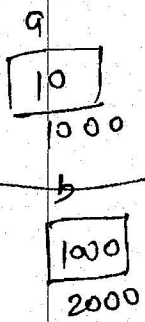
Dev

"Pointers" :-

Ex: ①

1. int a = 10
2. int *b = &a
3. Pf(a) = 10
4. Pf(b) = 1000
5. pf(*b) = 10

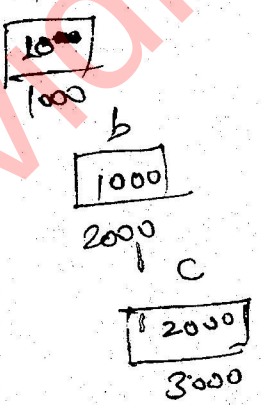


Inside b address so 8 Bytes m/y required.

* value at 'b' 1000

Ex: ② :-

```
int a=10
int *b = &a
int **c = &b
Pf(a) = 10
Pf(*b) = 10
Pf(*(*c)) = 10
```



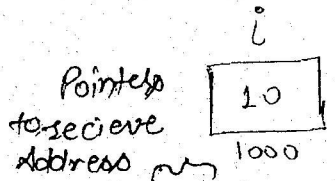
Pf(a) i.e value inside a

Assume, By default m/y Address size = 8B (for class only)

Ex: ③ :-

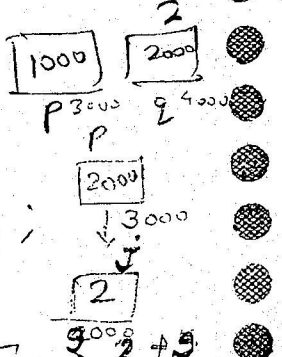
```
# include <stdio.h>
int i=10, j=5;
main()
{
    f(&i, &j);
    Pf(i, j);
}
```

address sending



f(int *p, int *q)

```
{ p = q;
  *p = *q;
  *q = *p + *q;
}
```



O/P:- 10, 4

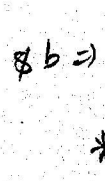
StudentSuvidha.com

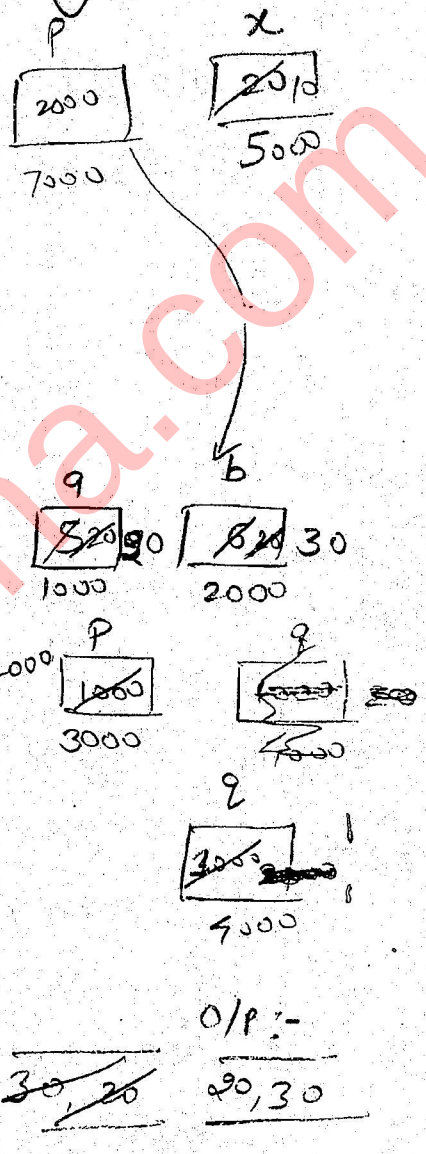
Download all Notes and papers from StudentSuvidha.com

★ 'Preprocessor (#) purpose' is to remove all shortcuts by Actual definition.

