

7/9/23 Week → 3 Class → 1

APCO
Date: 7/9/23

Array Level → 1

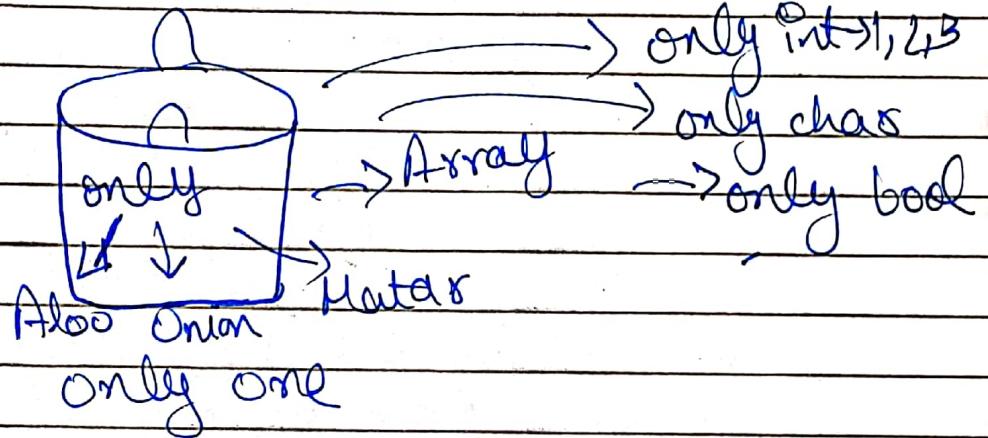
What is Array? → List of similar items

↳ Data Structure

Collection of elements

Continuous Memory block

Mummy



In Array → 1, 2, 6 arr true → X

Why? → 2 no item

int a ↴ ↗ int b

10 no → Sum n1

n2

n3

n4

n5

n6

{n1 to n6}

8/9/23

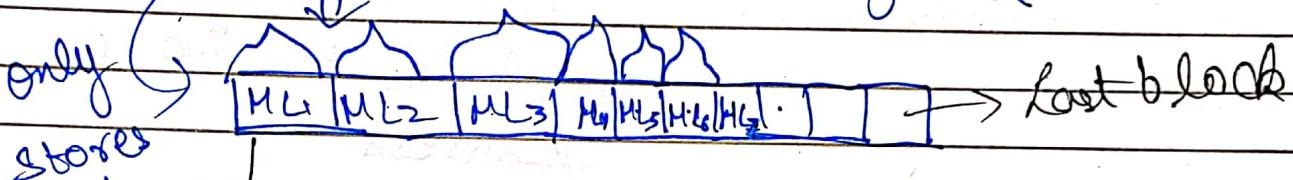
APCO
Date: 8/9/23

loopno → n1
n2
↓
n100

→ loopno → Will you declare 1000 no one by one

use Array → create 10,000 memory location in Single line

Ex- `int arr[10];` → Name of Array in Single line
only stores arr



Continuous Memory → 1 line back
base address, starting block

1 int → 4 byte

10 int → 4 * 10 → 40 byte

Syntax → `int arr [10];` → static array creation

only 40 byte space allotted

⇒ `int a = 5`

user today like
in date to name a
how

a ↗ 5

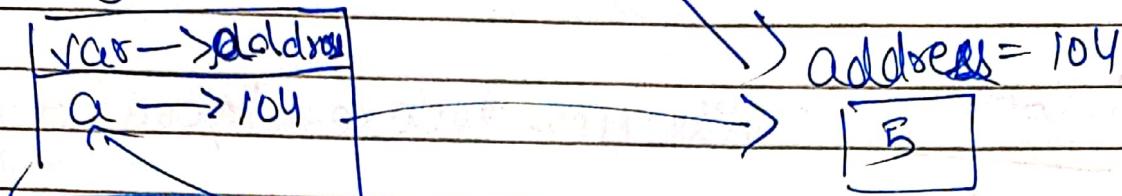
Item & may
all
Par in Reality
↳ P.T.O

8/9/23

APCO
Date: 136

int a = 5

Symbol table

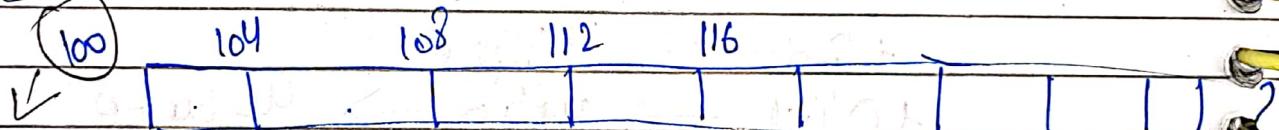


cout << a

Ex → Some Windows → 0x0

you have to give exact Address

Ex → int arr [10];



