

29th | Sep Dec | Arrays
array is a collection of same data type. and has continuous memory.

public class tool {

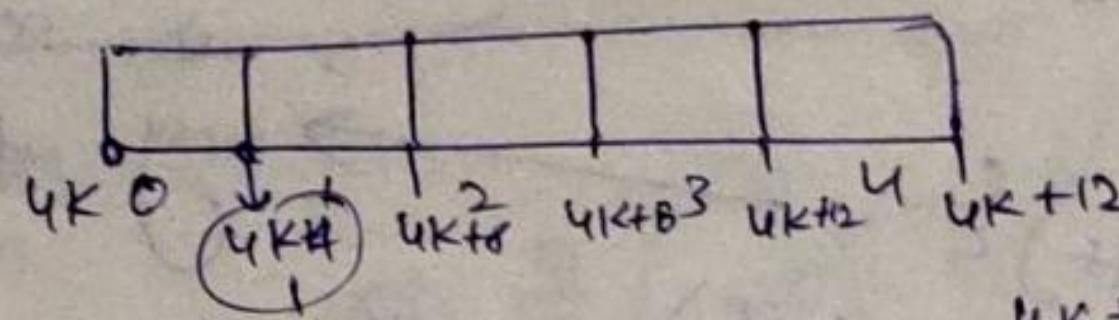
p.s. Scanner scr = new Scanner (System.in);

public static void test1() {

// how to declare array.

int[] arr = new int[n]; // n size

~~Array input uttalega~~



4K+4 → because 1 integer = 4 byte

why use array

→ why use arrays? ~~eg. when~~

Sol = agar hume 100 data store karna hai toh traditional way is 100 variable create kar sake h. 100 data kaise be liye 100 variable create karega. out a, int b, ... or hume same variable badiye

Karake kaise. So array give concept,

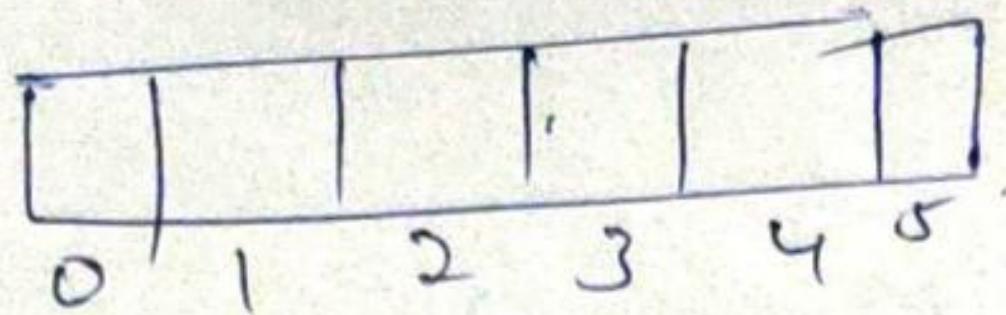
ki memory ki tension na ho, i.e. continuous memory bado

∴ Jiske bde size ke data chahiye utne bde size ka chunk of memory chalo or unko divide kardo 4 byte me.

so har box me data hota h.

it starts from 0 to n-1
↳ Always less than its length.

How to access the arr. Array?



① $\text{arr}[3] =$ means 3rd index to access 4th element

② $\text{arr}[6]; \rightarrow$ Not exist it gives null pointer exception,
Index Out of Bound

by default value 0:

```
int arr = new int[n]; // Declaring n size array -  
// Cpp runtime memory decide bhi hoga i.e user dependent mem.]
```

for (int i=0; i<n; i++) {
 arr[i] = scn. nextInt();
}

for (int i=0; i<n; i++) {
 cout < arr[i] < endl;

}

```
ps v m { stay(args) {  
    fexit();  
}
```

3.

Q: In CPP, how to declare array
A: vector

vector <int> arr(n, 10)
n size
Default value is 10.

