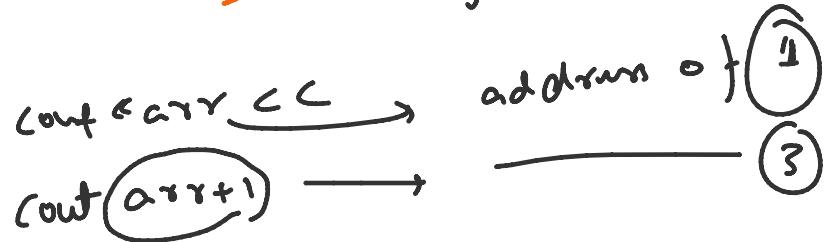


2 D arrays contd.

03 July 2023 19:21

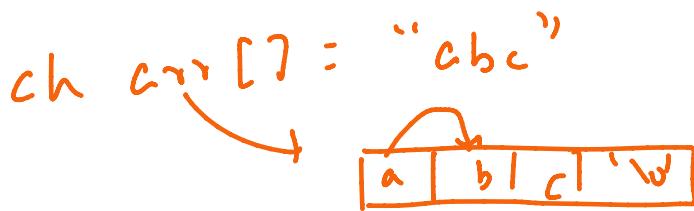
2D arrays:

$$\text{cout} \ll \text{arr}[1][2] = \boxed{\{1, 2, 3\}}_1 \quad \boxed{4}_2 ;$$



$$\text{int arr2[3] = \{1, 2, 3\}}$$

$\text{cout} \ll \text{arr2} + 1 \rightarrow \text{address}$ 0+2



$\text{cout} \ll \text{arr} \rightarrow abc$

$\text{cout} \ll \text{arr} + 1 \rightarrow bc$

$\text{cout} \ll \text{arr} + 2 \rightarrow c$

$$\text{char arr[4][2] = \{A, B, C, D\};}$$

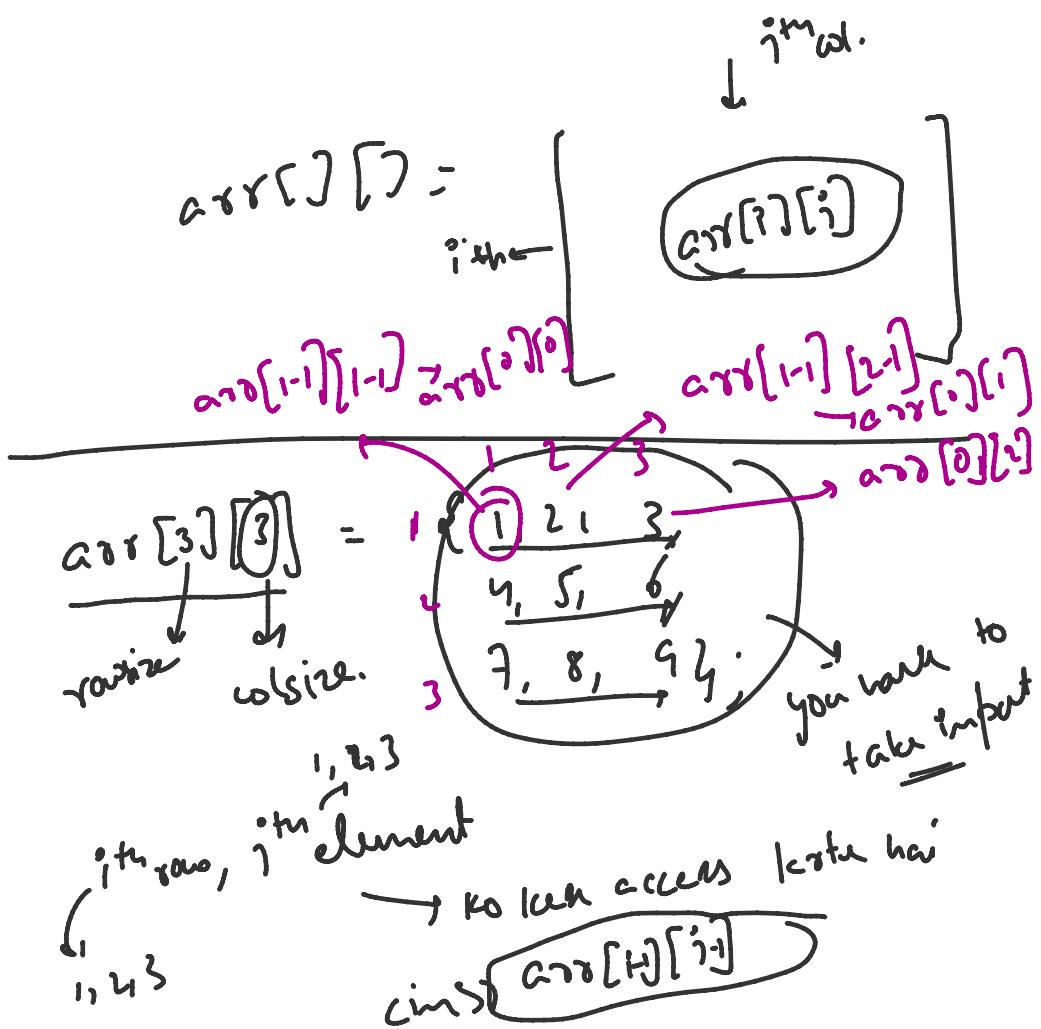
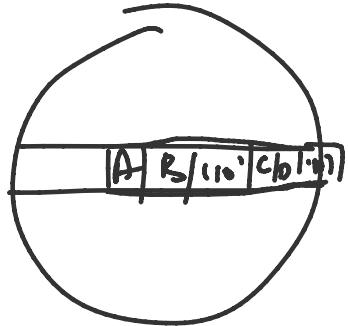
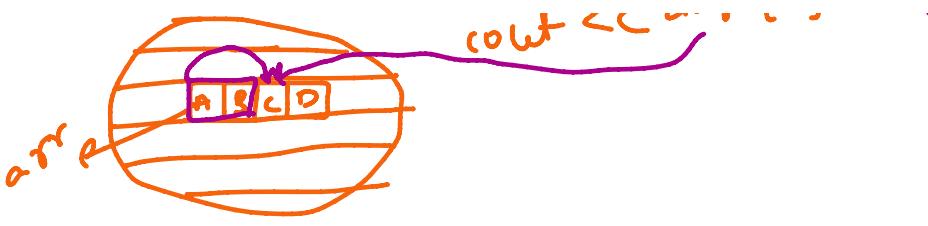
