unacademy

Arrays & Pointers Part - V

Comprehensive Course on C- Programming

Pankaj Sharma • Lesson 23 • May 30, 2024

unacademy

CS & IT Engineering

C Programming

Arrays & Pointers-V

Lecture Number- 23

By- Pankaj Sir

unacademy

Topics

to be covered

1. Arrays & Pointers-V

$int\ a[3][2][3] = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18\};$

st → 1000

4 bytes

(i) $a \rightarrow 1000\ (\&a[0])$

(ii) $\&a \rightarrow 1000$

(iii) $a[0] \rightarrow \&a[0][0]\ (1000)$

(1000)

iv) $a[0][0] \rightarrow \&a[0][0][0]$

v) $a[0][0][0] \rightarrow$ add x, → ve

1

vi) $a+1$

vii) $\&a+1$

viii) $a[0]+1$

ix) $a[0][0]+1$

x) $a[0][0][0]+1$

xi) $*a$

xii) $**a$

xiii) $***a$

xiv) $*a+1$

xv) $**a+1$

int a[3][2][3] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18}; 4 byte

vi) a+1 → &a[0] + 1

&a[0]+1×24

→ 1000+24

= 1024

vii) &a+1

→ whole array add (72 bytes)

&a+1×72 = 1000+72

= 1072

viii) a[0]+1

→ &a[0][0] + 1×12

→ 1012

ix) a[0][0]+1

x) a[0][0][0]+1

xi) *a

xii) **a

xiii) ***a

xiv) *a+1

xv) **a+1

int a[3][2][3] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18}; 4 byte

ix) a[0][0]+1 &a[0][0][0] + 1 x 4 ⇒ 1000+4 = 1004

x) a[0][0][0]+1 1+1 = 2

xi) *a ~~*~~&a[0] ⇒ a[0] ⇒ &a[0][0] 1000

xii) **a ~~*~~(&a[0][0]) = a[0][0] ⇒ &a[0][0][0] 1000

xiii) ***a ~~*~~(~~&~~a[0][0][0]) = 1

xiv) *a+1 → &a[0][0]+1 ⇒ &a[0][0]+1 x 12 = 1000+12 = 1012

xv) **a+1 → &a[0][0][0] + 1 x 4 ⇒ 1000+4 = 1004

Declaration & Initialization

1.) int a[ ]; Invalid

2) int a[ ] = {10, 20, 30}; ✓ valid

(i) If only declaration is there without initialization

⇒ It is mandatory (compulsory) to provide the size of each dimension.

```
int a[ ][ ] ; ✗
int a[ ][4] ; ✗
int a[2][4] ; ✓
```

Invalid

(ii) In case, we are initializing an array, there is flexibility that u can omit the size of 1st dimension.

No other dimension is having such flexibility.

(i) int a[ ]; Invalid

(ii) int a[2]; ✓

(iii) int a[ ] = {10,20}; ✓

(iv) int a[2] = {10,20}; ✓

v) int a[2] = {1}; ✓

1) int a[][]; ✗

2) int a[2][]; ✗

3) int a[][3]; ✗

4) int a[2][3]; ✓

declaration without initialization

5) int a[][] = {1,2,3,4,5,6}; invalid

6) int a[][3] = {1,2,3,4,5,6}; ✓

7) int a[2][] = {1,2,3,4,5,6}; ✗

8) int a[2][3] = {1,2,3,4,5,6}; ✓

Initialization

unacademy

int a[ ][3] = {1, 2, 3, 4, 5, 6}

printf("%u", a+1) ⇒

unacademy

int a[x][3] = {1,2,3,4,5,6};

x*3 = 6

x = 2

int a[2][3] = {1,2,3,4,5,6}

unacademy

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

$x \times 3 = 4$

$x = \lceil 1.33 \rceil \Rightarrow 2$

```
printf("%d", sizeof(a)) => 24
```

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

unacademy

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

$x \times 3 = 2$

$x = \frac{2}{3} \Rightarrow \lceil .666 \rceil$

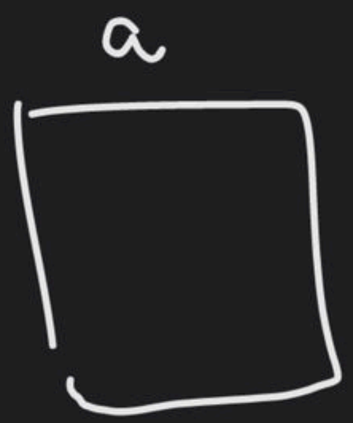
$= 1$

int a[1][3] = {1,2}

Pointers

Special variable that are used to store address of other variables.