Que:- Consider the following 'C' program

```
void f(int x, int *p)
{
    *p = x;
    x = 10;
}
int main()
{
    int a = 5, b = 6;
    int *p = &a, **q;
    *p = 20, q = &p;
    f(a, &b);
    *q = &b;
    *p = 30;
    pf(a, b);
}
```





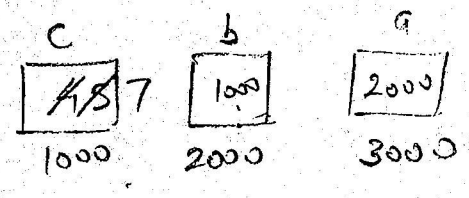
O/P:- ~~30, 20~~ 20, 30

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

Que:- void main()

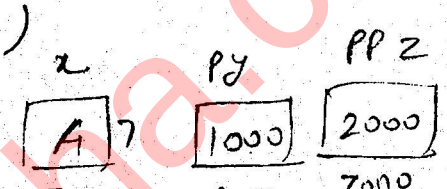
```
{ int c, *b, **a;
  c = 4, b = &c
  a = &b;
  Pf("%d", f(b, a));
}
```



by default

```
int f(int x, int *py, int **ppz)
```

4 → c, 1000 → b, 2000 → a

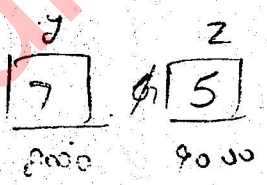



```
{
  int y, z;
  **ppz += 1;
  z = **ppz;
  *py += 2;
  x += 3;
  return (x+y+z);
}
```

(unary operator) Higher Priority → `*pz = **ppz+1` ← (Binary operator)

operand > operator Priority > Priority

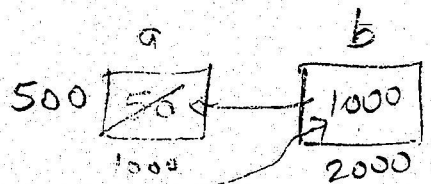
y = *py;



7 + 7 + 5 = 19

Que:- Consider the following C program :-

```
main()
{
  int a = 50;
  int *b = &a;
  Scanf("%d", b);
  Printf("%d", a + 25);
}
```



500 ← input

o/p: 500 + 25 = 525

o/p: 50 + 25

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

Que :-

```
int f(int *a, int n)
{
  if (n<=0)
    return(0);
  else
  if (*a % 2 == 0)
    return *a + f(a+1, n-1);
  else
    return *a - f(a+1, n-1);
}
main()
{
  int a[] = {12,7,13,4,11,6};
  printf("%d", f(a,6));
  return(0);
}
```

1000 6

a: 1000 (9000) n: 6 (10000)

Binary

*a → unary (Higher Priority)

12 % 2 == 0 ✓

12 + f(1002, 5)

7 − f(1004, 4)

13 − f(1006, 3)

16 + f(1008, 2)

12 − f(1010, 1)

6 + 0

a	12	7	13	4	11	6
	1000	02	04	06	08	10

o/p : 15

In Array

$a[i] = *(a+i)$

Base Addr.

a[i] ← shortcut

*(a+i) ← Actual

$$a[i] = *(a+i) = *(i+a) = i[a]$$

In Arrays

1000

a = a+4 — error

1008

assignment

Base address can't be changed.

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

b ← a[i][J]

Shortcut ⇓

b[J]

⇓

*(b+J)

⇓

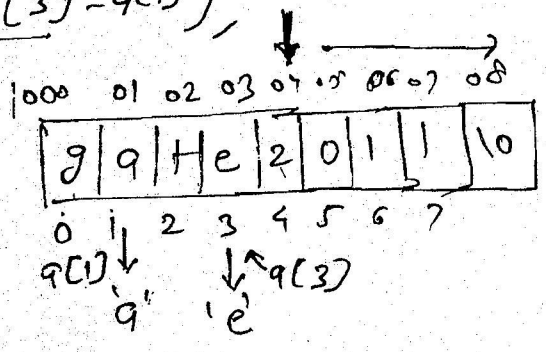
*(a[i]+J)

⇓

((a+i)+J)