Why not use pointer?

age by chance 2D, 3D Brane ki
need path or use return known
ki need path to kaise bhega jo
pointer to pointer → very
so we use vector

CPP: me but display your likege

void display (vector<int> & arr) {
 arr[0] = 10; arr[1] = 20; arr[2] = 30; arr[3] = 40; arr[4] = 50; arr[5] = 60;

for (int i=0; i < arr.size(); i++)

cout << arr[i] << " ";

}

Java Display function

P.S. void

display (int[] arr) {

for (int i=0; i < arr.length; i++) {

System.out.println(arr[i]);

}

get size of array
print each element
array has 6 elements
range is 0 to 5

Java me [length];
use hole in our
call no [size()];

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

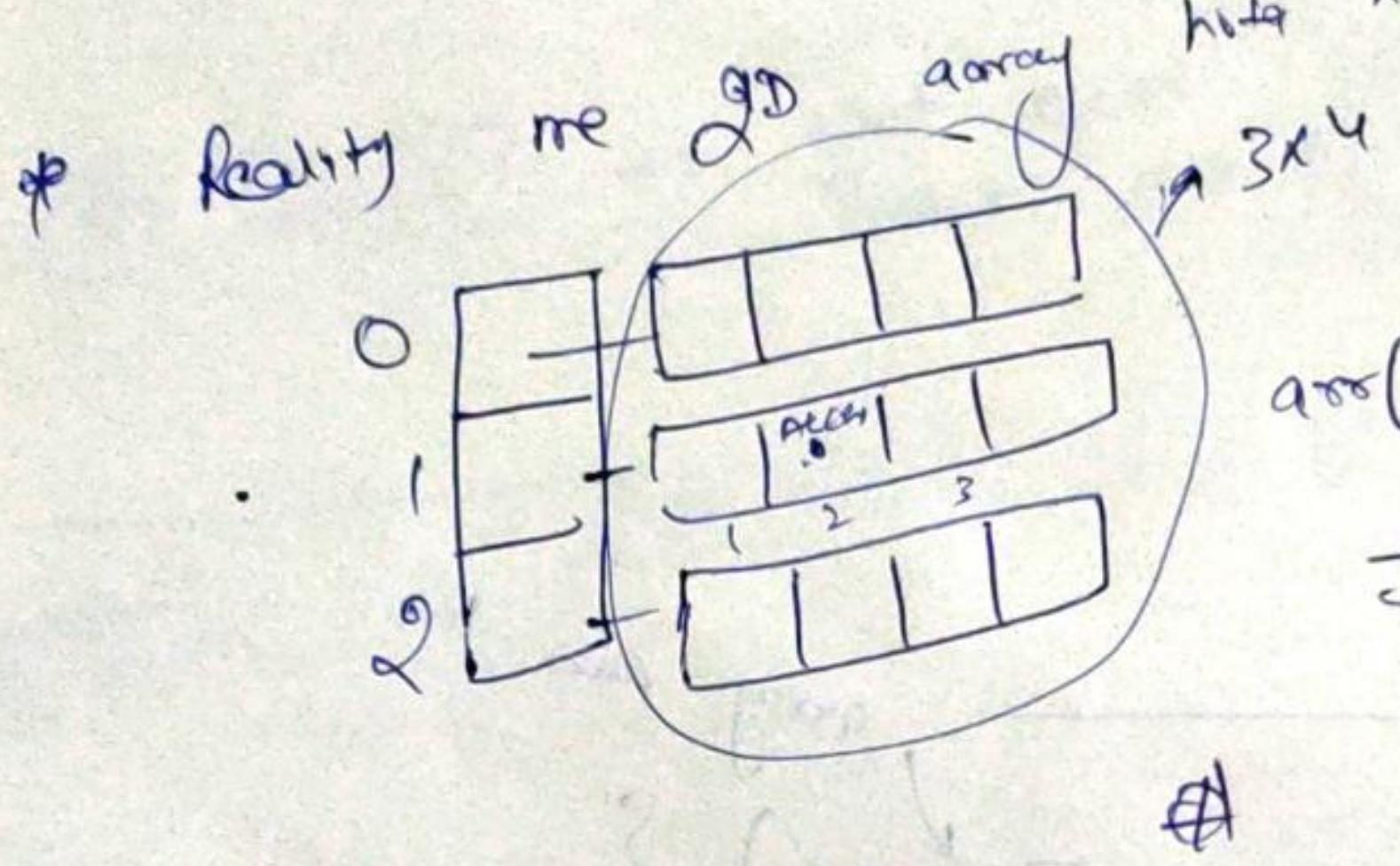
arr[i]