for char 2D arrays

$\text{cout} \ll \text{arr} \rightarrow ABCD \dots \{ ' \} \rightarrow$ miss with not wrapped

$\text{arr}[0] \rightarrow ARCD \dots \dots$

$\text{cout} \ll \text{arr}[1] \rightarrow CD \dots \{ ' \}$





```

for 1st row
for (int j=0; j < colsize; j++)
    cin >> arr[0][j];
}

for (int i=0; i < rowsize; i++) {
    for (int j=0; j < colsize; j++) {
        cin >> arr[i][j];
    }
}

```

for i in row
 ↓
 for j in col
 ↓
 cout < rowsize

```

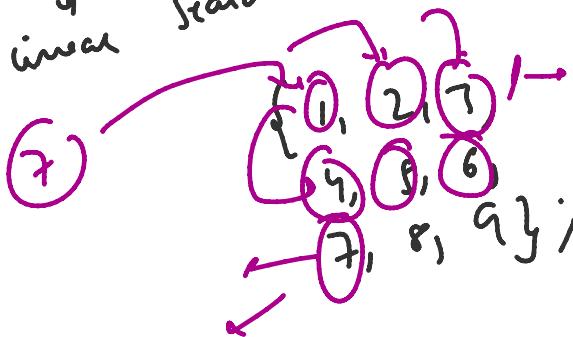
for (int i = 0; i < rowsize; i++)
  for (int j = 0; j < colsize; j++)
    cout < arr[i][j]
  
```

int eff

(7)

- (row >> eff)