int a;



int *p;

*p ⇒ Integer

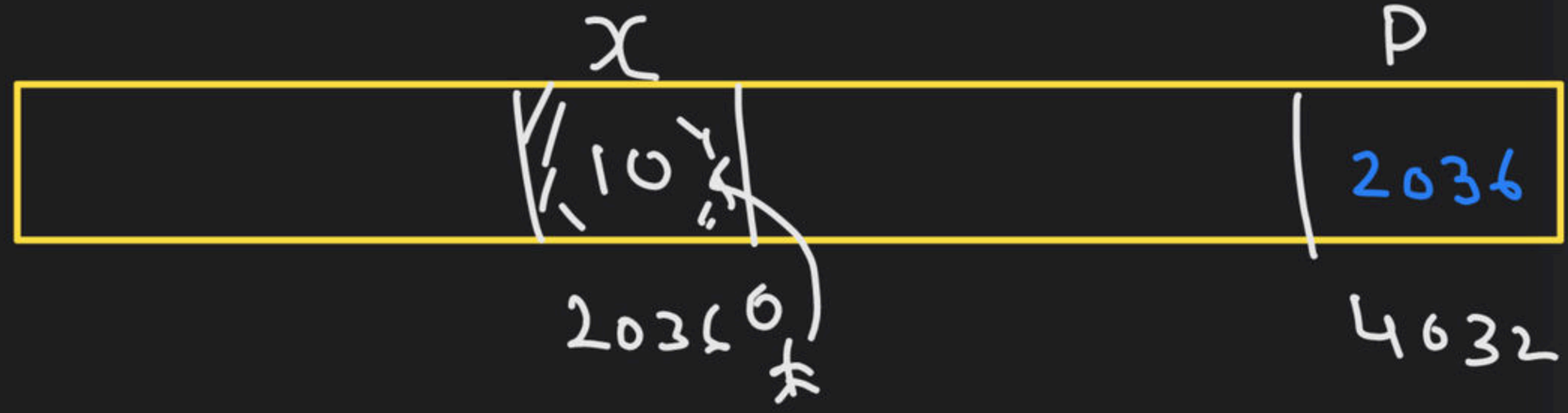
int *p;

p is a pointer to integer

p can store add. of some int. var.

unacademy

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



$p = \&x;$

2036

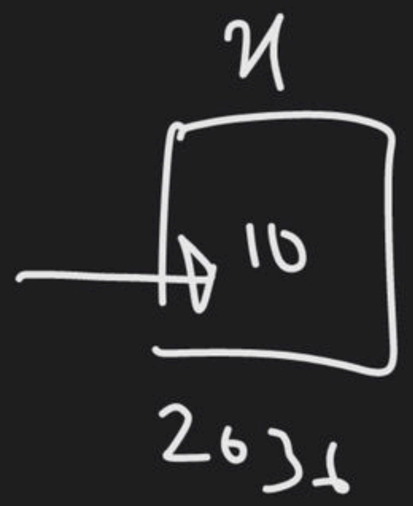
$p = 2036$ (Memory loc. 2036)

$*p$ = value at (Memory loc. 2036) = 10

```
pf("%u", x);
pf("%u", p);
```

```
pf("%u", *p)
```

