a[0]$$

$$q = \&a[0] + 2 = \&a[2]$$

$$(*q) = *a[2] = a[2] = \&a[2][0]$$

$$(*q)[1] = (*q + 1) = *(\&a[2][0] + 1)$$

$$= *a[2][1]$$

A. 4 12

B. 4 8

C. 2 10

D. 2 6

The output is:

Q.7

Consider the following program:

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

```
int main()
```

```
{
```

```
int a[3]={0,1,2};
```

```
int *p=(int *)(&a+1);
```

```
printf("%d\t%d", *(a+1), *(p-1));
```

```
return 0;
```

```
}
```

The output is-

A.

Garbage value

B.

Segmentation fault

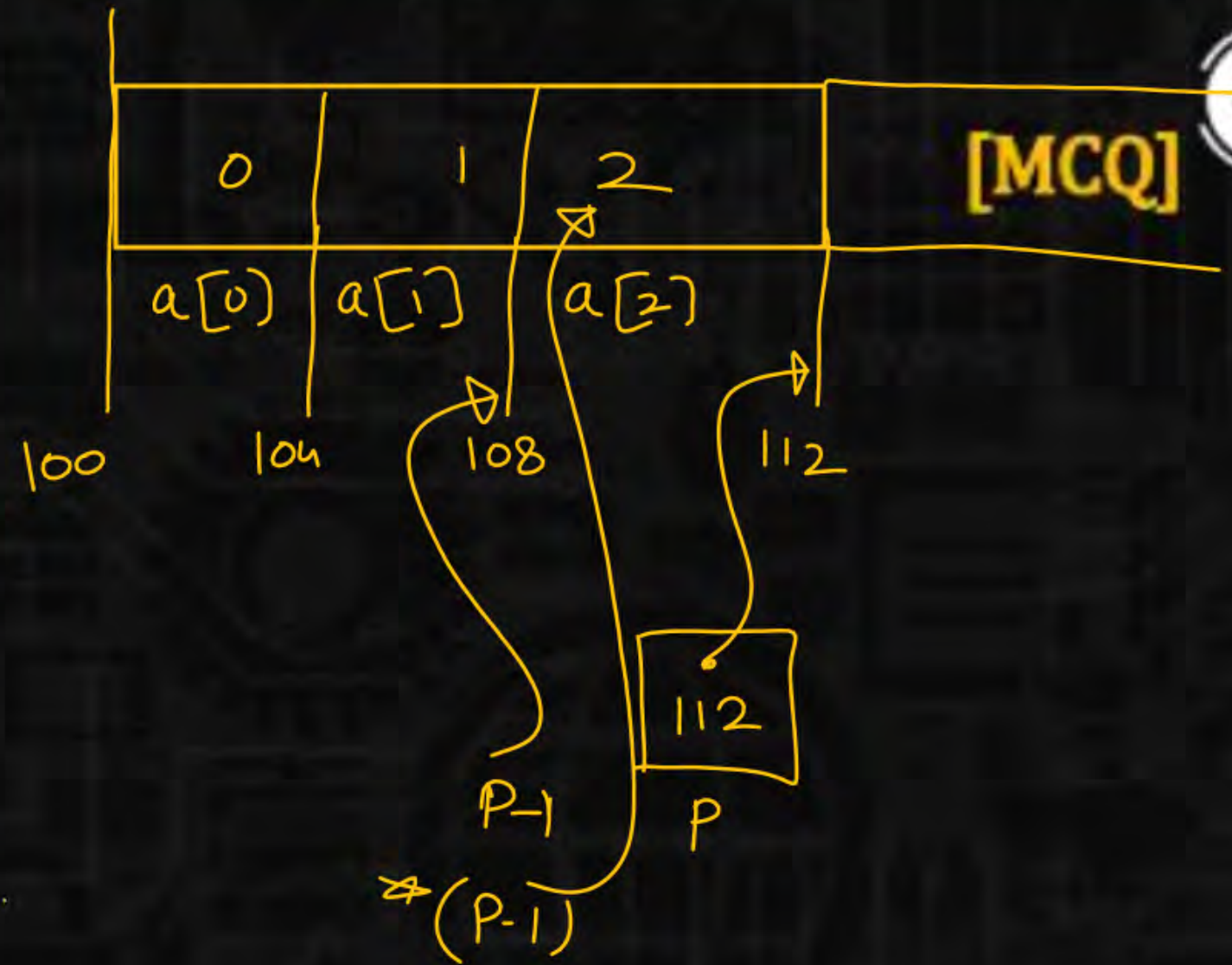
☒ C.