↓ linear search



ans [] =

{ 1, 2, 3,
 4, 5, 6,
 7, 8, 9 };



Ans

[7, 4, 1, 2, 5, 8, 9, 6, 3]

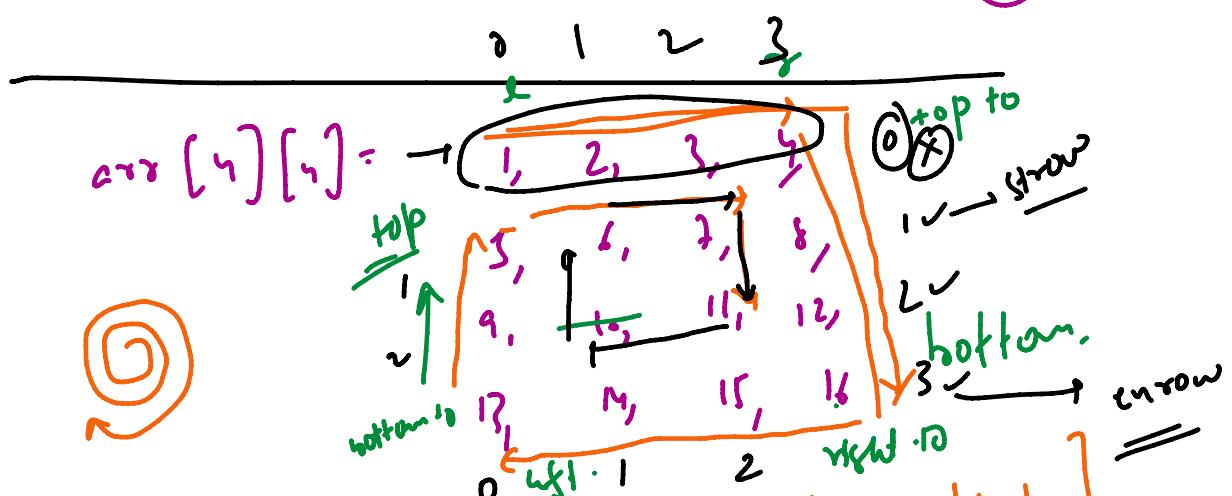
	0	1	2
0	1	2	3
1	4	8	6
2	7	5	9

I have to print according to column.
 ... the gap will be for columns.

→ have to print
outer loop will be for columns.
on which → **(bottom to top)** print elements.
1st col → Top to bottom
and → **(bottom to top)**

if my column no. is even → bottom to top
else → top to bottom.

for $i < 3$ w.r.t. for ($\text{int } i = 0; i < 3; i++$)
bottom to top. $\text{cout} \ll arr[i][i];$ → i .



$[0/p \rightarrow 1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6, 7, 11, 10]$

	0	1	2	3
0	1, 2, 3, 4,			
1	5, 6, 7, 8,			
2	9, 10, 11, 12,			
3	13, 14, 15, 16			

$$\begin{aligned} S_{LC} &= 1 \\ O_{LC} &= 3^2 \\ H_{RC} &= 4^2 \\ O_{RC} &= 3^2 \end{aligned}$$