$x \Rightarrow 10$



$P \Rightarrow 2031$

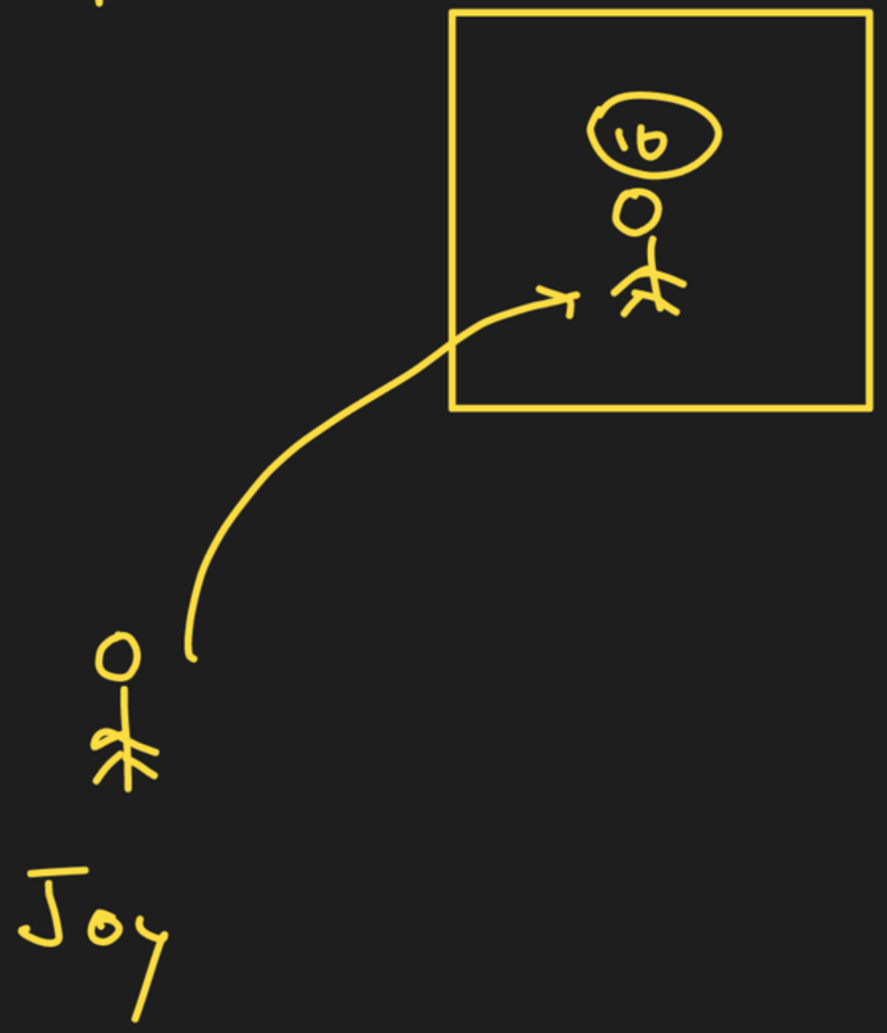
```
int x = 10;
int *p;
int **q;
```

p → address of integer var.

p is a pointer to integer.

