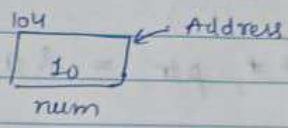
apsara

Date

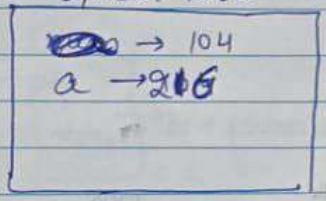
* Pointers → variable → store address of other variable

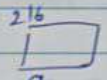
int num = 10




int a = 5

Symbol Table

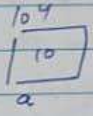



int a = 10

cout << a << endl; → 10

cout << &a << endl; → 104




Note:- In pointer, address hi store hota hai, uske alawa kuch nhi store hota.

Pointer Creation

↳ int *ptr = ~~~

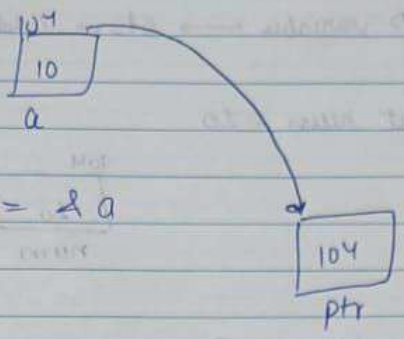
variable name

~~~ address

Pointer to integer data

char *

pointer to char data
~~~

apsara

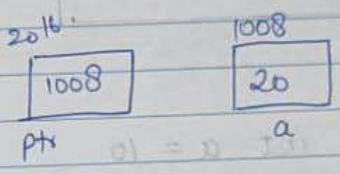
Date

int a = 10




int * ptr = &a

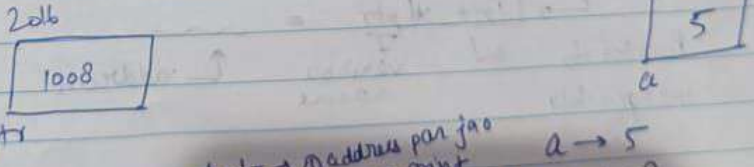
int a = 20
int* ptr = &a




To access values stored at address stored in ptr?

↳ dereference → * → cout << *ptr;

int a = 5
int * ptr = &a




*ptr → ① address par jao
② value print

a → 5
& a → 1008

ptr → 1008
&ptr → 2016

apsara

Date: / /

```
int a = 5
int *ptr = &a
```
sizeof(ptr) → 8

```
char ch = 'k'
char *cptr = &ch;
```
sizeof(ptr) → 8

```
long kchi = 10;
long* lptr = &kchi;
```
↳ sizeof(lptr) → 8

Note: pointer m hamesha address hi store hota hai

```
int *ptr
cout << *ptr
```
→ bad practise

→ int *ptr → Aisa karne par ptr ke andar koi garbage value aajayegi or wo value kisi aise location ki hai jo mere program ko allow nhi hai. Jab m dereference karunga, toh m aise memory location ko access karne ki try kar raha hu jo allow nhi hai. Iss case m run time error aayega ya segmentation fault.

Better way

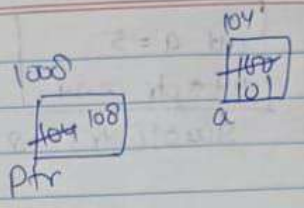
```
int * ptr = 0;
cout << *ptr << endl;
```
↳ It also shows run time error but rather ki hum kisi garbage value ko store kare, uski jagah hum 0 ko store kar dete hai

Page No.

apsara

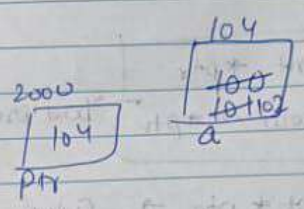
Date: __/__/__

```
int a = 100;
int * ptr = &a
```




```
a = a+1;
ptr = ptr+1;
```

