

4/10/23

Week → 6

Date 50

Class → 1 Pointers →

variable which store address of other variable

int num = 10;

104 [10]

access through Address

num → 4 byte

Symbol Table

S-T
num → 104
a → 216

216 [5]

a

int a = 5;

Pointers → Syntax →

int *ptr = num
↑
address

pointer to integer
data

(*) → Dereferencing

Spiral

4/20/23

Date 60

char * → pointer to char data

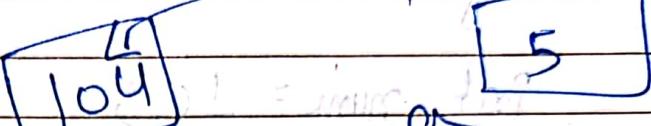
long * → pointer to long data

int * ptr = address;



ptr is a pointer to integer

int a = 5;



int * ptr = &a;

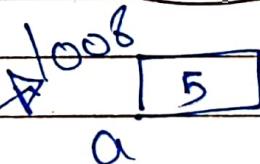
ptr

Access → value stored at address goto that

(3-2) → print it

*ptr → 5

int a = 5;

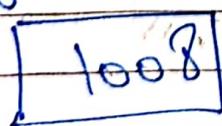


int * ptr = &a;

a → 5

&a → 1008

2016



ptr → 1008

&ptr → 2016

*ptr → 5

Address → pointer & print value

Spiral

4/20/23

Date 6.1

Reference Variable \rightarrow Diff from pointer

Diff name of same variable.

int *ptr = &a;

int ptr = &a;

Invalid

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

char ch = 'A';
char *cptr = &ch;

size of (ptr) \Rightarrow

size of (cptr)

long long ch = 10;
long *ptr = &ch;

\Rightarrow 8

Pointer used in Dynamic
Allocation

Spiral

4/30/23

Date 62

→ Don't do this

Pointers declaration

int *ptr; → Bad Practice

→ [ptr] → garbage value
ptr

Code << ptr → illegal memory allocation

→ Segmentation fault or Runtime Error

Use this → int *ptr = 0; // nullptr;
Always

Nullptr
means No address
against pointer

1

int a = 100;
int *ptr = &a

$$a = a + 1 \\ \text{ptr} = \text{ptr} + 1$$

108

101

104
101
a

108

$$104 + 1 = 108$$

Next Address

Spiral

4/30/23

Date... 63

Note: 1 byte smallest addressable space

(2) \Rightarrow

$\text{int } a = 100$

$\text{int } * \text{ptr} = \&a;$

$a = a + 1 \rightarrow 101$

$* \text{ptr} = * \text{ptr} + 1$

$\rightarrow 102$

10^4

\boxed{a}

10^4

102

ptr

10^4

Value present
at address store
in ptr

(3) \Rightarrow

$\text{int } a = 100;$

$\text{int } * \text{ptr} = \&a;$

208

ptr

104

104

a

point

$\hookrightarrow a \rightarrow 100$

$\hookrightarrow \&a \rightarrow 104$

$\hookrightarrow *a \rightarrow \text{Error}$

$\hookrightarrow \text{ptr} \rightarrow 104$

$\hookrightarrow * \text{ptr} \rightarrow 100$

$\hookrightarrow \&\text{ptr} \rightarrow 208$

$\rightarrow (*\text{ptr})++ \rightarrow 100$

$\downarrow 104$

$\hookrightarrow ++(*\text{ptr}) \rightarrow 102$

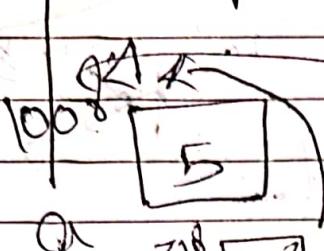
$\hookrightarrow *\text{ptr} = *\text{ptr} + 2 \rightarrow 102 \rightarrow 51$