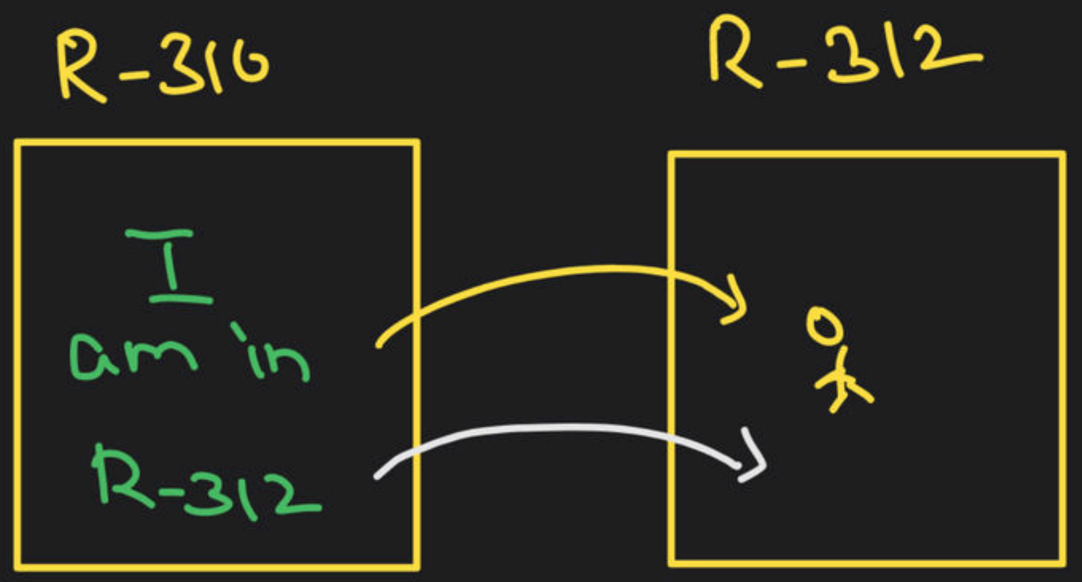
q is a pointer to pointer to integer

int x = 10

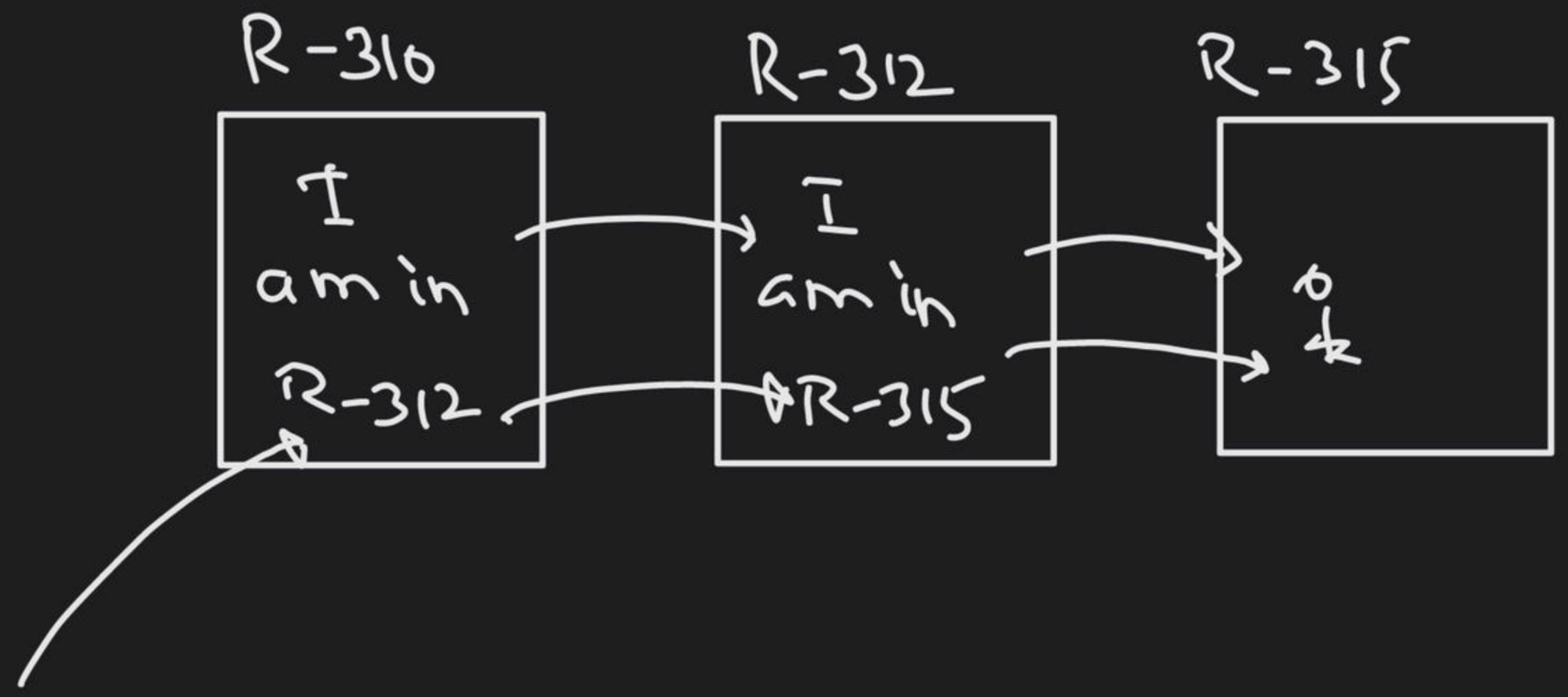


unacademy

int *p;



int **q;



int **pq;