Base Address

Address = Base Address + index * Size of Data Type



Address of operator → 8



Don't confuse with & operator

Ex → int a = 5

cout << &a ;

8/9/23

APCO
Date: 8/17

int arr[10];

cout < arr; } cout < &arr

give Base Address

→ Size of → give size of Data Type or
Data Structure

★ → In C++ → length doesn't exist with
array but vectors have → length But
JS & Java array contain → length fun.

Array Initialization →

int arr [] = { 1, 3, 2, 6, 8 }; → get size automatically with value

int brr [5] = { 3, 2, 4, 6, 9 };

int brr [5] = { 2, 4 };

↓ ↗ → No Insert
2 1 4 1 0 0 1 0

while
initialization

→ int drr [2] = { 2, 4, 4, 6, 4 };

2 1 4

→ ERROR

8/9/23

APCO
Date: 1/10

Custom input while creating Array

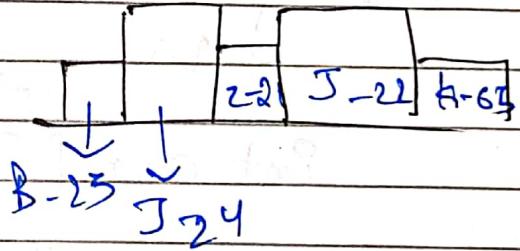
Putting
Custom
Input in]; \Rightarrow Some Compilers
give error, some Not

\rightarrow Bad Practice \rightarrow Because every programme have certain amount memory & maybe you are requesting more memory than allotted.

To solve this use

Dynamic Array (Vector)

Indexing In Array \rightarrow



Send B-25
Amazon to?



