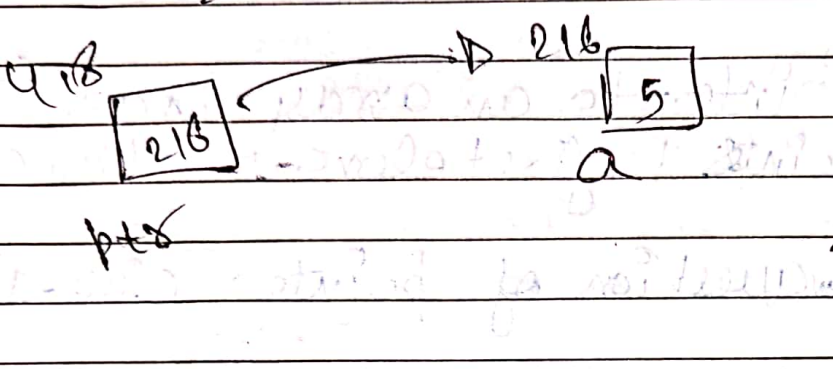
8/10/23 Class-8

Date 43

Revision:

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



Q) -> int arr[5] = {1, 2, 3, 4, 5};

int * ptr = &arr -> X

int *aptr = arr; -> ✓

&arr -> carries address of full arr

pointer to an integer No to an Array

Pointer to an Array =>

```
int arr[5] = {1, 2, 3, 4, 5};
int (*ptr)[5] = &arr
```

Spiral

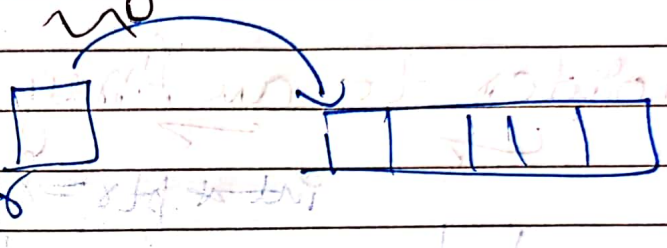
8/10/23 Date 19

Note:- → data base addresses but edge case

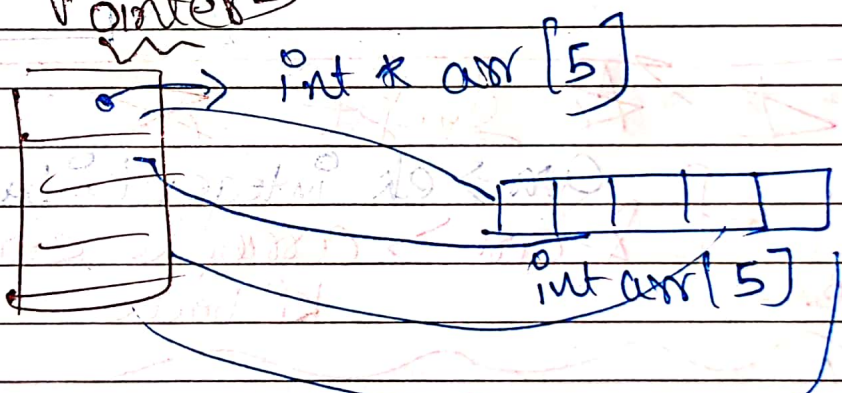
&arr / &ch

else → int/char *ptr[size] = &a (or &arr)

Pointer to an array



Array of Pointer →



Ex → int * arr[4]

int *	int *	int *	int *

arr

num[] = {1,2,3,4}

→ arr[0] → &num[0]
→ arr[1] → &num[1]
→ arr[2] → &num[2]
→ arr[3] → &num[3]

Spiral

8/10/23

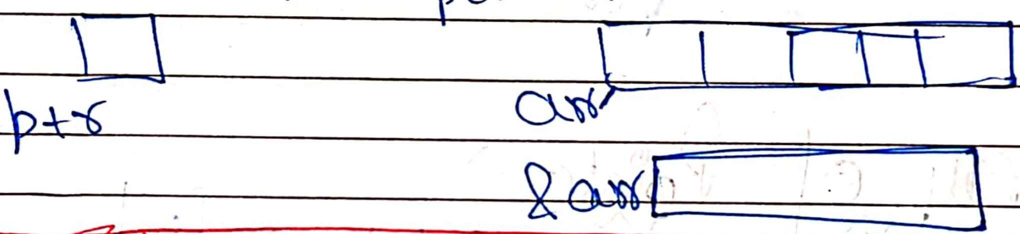
Acess ->

* *arr[0]
* *arr[1]
* *arr[2]

More understand in hash, Graph

Pointer to an Array ->

int * ptr = &num



☆ ☆ Impt ☆ ☆

Diff erence <[arr -> ek integer ki baat
&arr -> array ke starting index ki baat

Pointer & function ->

main ()
{
(416) [| | | | |] -> size -> 20
solve(arr)
}

solve (int arr[])
{
204 [416]
arr ->
↓
size = 8
}

Date 21

8/10/23

main()
{
104 int arr[] = {1,2,3}
1 2 3
arr solve(arr, 3)

-> arr -> 104
->&arr -> 104
}

solve(int arr[], int size)
{
3
size
216 104
arr

arr -> 104
&arr -> 216
}

& arr -> Diff

Q) -> main()
{
216 10 20 30
arr
arr = {10,20,30}
solve(arr)
//print arr
}
-> 11, 20, 30