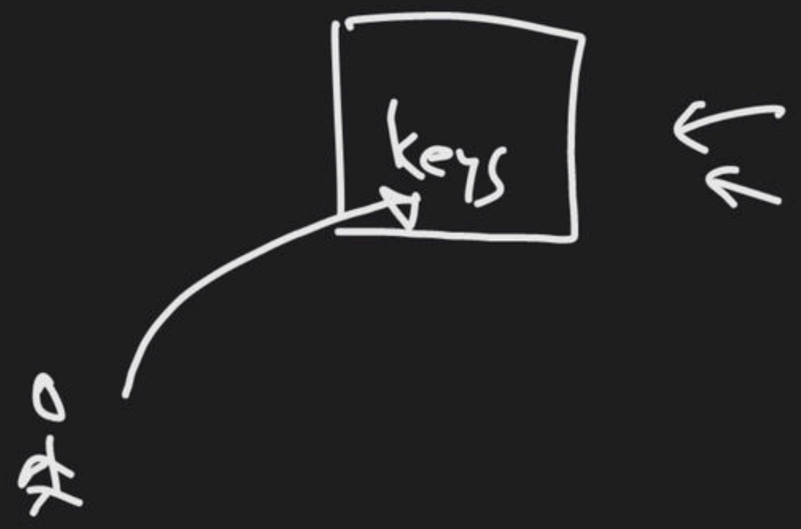
pq →

Pointer to pointer

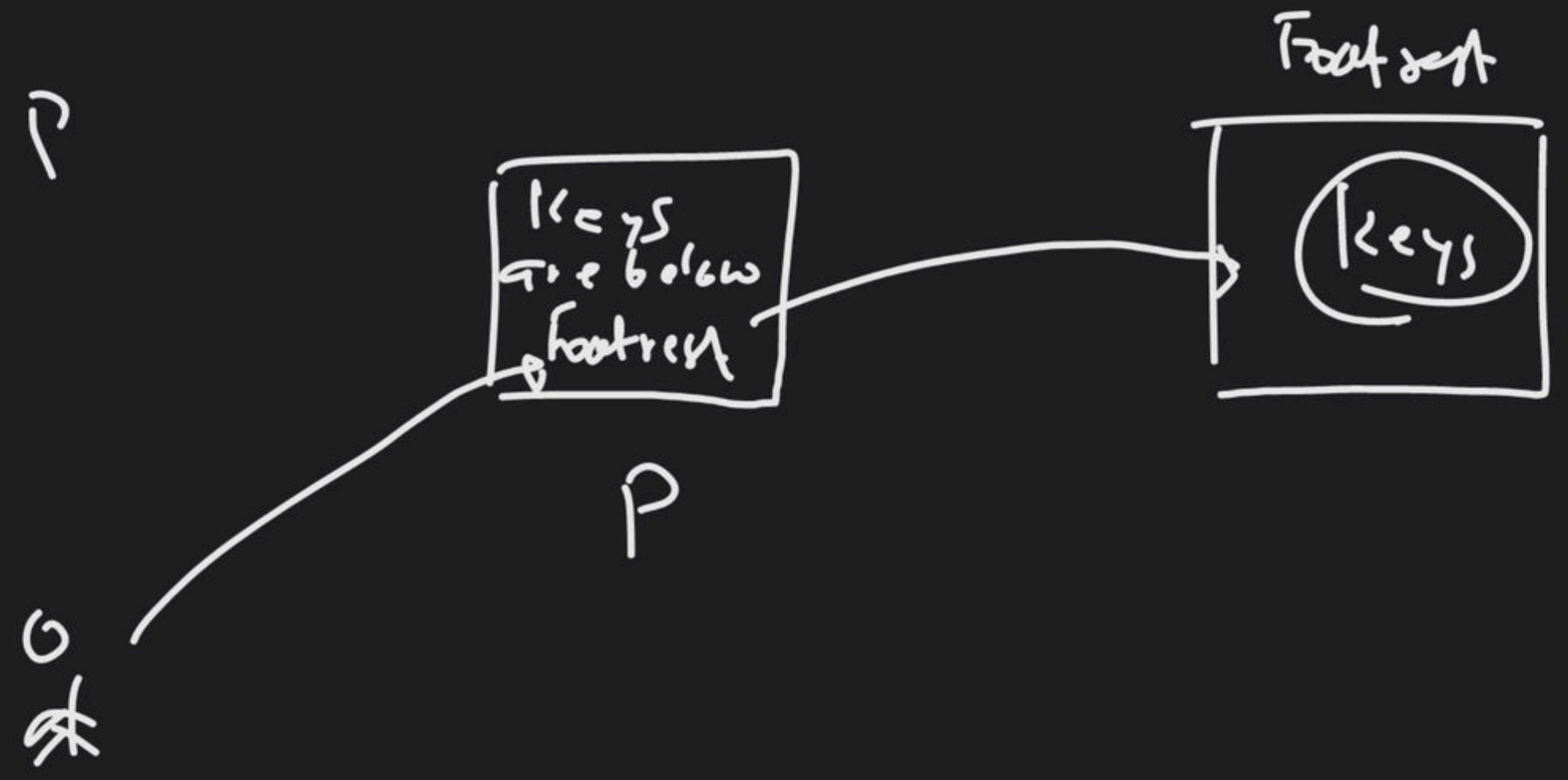
```
int x = 10;