arr[0] → 10 | 40 | 70 |
arr[1] → 20 | 50 | 80 |
arr[2] → 30 |

arr[3] → 10 | 20 | 30 | 40 |
arr[4] → 50 | 60 | 70 |
arr[5] → 80 | 90 | 100 |
arr[6] → 110 | 120 | 130 |
arr[7] → 140 | 150 | 160 |
arr[8] → 170 | 180 | 190 |
arr[9] → 200 | 210 | 220 |
arr[10] → 230 | 240 | 250 |
arr[11] → 260 | 270 | 280 |
arr[12] → 290 | 300 | 310 |
arr[13] → 320 | 330 | 340 |
arr[14] → 350 | 360 | 370 |
arr[15] → 380 | 390 | 400 |
arr[16] → 410 | 420 | 430 |
arr[17] → 440 | 450 | 460 |
arr[18] → 470 | 480 | 490 |
arr[19] → 500 | 510 | 520 |
arr[20] → 530 | 540 | 550 |
arr[21] → 560 | 570 | 580 |
arr[22] → 590 | 600 | 610 |
arr[23] → 620 | 630 | 640 |
arr[24] → 650 | 660 | 670 |
arr[25] → 680 | 690 | 700 |
arr[26] → 710 | 720 | 730 |
arr[27] → 740 | 750 | 760 |
arr[28] → 770 | 780 | 790 |
arr[29] → 800 | 810 | 820 |
arr[30] → 830 | 840 | 850 |
arr[31] → 860 | 870 | 880 |
arr[32] → 890 | 900 | 910 |
arr[33] → 920 | 930 | 940 |
arr[34] → 950 | 960 | 970 |
arr[35] → 980 | 990 | 1000 |
arr[36] → 1010 | 1020 | 1030 |
arr[37] → 1040 | 1050 | 1060 |
arr[38] → 1070 | 1080 | 1090 |
arr[39] → 1100 | 1110 | 1120 |
arr[40] → 1130 | 1140 | 1150 |
arr[41] → 1160 | 1170 | 1180 |
arr[42] → 1190 | 1200 | 1210 |
arr[43] → 1220 | 1230 | 1240 |
arr[44] → 1250 | 1260 | 1270 |
arr[45] → 1280 | 1290 | 1300 |
arr[46] → 1310 | 1320 | 1330 |
arr[47] → 1340 | 1350 | 1360 |
arr[48] → 1370 | 1380 | 1390 |
arr[49] → 1400 | 1410 | 1420 |
arr[50] → 1430 | 1440 | 1450 |
arr[51] → 1460 | 1470 | 1480 |
arr[52] → 1490 | 1500 | 1510 |
arr[53] → 1520 | 1530 | 1540 |
arr[54] → 1550 | 1560 | 1570 |
arr[55] → 1580 | 1590 | 1600 |
arr[56] → 1610 | 1620 | 1630 |
arr[57] → 1640 | 1650 | 1660 |
arr[58] → 1670 | 1680 | 1690 |
arr[59] → 1700 | 1710 | 1720 |
arr[60] → 1730 | 1740 | 1750 |
arr[61] → 1760 | 1770 | 1780 |
arr[62] → 1790 | 1800 | 1810 |
arr[63] → 1820 | 1830 | 1840 |
arr[64] → 1850 | 1860 | 1870 |
arr[65] → 1880 | 1890 | 1900 |
arr[66] → 1910 | 1920 | 1930 |
arr[67] → 1940 | 1950 | 1960 |
arr[68] → 1970 | 1980 | 1990 |
arr[69] → 2000 | 2010 | 2020 |
arr[70] → 2030 | 2040 | 2050 |
arr[71] → 2060 | 2070 | 2080 |
arr[72] → 2090 | 2100 | 2110 |
arr[73] → 2120 | 2130 | 2140 |