To access particular

block of array Address

we use

because we cannot remember

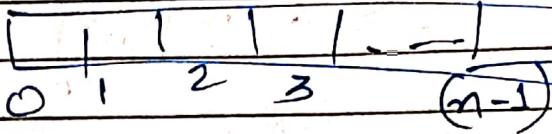
Hexadecimal address

2/9/23

APCO

Date: 339

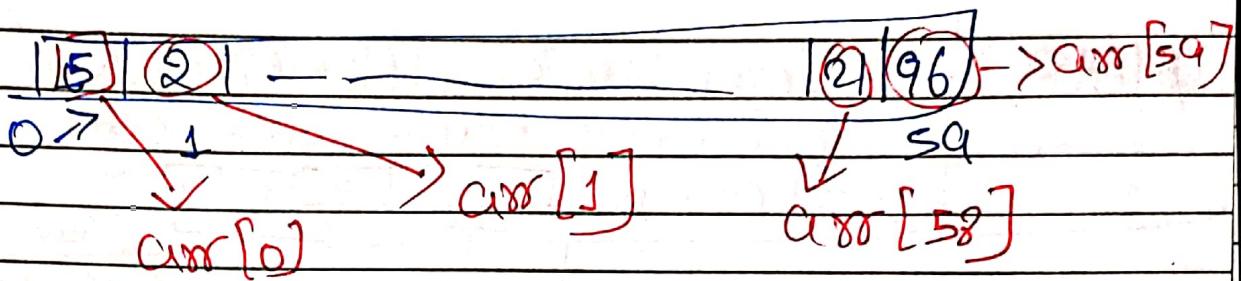
Array $\rightarrow n - \text{size}$



Index $\rightarrow 0$ to $n-1$

ex1 \rightarrow int arr [60] $n = 60$

Index $n-1 = 59$

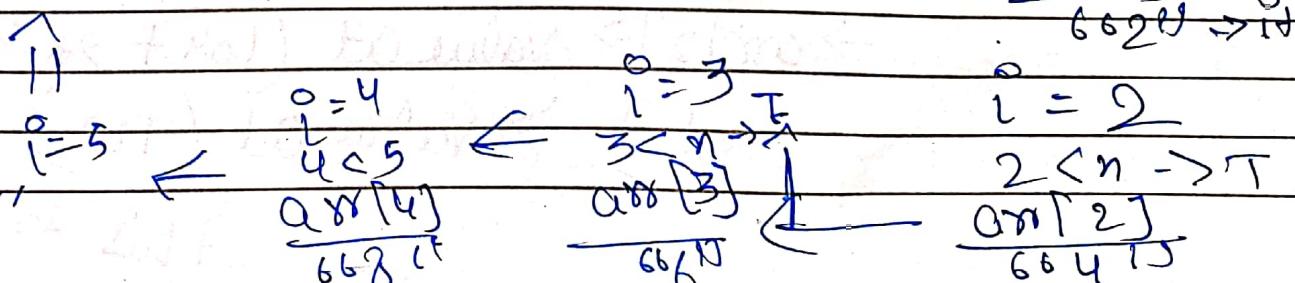


ex2 \rightarrow int arr [5] = {3, 5, 8, 9, 12}

0	3	\rightarrow arr[0]	
1	5	\rightarrow arr[1]	
2	8	\rightarrow arr[2]	int n = 5;
3	9	\rightarrow arr[3]	\rightarrow n 5
4	12	\rightarrow arr[4]	

\rightarrow for (int i=0; i < n; i++)
cout << arr[i];

exit from loop

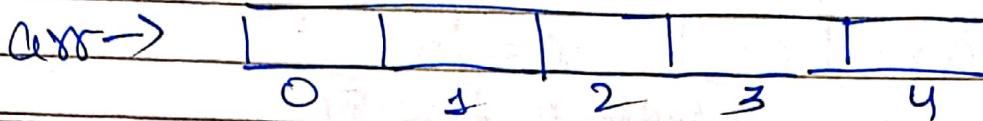


3/9/23

APCO
Date: 190

→ taking input in an Array

int arr[5];



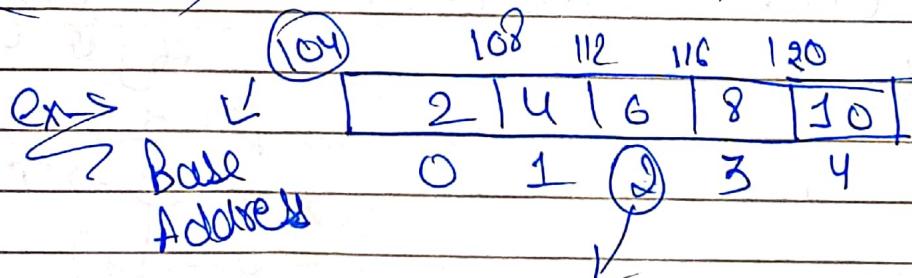
Note → If array me kuch bhi nhi daala / ek bhi value nhi → garbage

→ If array [2] 4 1 0 0 0 have some value then Baki value zero hogi

→ cin >> arr[0]
cin >> arr[1]
cin >> arr[2]
cin >> arr[3]
cin >> arr[4]

int m=5
for (int i=0; i<n; i++)
cin >> arr[i]

Address = Value (Base Address) + (index * data type size)
(arr[i]) → at



→ arr[2] → Value at (104 + 2 * 4)

arr[2] → Value at (112)

This address

PS → Problem Statement

8/9/23

APCO
Date: 1/21

(PS)

- to size array → `int arr[10];`
- take i/p in that array
- double each value of that array & print

$n = 10$

`for (i=0; i<n; i++)`

$cin >> arr[i]$

`for (i=0; i<n; i++)`

~~& arr[i] = 2 * arr[i];~~
~~cout << arr[i];~~

(Q2) →

5 size array → `int arr[5];`

5 i/p

$n = 5$

→ total Sum print

`for (i=0; i<n; i++)`

`int sum = 0`

$n = 5$

`for (i=0; i<n; i++)`

$sum = sum + arr[i];$

$cin >> arr[i];$

$\leftarrow arr[0]$

$\leftarrow arr[1]$

$\leftarrow arr[2]$

$\uparrow \leftarrow arr[3]$

$\downarrow \leftarrow arr[4]$

→ Work only if array don't any empty box

$n = \underline{\text{size of (arr)}}$ → Total

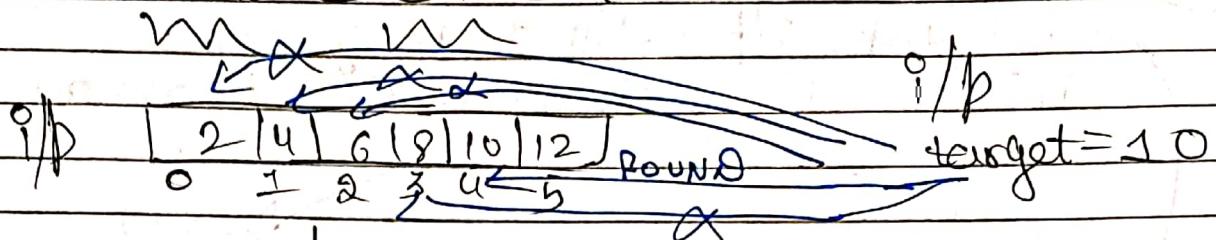
$\text{size of (arr[0])} \rightarrow \text{size of one}$

$$n = \frac{5 \times 4}{4} = \frac{20}{4} = 5$$

8/9/23

APCO
Date: 1/22

Linear Search → Array is Sorted/Unsorted
↳ Cond'