*arr

solve(int arr[])
{
*arr = *arr + 1
}

416 216
arr

*arr = 10
*arr = 10+1 = 11

Spiral

8/10/23

Date 82

Pointer to Pointer →

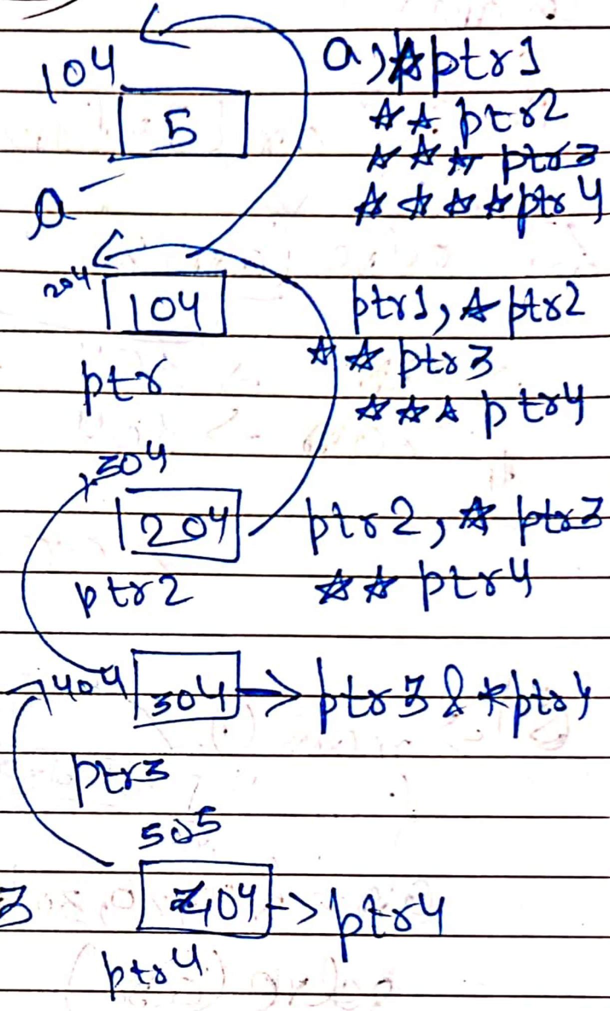
int a = 5;

int* ptr = &a;
Address

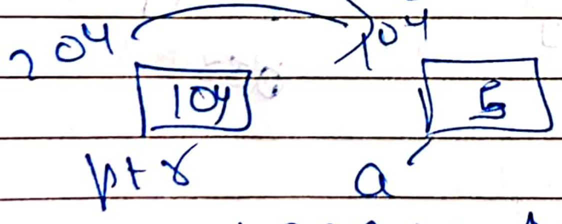
int ** ptr2 = &ptr
address

int *** ptr3 = &ptr2

int **** ptr4 = &ptr3



⇒ block jitna aage hai utne *



1 age *ptr = 5

ex-5

8/10/23

Date 23

Q1 →

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

304 → 204 → 104

204	104	5
q	p (*p)	a

*q

**q

a → 5
&a → 104
p → 104
&p → 204
*p → 5
q → 204
&q → 304
*q → 104
**q → 5

Q2 →

```
main()
{
    int a=5;
    int *p=&a;
    cout << p -> 104
         &p -> 204
         *p -> 5
    Solve(p)
        p -> 104
        &p -> 204
        *p -> 10
}
```

104 | 5 | a

204 | 104 | p

```
Solve(int *p)
{
    (5) +5=10
    *p = *p+5
    p -> 104
    &p -> 204
    *p -> 10
}
```

8/10/23

Date 84

Q3 ->

```
int a = 10;
int *p = &a;
int **q = &a;
```

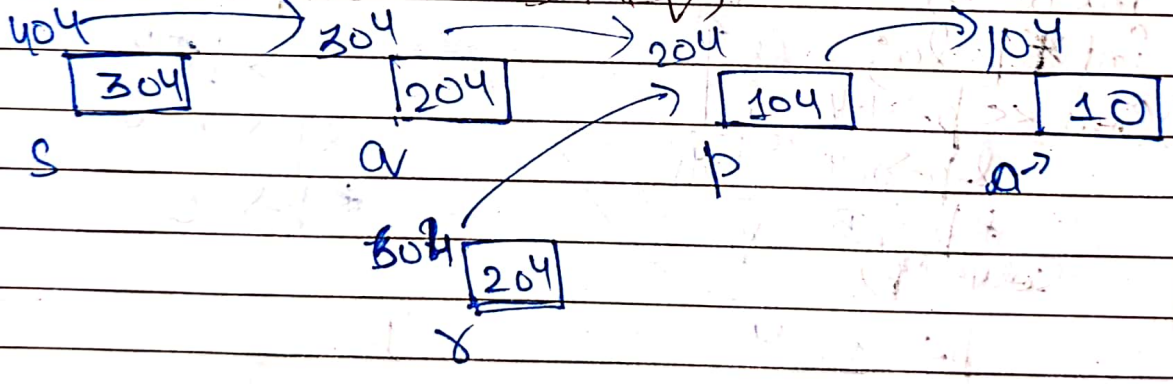
-> Address of Integer

-> Only take Address of pointer

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

Q4 ->

```
int a = 10;
int *p = &a;
int **q = &p;
int **r = &p;
int ***s = &q;
```




```
*s -> 204
**r -> 10
**s -> 10
**q -> 10
**s -> 104
*q -> 104
*p -> 10
p -> 104
&s -> 404
```

```
&r -> 604
&q -> 304
**s + 1 -> 108
```

Spiral

8/10/24

Date 23

Pass by value

4 -> int main()
{
int a = 5;
-> int *p = &a;
p -> 104
&p -> 204
*p -> 5

solve(p)
*p
p -> 104
&p -> 204
*p -> 5

}

104 | 5 | a
204 | 104 | p

*&p

solve(int *p)
{
-> p = p + 1
104 + 1 = 108
p -> 108
204 | 108 | p

int *p -> Pass by value
int *&p -> Pass by Reference