$\hookrightarrow *\text{ptr} = *\text{ptr} - 2 \rightarrow 51 - 2 = 49$

Spiral

4190123

Date 64

(4) \rightarrow	$\text{inta} = 5_0$ $\text{int } *p = 8a_0$	$\rightarrow a \rightarrow 5$ $\rightarrow 8a \rightarrow 1008$ $\rightarrow *a \rightarrow \text{Error}$
Q14:	 1008 [5] 816 [1008] 318 [1008]	$p \rightarrow 1008$ $*p \rightarrow 816$ $at \ p \rightarrow B$

~~int *q = p ->pointer copy q -> 1008~~

$$q \rightarrow 100\%$$

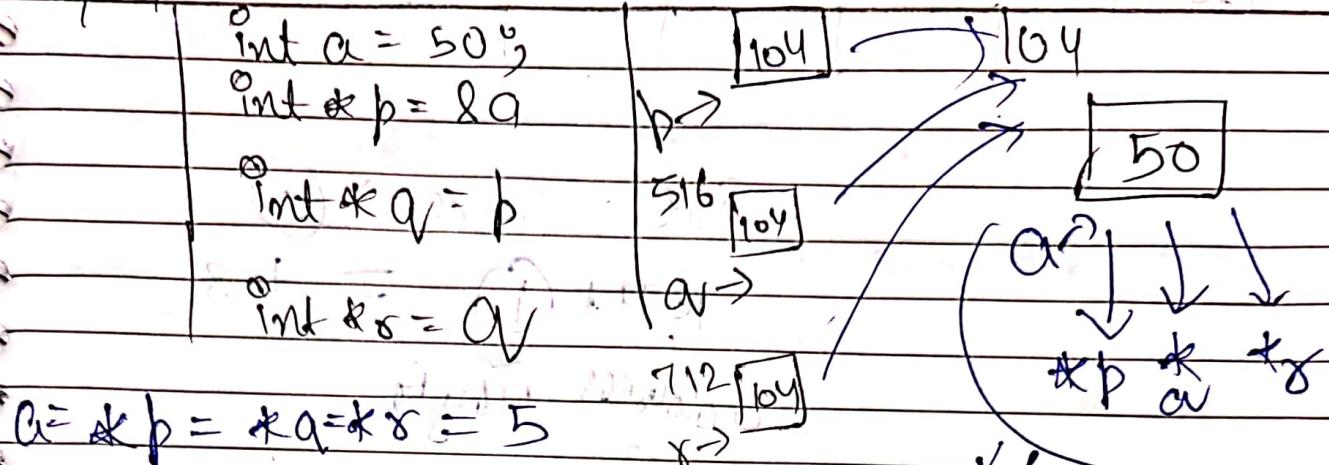
~~skg~~ → iB cD

• `int * a = &b` \rightarrow 5 \rightarrow Integer
 \rightarrow error

$$\begin{array}{ll}
 \text{a} \rightarrow 50 & \text{b} \rightarrow 104 \\
 \text{d} \text{a} \rightarrow 104 & \text{d} \text{b} \rightarrow 420 \\
 \text{c} \text{a} \rightarrow 8500 & \text{c} \text{b} \rightarrow 50
 \end{array}
 \quad
 \begin{array}{ll}
 \text{c} \rightarrow 104 & \gamma \rightarrow 104 \\
 \text{d} \text{c} \rightarrow 510 & \text{d} \text{g} \rightarrow 412 \\
 \text{e} \text{c} \rightarrow 50 & \text{e} \text{d} \rightarrow 30
 \end{array}$$

4/10/23

Date: 65



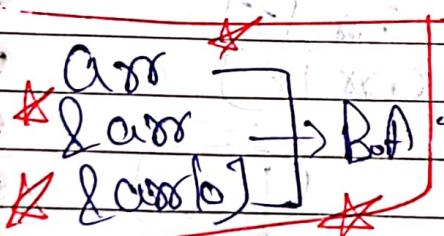
4 ways to access

Pointers with Arrays

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

arr | 10 | 20 | 30 | 40 | 50 |
 0 1 2 3 4

arr \rightarrow 104 (Base Address)