print row
from col1 to col2

Now it's
from start to end

$$\%p = 1, 2, 3, 4, 8, 12, 16, 13, 14, 13, 9, 5, 6, 7, 11, 10$$

In 1D array

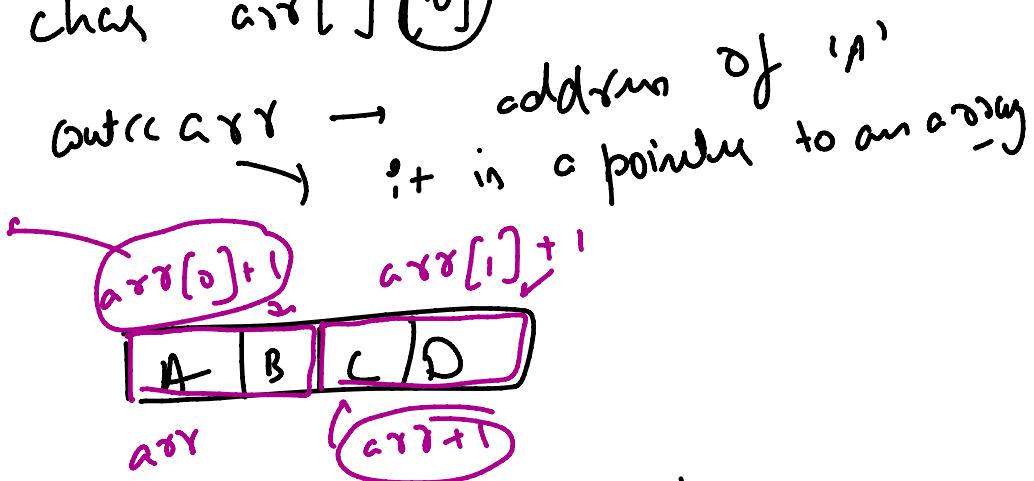


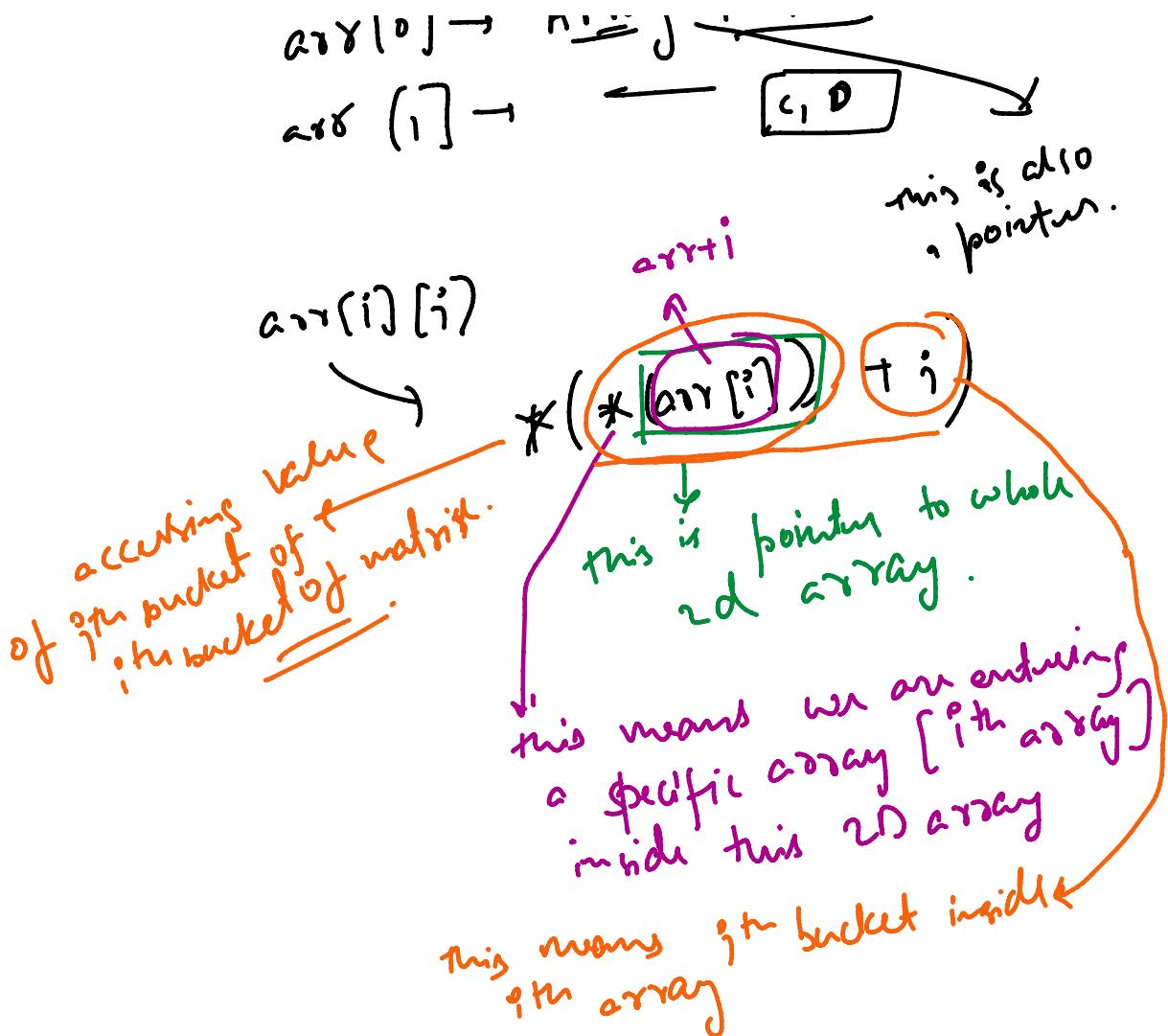
int arr [3][?]