Que :-

```
main()
{
    char a[9];
    char a[9] = "gate 2011";
    Printf("%s", a+a[3]-a[1]);
}
```



o/p :- 2011

e - a

101 - 97 = 4

a+4

1000+4 = 1004

1. Pf("%s", "gate2011");

⇒ gate2011

2. Pf("%s", 1000); from unit 1000

⇒ gate2011

3. Pf("%s", a); gate2011

4. Pf("%s", a+2); te2011

1000+2

⇒ 1002

char

5. Pf("%c", * (a+2)/1000); ⇒ t

1002

6. Pf("%s", * (a+2)/1000); X (wrong)

1002

StudentSuvidha.com

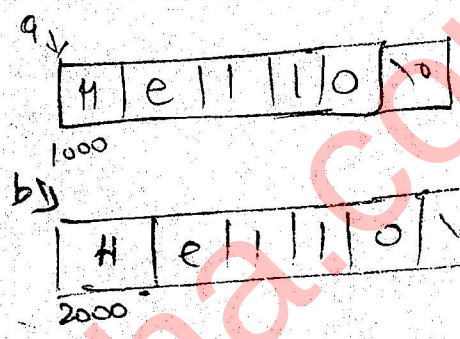
Download all Notes and papers from StudentSuvidha.com

8.) pf("%d", *(a+3)) ⇒ e (101) ↑ ASCII
1000 1003

:— * Strlen ("abcd") = 4 (Excluding Null)

:— * Sizeof ("abcd") = 5 (including null)

```
main()
{
    char a[] = "Hello";
    char b[] = "hello";
    if (a==b)
        pf ("hi")
    else
        pf ("bye")
}
```



(1000 == 2000) ✗

o/p :- "bye"

if (a==b) ⇒ error
Base Addr can't change

if (*a == *b)

H == H True

o/p ⇒ "Hi"

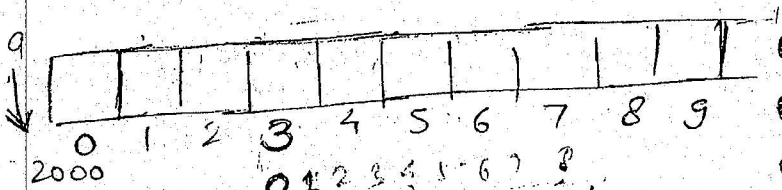
StudentSuvidha.com

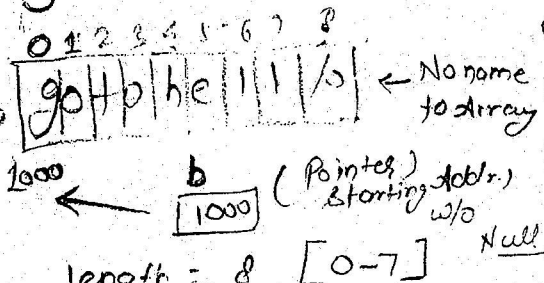
Download all Notes and papers from StudentSuvidha.com

Date 10/2017

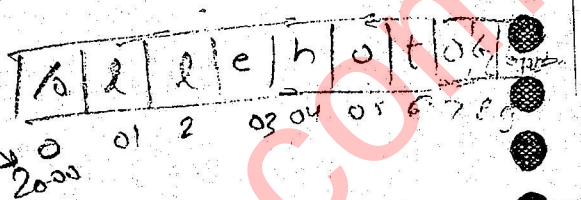
Ex:

```
main()
{
    char a[10];
    char *b = "go to hell";
    int length = strlen(b);
    for(i=0; i<length; i++)
    {
        a[i] = b[length - i];
    }
    printf("%s", a);
}
```





variable



b[8] → \0

8 - 0 = 8

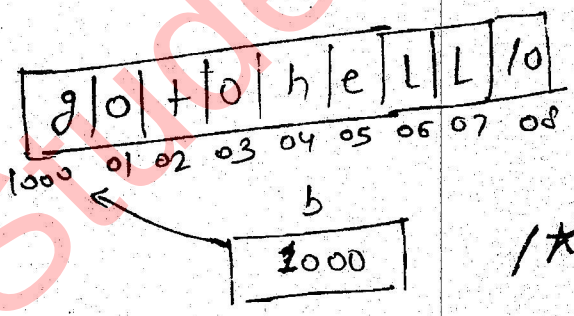
O/P Null

Starting \0 so no string printed

Note :-

"%s" Prints the data from given starting address to until '\0' symbol (Null) not found.