Some \leftarrow C & arr[0] \rightarrow 104
& arr \rightarrow 104

(1) \rightarrow

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

104 108 112 116 120
110 20 30 40 50

point \rightarrow arr \rightarrow 104

arr \rightarrow 104

arr[0] \rightarrow 10

arr[0] \rightarrow 104

*arr \rightarrow 10

*arr+1 \rightarrow 11

*arr+1 \rightarrow 11

*arr+2 \rightarrow 20

*arr+3 \rightarrow 30

*arr+4 \rightarrow 40

Spiral

4/10/23 10⁴ + 1 108 + 1 112 + 1 1B

Date 66

int → size → 4

$$104 + 1 = 108$$

address usage

104 + ② \rightarrow memory block aage

$$104 + 2 = 112$$

$$*arr = *arr + 0 = *(arr + 0)$$

MCQs $*(\text{arr} + i) \rightarrow \text{arr}[i]$

$*(\text{arr} + 0) \rightarrow \text{arr}[0]$

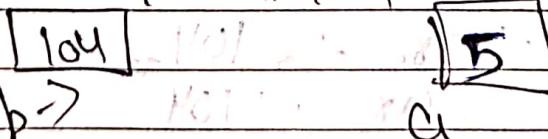
$*(\text{arr} + 1) \rightarrow \text{arr}[1]$

$*(\text{arr} + 2) \rightarrow \text{arr}[2]$

Note: int a = 5;

int *p = &a;

216 \rightarrow 104



$p = p + 1 \rightarrow ? \rightarrow \text{val}$

Can we do $a = arr = arr + 1 \rightarrow \text{Error}$

4/30/23

10⁴

10

Date.....

67

Doubt S

int a = 10; a^{10⁴}

int a = Error

int arr = [10, 20, 30]

arr = 10⁴

arr

But a → assign
to int

Not address

(arr) → address
se assign
hai

ya de hogi

a

arr[0]

Q) ->

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

int *p = arr;

int *q = arr + 1;

10 20 30 40 50

arr

512

104

512

104

420

1108

arr → 104

& arr → 104

arr[0] → 10

arr[0] → 104

p → 104

& p → 512

p → 10

p → 104

q → 108

& q → 420

q → 20

q → 104

p + 1 → 11

p + 2 → 12

p + 1 + 2 → 22

(p + 4) → 124

ya koh garbage value ←

print 0x

array out of bound
(Segment fault)

Spiral

4) 10/23

Char pointer

Date... (68)

char [50] = "love"

Put arr = (10, 20)

int *p = arr

cout << p -> 104

cptr -> love

512
104
cptr

104
10 | 20
arr ->

Ex - char ch [50] = "love";

char *cptr = &ch[0];

ch -> love

&ch -> 104

ch[0] -> l

&cptr address

*cptr -> l -> *(cptr + 0) -> cptr[0]

ch -> love

char ch [50] = "statement";

char *cptr = &ch[0];

ch -> statement

&ch -> 104

* (cptr + 3) -> f

cptr -> statement

&cptr -> 216

* (cptr + 3) -> f

104
10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
ch 0 1 2 3 4 5 6 7 8

cptr + 2 -> statement

*cptr -> s

cptr + 8 -> f -> spiral

4/10/23

Date (69)

104

③ $\text{char} = \text{&a}^{\circ}$

$\text{char * ch} = \&\text{ch}[0]$

$\text{cout} < \text{ch}$

104
ch

↳ a unununun print jab null character milte

NOTE: Null character milega ya infinite tak
Nhi jayegaa

④ $\text{char ch}[0] = \text{&a}^{\circ}$

$\text{char * c} = \text{ch}$

$\text{char * ch} = \text{&a}^{\circ}$

↳ It is a possible
But Bad Practice

Stores in
Temp location

Note: $\&\text{ch} \rightarrow$ address of all the
characters present in ch

Spiral