int *p;
int **q;
```

Kevin

int x = 10;

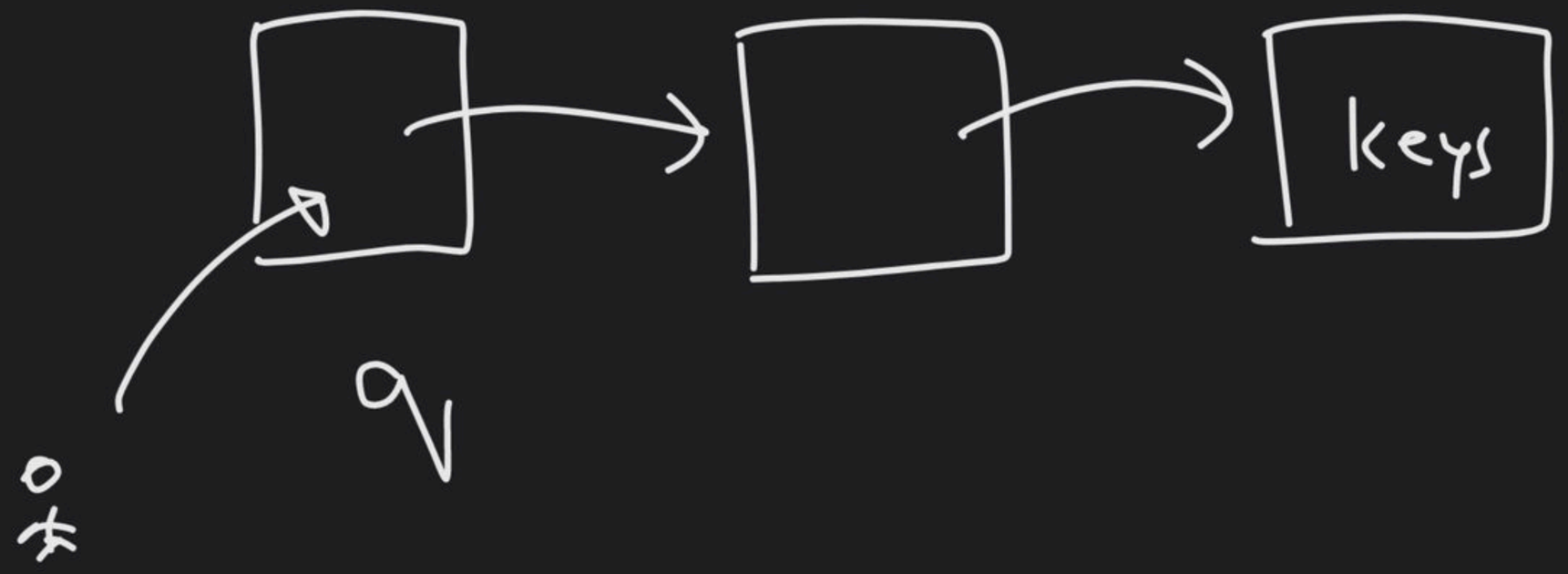


int *p



unacademy

int **q;