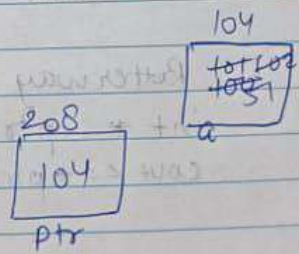
```
int a = 100;
int * ptr = &a;
  a = a+1;
```




```
*ptr = *ptr + 1;
```

↓ Value present at address stored in ptr

```
int a = 100;
int *ptr = &a
```




print
- a
- &a
- *a
- ptr

- *ptr
- &ptr
- (*ptr)++
- *(*ptr)
- *ptr = *ptr/2
- *ptr = *ptr...

apsara

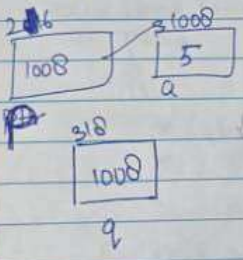
Date: / /

a = 100
&a = 104
*a = Error
ptr = 104
*ptr = 100

&ptr = 208
(*ptr)++ = 101
++(*ptr) = 102
(*ptr)/2 = 51
(*ptr)-2 = 49

④ int a = 5
int * p = &a;
int *q = p;

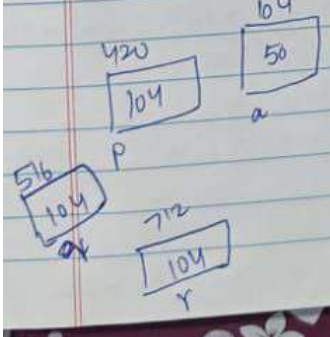
→ a ⇒ 5
→ &a = 1008
→ *a = Error
→ p = 1008
→ &p = 216
→ *p = 5
→ q = 1008
→ &q = 318
→ *q = 5




int * q = *p
↳ Error
*p → int hai
or *q ke andar address store hoga

⑤ int a = 50;
int* p = &a;
int *q = p;
int *r = q

→ a → 50
→ &a → 104
→ *a → Error
→ p → 104
→ &p → 420
→ *p → 50
→ q → 104
→ *q → 50
→ &q → 516
→ r → 104
→ &r → 712
→ *r → 50

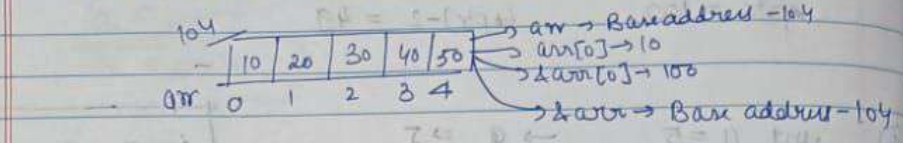


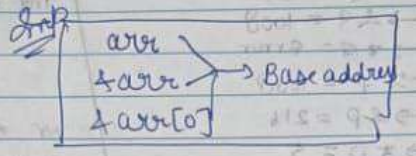
apsara

Date: / /

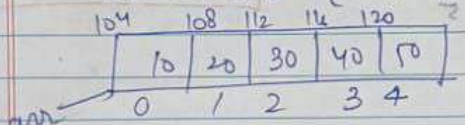
## * Pointer with arrays :=>

int arr[5] = {10, 20, 30, 40, 50}



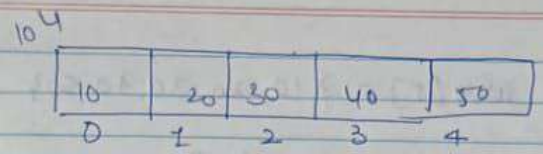

int arr[5] = {10, 20, 30, 40, 50}




print

- arr → 104
- &arr → 104
- arr[0] → 10
- &arr[0] → 104
- *arr → 10
- *arr+1 → 11
- *(arr)+1 → 11
- *(arr+1) → 20
- *(arr+2) → 30
- *(arr+3) → 40

Page No.