O/P → found 10 or Not

Arrays & functions

int main ()

int arr (5)

solve (arr) size);

y

bitne elements

Store hui

1 - n Array
↑
Solve (int arr),
int size)

Y

Q) → Count 0's & 1's in an Array

| 104 |

| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
Ans ↑ 0 ↑ 1 ↑ 2 ↑ 3 ↑ 4 ↑ 5 ↑ 6 ↑ 7

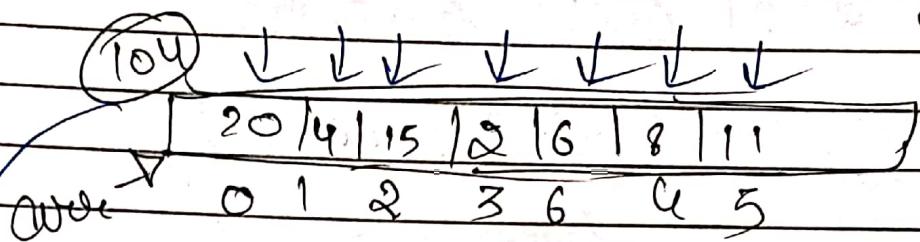
| 0 1 2 3

zero count
↓
3

| 0 1 2 3 4 5

one Count
↓
5

8/9/23

APCO
Date: 1/25(1) \rightarrow Minimum No in an array

Signed int

$$\hookrightarrow -2^{31} \rightarrow 2^{31}-1$$

limits.h \rightarrow INT_MIN \hookleftarrow INT_MAX

Best Practice \rightarrow Min no \Rightarrow min Ans
 Min Ans
 Initialize

\hookrightarrow Max no \rightarrow max Ans

INT MIN

INT MAX

because \rightarrow INT_MAX \rightarrow so chose No hi
 in array

\rightarrow int min Ans = INT_MAX $\textcircled{2}$

(A) \rightarrow int min Ans = INT_MAX;

8/9/23

APCO

Date: 12/4

(B) \rightarrow for ($i = 0$; $i < n$; $i + 1$)

{ if $arr[i] < \minAns$

$\minAns = arr[i]$

}

}

(B.2) \rightarrow for ($i = 0$; $i < n$; $i + 1$)

$\minAns = \min(i : 0 \text{ to } (arr[i], \minAns))$

↓

built-in function

}

(3) \rightarrow Reverse an array

i/p

arr \rightarrow [10 | 20 | 30 | 40 | 50 | 60]
0 1 2 3 4 5

o/p ->

[60 | 50 | 40 | 30 | 20 | 10]
0 1 2 3 4 5

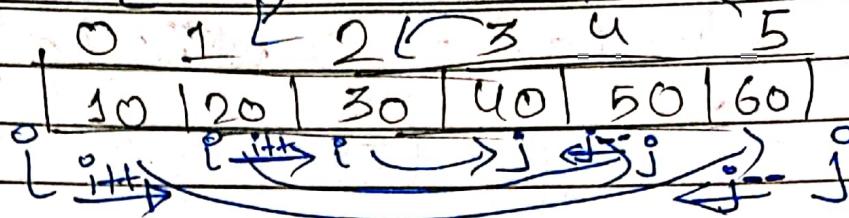
Even Call

8/9/53

APCO

Date: 1/12/55

Even Case →



Swap(a_i, a_j) → Inbuilt function

ex: Swap(arr[0], arr[5])

$i \rightarrow \text{Left} +$

$j \rightarrow \text{Right}$

Left = 0, Right = 5, Swap(i, j)

Two Pointers

Approach

[60 | 20 | 30 | 40 | 50 | 10]

$\rightarrow L++$ Left(L) Right(R) $\leftarrow R--$

[60 | 50 | 30 | 40 | 20 | 10]

$\rightarrow L=1, R=4$, Swap(arr[L], arr[R])