In Array a[] \0 is stored in starting so nothing Printed. i.e No output



/* Here b is a Pointer (char) variable seperate from Array */

Note :- Array Base Address can not be changed but Content can be changed.

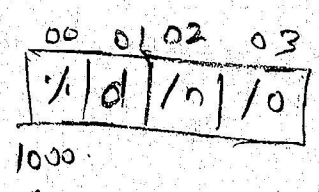
"String constant" Contents can not be changed because Result is undefined.

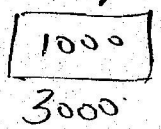
StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

Que:-

```
main()
{
  char *a = "%d\n";
  a++;
  a++;
  Printf(a-2, 300);
}
pf("%d\n", 300);
```





O/P:- => 300

Que:-

```
main()
{
  int a[] = {50, 70, 20, 10, 60, 40};
  int *b[] = {a+3, a+1, a, a+4, a+5, a+2};
  int **c = b;
  c++;
  Printf("%d %d %d", c-b, *c-a, **c);
  Printf("%d %d %d", c-b, *c-a, **c);
  ++*c;
  Printf("%d %d %d", c-b, *c-a, **c);
  ++**c;
  Printf("%d %d %d", c-b, *c-a, **c);
}
```

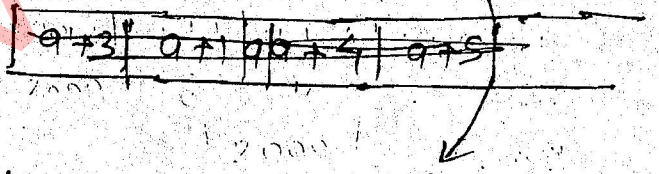
b is an array where every one int pointer

(1, 1, 70)

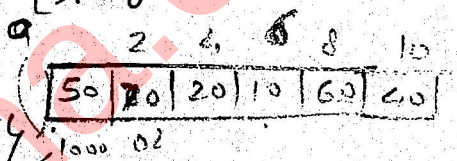
(2, 0, 50)

(2, 2, 70)

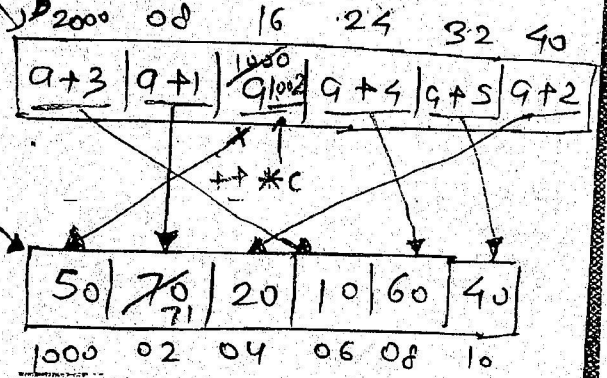
(2, 1, 71)



[Array int data]



[Array int Addr.]



C: 2000 (3000)

C++: 2008

C++ 2016

StudentSuvidha.com

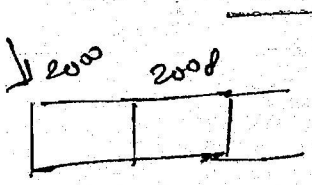
Download all Notes and papers from StudentSuvidha.com

difference b/w two pointers will give # of elements b/w them excluding last one.

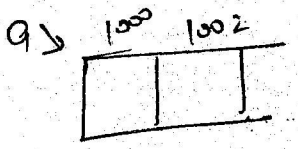
ex: a & b are two pointers.

a - - - - b

b - a ⇒ # of elements before b



2008 - 2000 = 8/8 = 1 element (addr. array) int



1002 - 1000 = 2/2 = 1 element (int dat array).

Note :- * & and ++ both are unary operators which are having same priority but Associativity is Right to left.

ex: *c++;

Note :-

* i) we can add integer constant to pointer variable.

Ex :- P+5 = skipping five elements from P (including P) in forward direction.

* ii) we can subtract integer constant from pointer variable

Ex: P-5 = skipping five elements from P (including P) in backward direction.

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

★ (iii) we can "Subtract" two pointer variables

Ex:- $P_2 - P_1$ = # of elements present from P_1 to before P_2.