→ this is a pointer to an
(bucket) ~~array~~ of size 3 (column wise)
↳ column pointer.

arr + 1 → 12 by the next address

char arr [] [?]





$\text{arr} \rightarrow$ $\text{arr} = \text{arr}[0]$
 pointer of arr
 (2D)

pointer to 1st array in arr

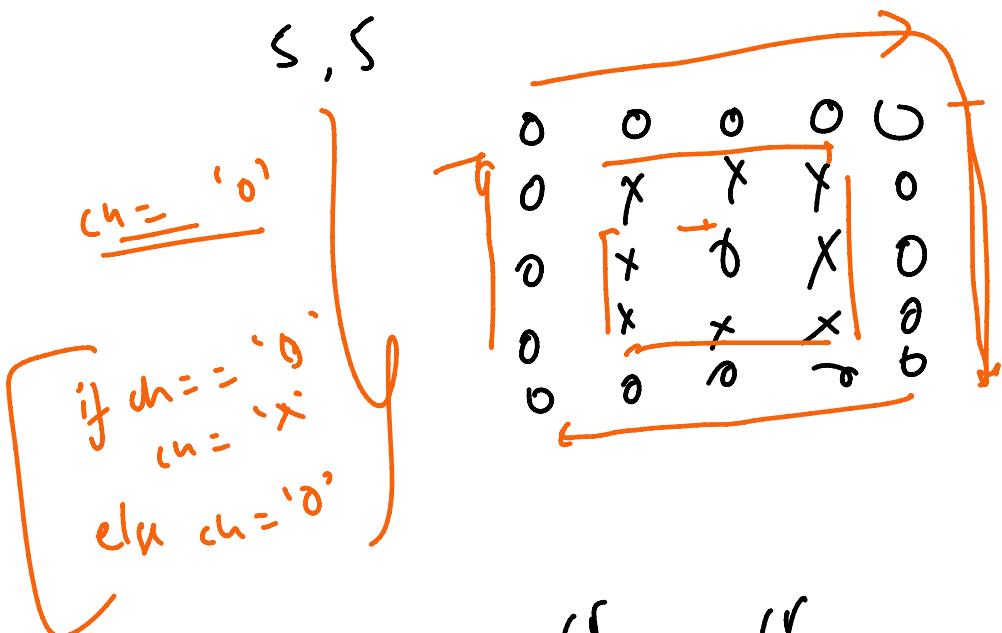
$\text{arr} + 1$
 pointer to
 $\text{arr} + 1$ st bucket
 i.e. 2nd array
 of arr.

$\text{arr}[i] + 1 \rightarrow$ pointer to
 2nd bucket in
 1st array

Spiral print

Spiral point

You will be given rows & columns
→ 0's & X's. like pattern



wut 'A' → 'B' → 0.

wut 'B' > 'A' → 1

$s_1 = \text{"hello"}$
 $s_2 = \text{"world"}$

Diagram showing two ovals labeled 105. Arrows point from each oval to the corresponding string "hello" and "world".

(1)

$s_1 = \text{cat}$
 $s_2 = \underline{\text{catc}}$

Diagram showing two ovals labeled 105. An arrow points from the first oval to the string "cat". Another oval contains the expression $s_1 < s_2 \Rightarrow$.