$L++, R--$

$\Rightarrow L=2, R=3 \rightarrow \text{swap}(arr[L], arr[R])$

[60 | 50 | 40 | 30 | 20 | 10]

R L → stop

$L++, R--$

If [$L > R$] → stop

8/9/23

APCO

Date: 1/26

for Loop \rightarrow while Loop

for ($\text{int } i = 0; i < n; i++$)

While loop \rightarrow $i = 0 \rightarrow$ Initialization

while ($i < n$)

\rightarrow Condⁿ

// logic of Loop

$i++; \rightarrow$ updation

Ex Counting $i \rightarrow N$
for-loop

for ($\text{int } i = 0; i < m; i++$)

$\text{cout} \ll i;$

\downarrow

while Loop
 $\text{int } i = 1$
while ($i \leq N$)

$\text{cout} \ll i;$
 $i++$

left \leq right

left $>$ right

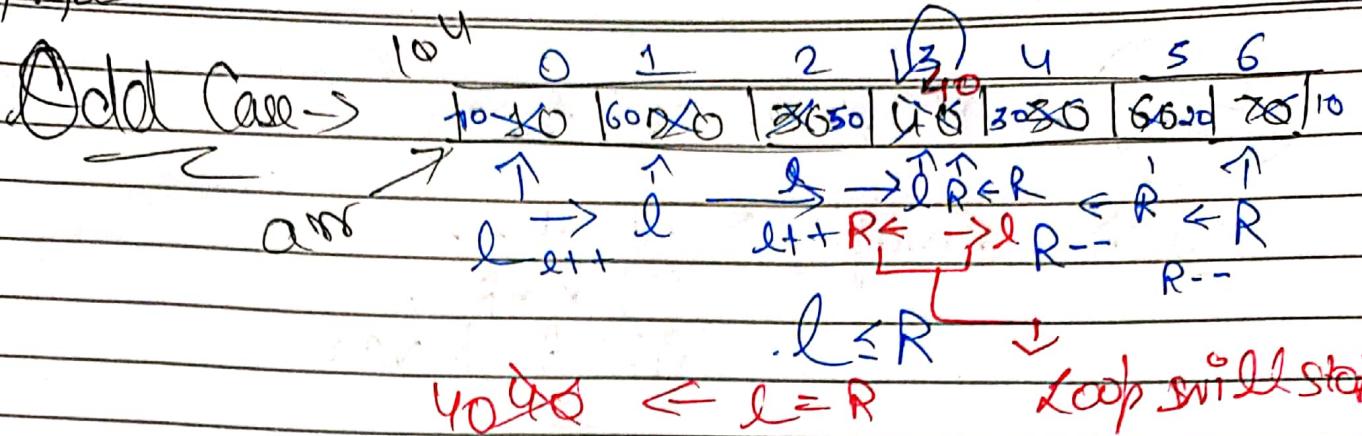
\downarrow
Stop loop

Loop
Condⁿ

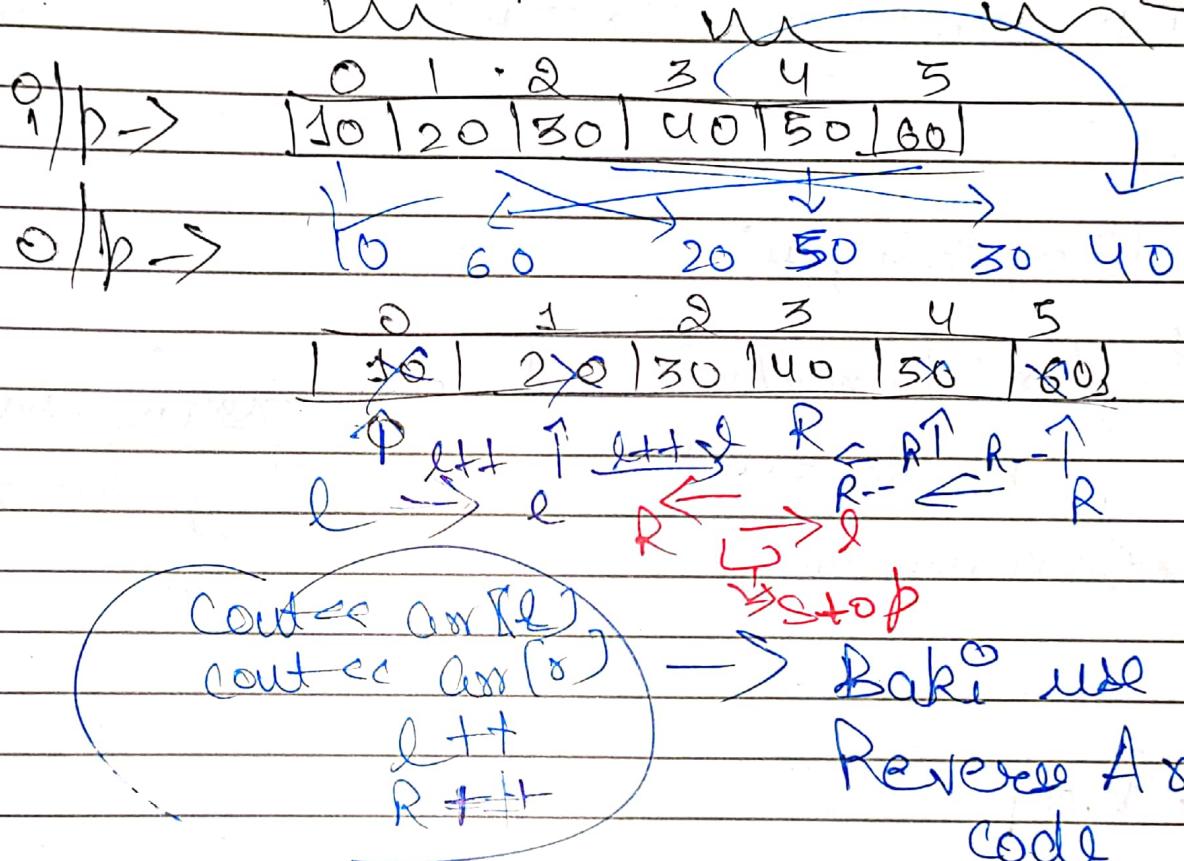
nahi Rakhna
chalta h

8/1/23

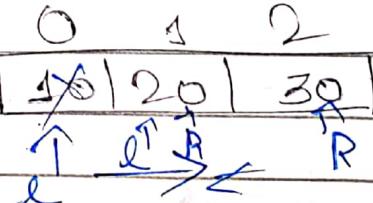
APEC
Date: 12/7



⑥ \rightarrow Extreme point in an Array \rightarrow



Odd Case \rightarrow



$l \leq R \rightarrow 10, 30, 20, 20$

\rightarrow Add condn if ($l == R$) \rightarrow cout << arr[l] | arr[R] \rightarrow only

Q/A/2023

APCO
Date: 123

H/w → Max in an Array

- ① → Swap → i) +, -
ii) XOR
iii) temp var

② → Max in an Array

Take max Ans = INT MAX