1 2

D.

Compilation Error

[MCQ]

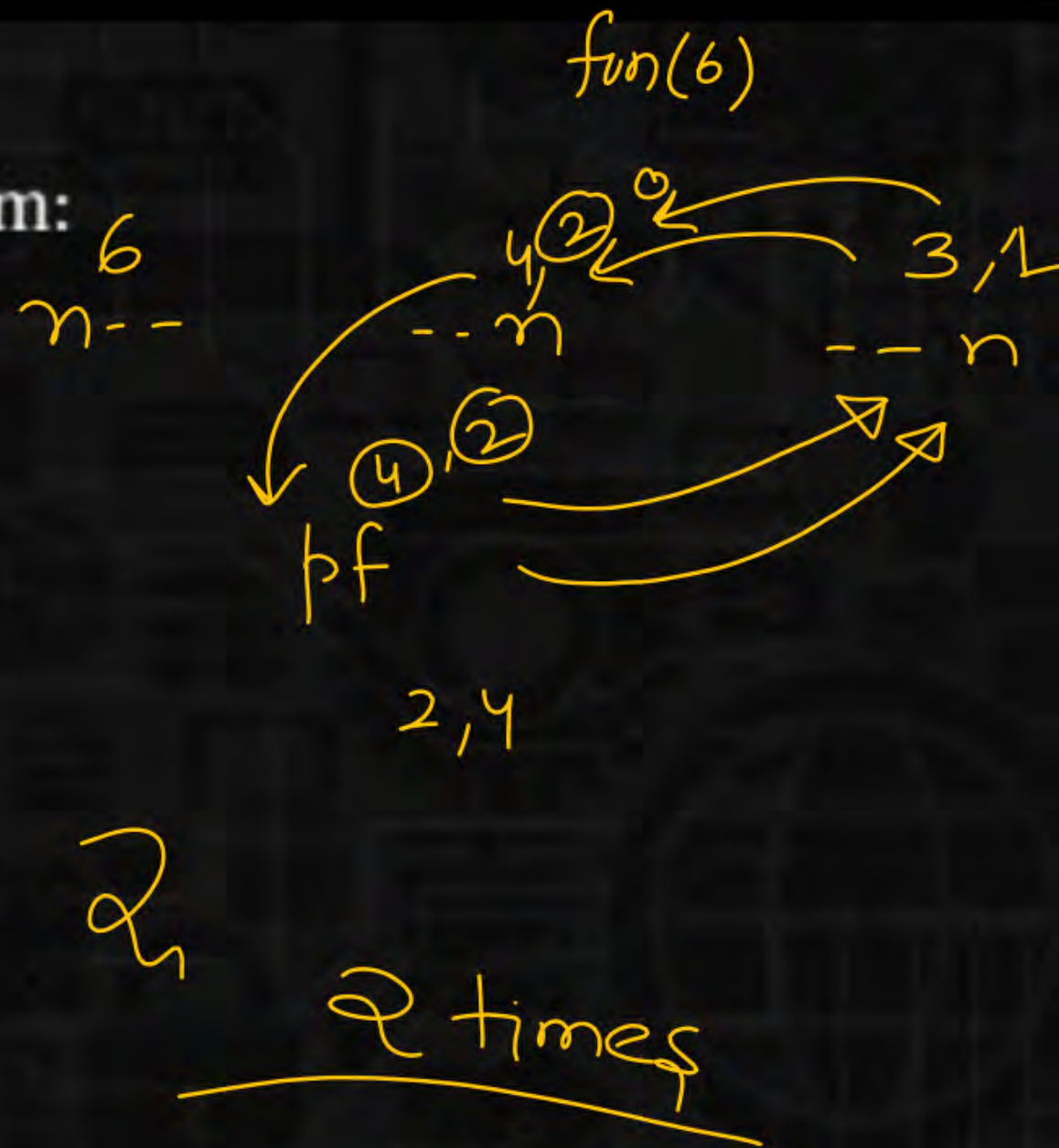


Q.8

Consider the following program:

```
#include<stdio.h>
void fun(int n){
    for(n--;--n;--n)
        printf("GATE WALLAH");
}
int main(){
    void (*p)(int)=fun;
    (*p)(6);
    return 0;
}
```

The output is-



[MCQ]



A.

Compilation Error

B.

Runtime Error

C.

Printf() is executed infinite number of times.

D.

Print() is executed two times