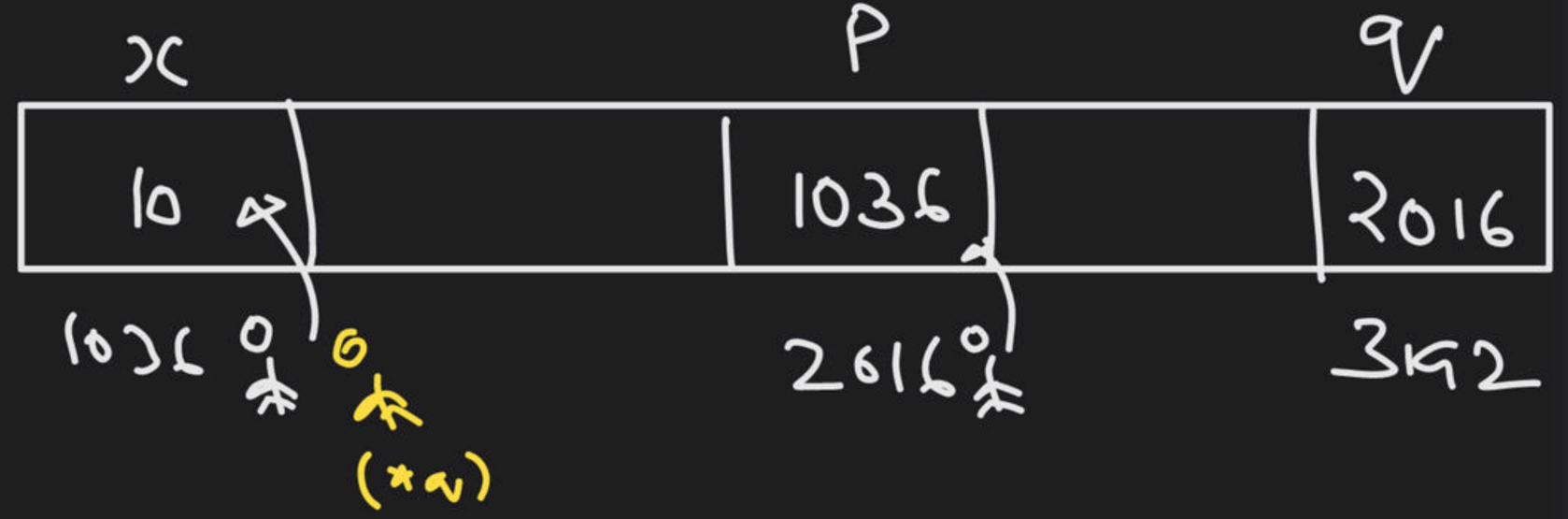
int *q ⇒ q is a pointer to pointer to integer

int *q ⇒ q = address of pointer to integer.

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

$$\boxed{\begin{aligned} p &= \&x; \\ q &= \&p; \end{aligned}}$$

```
pf("%u", p);   1036
pf("%u", *p);  10
pf("%u", q);   2016
pf("%u", *q);  1036
pf("%u", **q);  10
```



p = (Memory location 1036)

*p ⇒ value at (Mem. loc. 1036) = 10

q ⇒ Memory location 2016

*q ⇒ value at (Mem. loc 2016)

*q ⇒ memory location 1036

**q ⇒ value at (Mem. loc. 1036) = 10

unacademy

$$P = \ell x$$

$$\cancel{\ell} P = \cancel{\ell} \ell x$$

$$\boxed{\cancel{\ell} P = x}$$

10

$$q = \ell p$$

$$\cancel{\ell} q = \cancel{\ell} \ell p$$

$$\cancel{\ell} q = p$$

$$\cancel{\ell} q = \ell x$$

$$\cancel{\ell}\cancel{\ell} q = \cancel{\ell} \ell x$$

$$\boxed{\cancel{\ell}\cancel{\ell} q = x}$$

10

unacademy

Shivam $\xrightarrow{\text{After C}}$ C++

unacademy

THANK YOU!

Here's to a cracking journey ahead!