$n \rightarrow \text{size}$

for ($i=0$; $i < n$; $i++$)

if ($\text{arr}[i] > \text{maxAns}$)] → OR $\text{max}(\text{maxAns},$

$\text{maxAns} = \text{arr}[i]$

return maxAns

\downarrow
Subproblem $\text{arr}[i]$
function

③ → Swap

i) +, -

q/p let $a = 10$, $b = 5$

o/p $\rightarrow b = 10$, $a = 5$

first $\rightarrow a = a + b$;

Then $\rightarrow b = a / b$;
 $a = a - b$;

~~Ex-2~~ $a = 3, b = 4$

$$a = a + b = 7 \rightarrow \text{Total}$$

$$= (7) (4)$$

$$b = a - b \rightarrow \text{Total} - b = a$$

$$b = a = 3$$

$$a = a - b \rightarrow \text{Total} - a = b$$

$$(7) (3)$$

$$\rightarrow a = 4$$

(ii) XOR

$$a = a \wedge b;$$

$$b = a \wedge b;$$

$$a = a \wedge b$$

~~Ex-2~~ $a = 1 \rightarrow 01$ $b = 2 \rightarrow 10$

$$a = a \wedge b \rightarrow \begin{array}{r} 01 \\ 1 \wedge 2 \\ \hline 11 \end{array} \Rightarrow a = 3 \rightarrow 11$$

$$b = a \wedge b \rightarrow 3 \wedge 2 \Rightarrow \begin{array}{r} 11 \\ 1 \wedge 0 \\ \hline 01 \end{array} \Rightarrow b = 1$$

$$a = a \wedge b \rightarrow 3 \wedge 1 \Rightarrow \begin{array}{r} 11 \\ 01 \\ \hline 10 \end{array} \Rightarrow a = 2$$

(iii) Temp

$$\text{int Temp} = a$$

$$a = b;$$

$$b = \text{Temp};$$