int → size → 4

add → 104 + (1) => 4 memory block aage badhna
104 + 4 => 108

* arr ⟷ * arr+0 ⟷ * (arr+0)
* (104)

Value Stored (104) at address → 10

* (arr+0) ⟶ arr[0]
* (arr+1) ⟶ arr[1]
* (arr+2) ⟶ arr[2]
* (arr+i) ⟶ arr[i]
* (i+arr) ⟶ i[arr]
↳ Ye bhi hai

arr[i] is same as i[arr]
↓
behind the scene
arr[i]
↳ * (arr+i)
* (Base address + i)

int a = 5
int *p = &a

| 216 | 104 |
|---|---|
| p | |

| 104 | 5 |
|---|---|
| a | |

p = p+1 ? → chal jayega, lekin me garbage value print hogi

int arr[5] = {10, 20, 30, 40, 50}

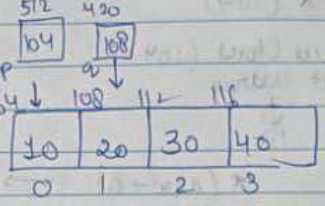
arr = arr + 1 ?

↳ Why not possible in this case?

int arr[4] = {10, 20, 30, 40}

int * p = arr;

int * q = arr + 1;




| | | | |
|---|---|---|---|
| arr → 104 | p → 104 | q → 108 | *p + 1 → 11 |
| &arr → 104 | &p → 512 | &q → 420 | *p) + 2 → 12 |
| arr[0] → 10 | *p → 10 | *q → 20 | *(q) + 2 → 22 |
| &arr[0] → 104 | | | *(q + 4) → Segmentation fault / out of bound |

int arr[4] = {10, 20, 30, 40}

size of (arr) → 4 × 4 = 16

int * p = arr;

size of (p) → 8

char ch[50] = "Love"

char * cptr = ch

cout << cptr;

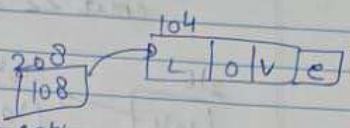
↳ Print entire array ⇒ Love

apsara

Date: __/__/____

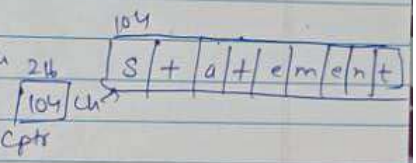
~~cout << cptr~~

1. char ch[50] = "Love"
   char * cptr = ch




→ ch ⇒ Love
→ &ch ⇒ 104
→ ch[0] ⇒ L
→ &cptr ⇒ 208
→ *cptr → *(cptr+0) → cptr[0] → L
→ cptr = Love

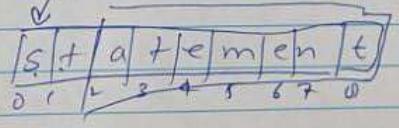
2. char ch[50] = "Statement" 216
   char * cptr = &cptr[0];




→ ch ⇒ Statement
→ &ch ⇒ 104
→ *(ch+3) ⇒ t
→ cptr ⇒ Statement
→ &cptr ⇒ 216

→ *(cptr+3) ⇒ t
→ cptr+2 ⇒
→ *cptr → S
→ cptr+8 ⇒

~~*(cptr+0)+2~~

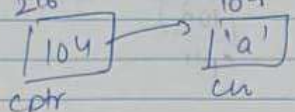



→ cptr+8 ⇒ t

cptr se puri ~~string~~ array print hogi or cptr+8 se 8th character se lekar jab tak null nhi milta.

Page No.: .....

Char ch = 'a'
Char* cptr = &ch
cout << cptr → a ... '\0'




Q. Char *chptr = "Babbar"

↳ Bad Practice

↳ temp storage m store hoga or hum kisi pointer ko temp location ko point kara rahe hai or pta nhi kab tak wo rahe.

Char ch[10] = "babbar"

Char * cptr = &ch → iss case m &ch pure entire 10 character array ka address de rha hai or cptr is a pointer to a char data