[Both Pointer should point to same Array]

[can't Add/mul/Div two Pointer (Error will come)] because there is no meaning.

~~(iv)~~

Que:-

```
main()
{ float b[] = { 70, 60, 50, 20, 40, 30 };
  float *a[] = { b+3, b+1, b+4, b, b+5, b+2 };
  float **c = a;
  ++*++c;
  Pf (c-a, *c-b, **c)
   1     2     50
  ++**c;
  Pf (c-a, *c-b, **c)
   1     2     51
  *++*(c+2);
  Pf (c-a, *c-b, **c);
}
```

b

70	60	50	20	40	30
1000	04	08	12	16	20

a

1020	1008	1016	1004	1020	1008
2000	08	16	24	32	40
b+3	b+1	b+4	b	b+5	b+2

c

2000
5000

⇒ 2008 → 2024

2024 but not changed

2024 - 2000

1, 2, 51

⇒ (1, 2, 50)

(1, 2, 51)

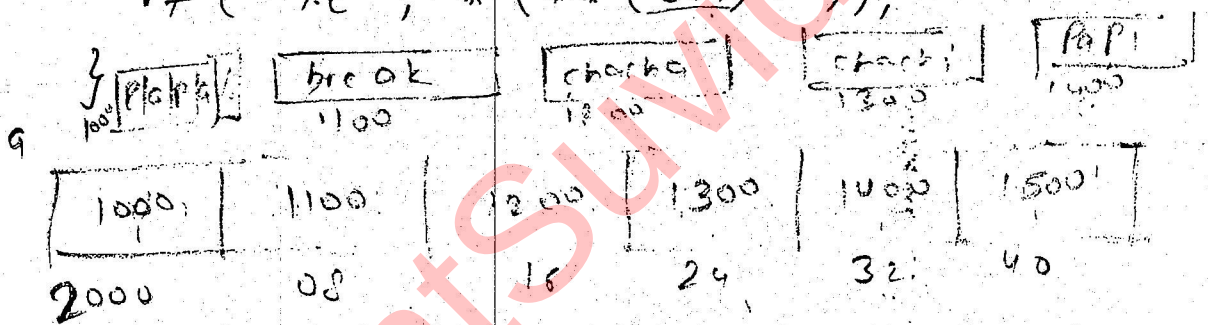
(1, 2, 51)

/* c has no assignment so c value ω o

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

```
main()
{ char *a[] = { "Papa", "break", "chacha", "chachi",
                "Papi", "stupid" };
  char **b[] = { a+3, a+4, a+2, a, a+5, a+1 };
  char ***c = b;
  Pf("%s", *(*(c+2)+2));
  Pf("%s", **c+3);
  Pf("%s", *c[1]+2);
  Pf("%c", *(**(c+1)+2));
}
```




```
Pf("%d", Pf("%s", "abcd"));
```

O/P:- abcd4

O/P:

1. Papi
2. chi
3. Pi
4. P

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

Date 07/10/2017

```
main()
{
  int a[5][3] = {10, 20, 30 ---- 150};
  Pf("%d", (a == a[0]) && (*a == a));   => 1
  Pf(" *(*(a+3)) ", *(a[3]) + 2);
  Pf( * ((a+2) + 3) + 2);
  Pf( * a[2] + 2);
}
```

a[0] = *(a+0) — 1st row selected

(a == a[0]) — 0th row selected

(*a == a) — *(a+0)

3 row skipped

3rd row skipped, selected → 1018

2 row skipped → 1018 + 3 row skip → 1030

Skip 2 row

Select ⇒ *(1012) + 2 ⇒ 70 + 2 ⇒ 72

a	10	20	30	40	50	60
	1000	1002	1004	1006	1008	1010
			r_1		r_2	

Skip one Row: a ⇒ 1000, a+1 ⇒ 1002

Next Row → 1006

Skipping 2 1D's or 2 rows: a+2 ⇒ 1012

row major

a		
1000 10	1002 20	1004 30
1006 40	1008 50	1010 60
1012 70	1014 80	1016 90
1018 100	1020 110	1022 120
1024 130	1026 140	1028 150

a+1 →

skip 2 rows

a+2 ⇒ 1012

*(a+2) = 1012

*(a+2) + 1 — Row selected; one element skip

((a+2)+1) = 80

a+2 = 1012

(a+2)+1 = 1018 — 2 rows skip; 1 row skip

*((a+2)+1) = 1018 — 3rd row selected

1 element

*((a+2)+1) + 1 — 3rd Row 1018 = 1020

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

pf (* ((a + 2) + 3) + 2)

1000 + skip 2 rows → 1012 + skip 3 rows

1012

1012 + 18 = 1030

* (1030) + 2

row selected

⇒ 1034 ⇒ out of array Garbage value

pf (* a[2] + 2);

* (a + 2)

skip 2 rows

1000 + 12 ⇒ * (1012) selected

* (1012) + 2

70 + 2 = 72

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

Que main()

{

int a[7][5][3];

Pf(*a + 3 + 5);

Pf(**((a+2)+3));

Pf(*((*(a+2)+3)+2)+2)

}

1000

1048

1066

1078

1000 1018 1048

1st 2D — 18B — 30B

1060 — 2 2D

7 2D's ⇒ 30B 2D

5 rows ⇒ 1 row

3 cols ⇒ 3×2 = 6B

6B 6B

1000 + 3 + 5 30B

1048

1060 + 3 ⇒ 1066

row

3rd 2D selected

1st row

1060

1078 + 12

= *(1090)

row selected

a ⇒ 1000

*a ⇒ 1000 0th 2D selected

**a ⇒ 1000 0th Row (In 0th 2D)

***a ⇒ value

a+1 ⇐ 1 2D skipped ⇒ 1030

*(a+1)+1 ⇒ 1030 + 6B = 1036

1 row skipped — 2nd 2D selected

*a ⇒ 1000

*a+1 ⇒ 1210

3D

210 element

**((a+1)+1) ⇒ 1036 + 2B ⇒ 1038

1 element skipped — *(a+1)+1

*** a+1 ⇒ value at (1000) value

value+1 let 50 + 1 = 51

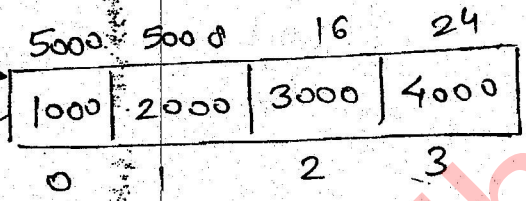
StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

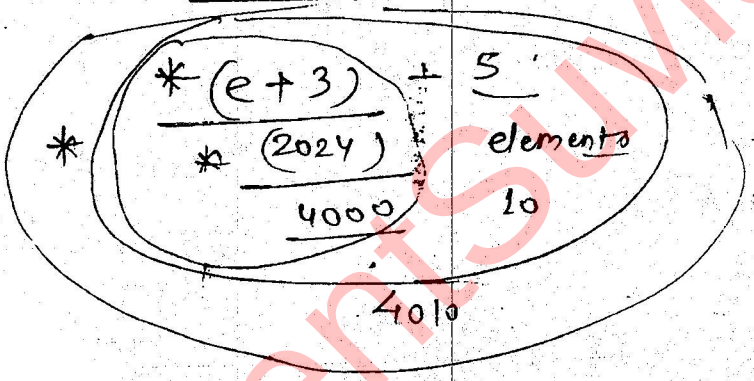
Que:-

```
int a[5] = { 10, 20, 30, 40, 50 };
int b[3] = { 11, 21, 31 };
int c[1] = { 12 };
int d[7] = { 13, 23, 33, 43, 53, 63, 73 }
           0   1   2   3   4   5   6
int *e[4] = { a, b, c, d };
            0  1  2  3
```

:-



e[3][5] = 63



This is the Example for creating ~~4 rows 6 col~~ 2D Array with 4 Rows and Non uniform Column

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com

Que Consider following C-declaration

```
int a[10][5];
int *b[10];
```

T/F :-

a) a[5][2] = value ✓ (Assignment)

b) a[5] = address ✗ [base addr of 5th row can't change]

c) b[5] = address ✓

d) b[5][3] = value ✓

StudentSuvidha.com

Download all Notes and papers from StudentSuvidha.com