arr[74] → 2150 | 2160 | 2170 |
arr[75] → 2180 | 2190 | 2190 |
arr[76] → 2200 | 2210 | 2220 |
arr[77] → 2230 | 2240 | 2250 |
arr[78] → 2260 | 2270 | 2280 |
arr[79] → 2290 | 2300 | 2310 |
arr[80] → 2320 | 2330 | 2340 |
arr[81] → 2350 | 2360 | 2370 |
arr[82] → 2380 | 2390 | 2390 |
arr[83] → 2400 | 2410 | 2420 |
arr[84] → 2430 | 2440 | 2450 |
arr[85] → 2460 | 2470 | 2480 |
arr[86] → 2490 | 2500 | 2510 |
arr[87] → 2520 | 2530 | 2540 |
arr[88] → 2550 | 2560 | 2570 |
arr[89] → 2580 | 2590 | 2590 |
arr[90] → 2600 | 2610 | 2620 |
arr[91] → 2630 | 2640 | 2650 |
arr[92] → 2660 | 2670 | 2680 |
arr[93] → 2690 | 2700 | 2710 |
arr[94] → 2720 | 2730 | 2740 |
arr[95] → 2750 | 2760 | 2770 |
arr[96] → 2780 | 2790 | 2790 |
arr[97] → 2800 | 2810 | 2820 |
arr[98] → 2830 | 2840 | 2850 |
arr[99] → 2860 | 2870 | 2880 |
arr[100] → 2890 | 2900 | 2910 |
arr[101] → 2920 | 2930 | 2940 |
arr[102] → 2950 | 2960 | 2970 |
arr[103] → 2980 | 2990 | 2990 |
arr[104] → 3000 | 3010 | 3020 |
arr[105] → 3030 | 3040 | 3050 |
arr[106] → 3060 | 3070 | 3080 |
arr[107] → 3090 | 3100 | 3110 |
arr[108] → 3120 | 3130 | 3140 |
arr[109] → 3150 | 3160 | 3170 |
arr[110] → 3180 | 3190 | 3190 |
arr[111] → 3200 | 3210 | 3220 |
arr[112] → 3230 | 3240 | 3250 |
arr[113] → 3260 | 3270 | 3280 |
arr[114] → 3290 | 3300 | 3310 |
arr[115] → 3320 | 3330 | 3340 |
arr[116] → 3350 | 3360 | 3370 |
arr[117] → 3380 | 3390 | 3390 |
arr[118] → 3400 | 3410 | 3420 |
arr[119] → 3430 | 3440 | 3450 |
arr[120] → 3460 | 3470 | 3480 |
arr[121] → 3490 | 3500 | 3510 |
arr[122] → 3520 | 3530 | 3540 |
arr[123] → 3550 | 3560 | 3570 |
arr[124] → 3580 | 3590 | 3590 |
arr[125] → 3600 | 3610 | 3620 |
arr[126] → 3630 | 3640 | 3650 |
arr[127] → 3660 | 3670 | 3680 |
arr[128] → 3690 | 3700 | 3710 |
arr[129] → 3720 | 3730 | 3740 |
arr[130] → 3750 | 3760 | 3770 |
arr[131] → 3780 | 3790 | 3790 |
arr[132] → 3800 | 3810 | 3820 |
arr[133] → 3830 | 3840 | 3850 |
arr[134] → 3860 | 3870 | 3880 |
arr[135] → 3890 | 3900 | 3910 |
arr[136] → 3920 | 3930 | 3940 |
arr[137] → 3950 | 3960 | 3970 |
arr[138] → 3980 | 3990 | 3990 |
arr[139] → 4000 | 4010 | 4020 |
arr[140] → 4030 | 4040 | 4050 |
arr[141] → 4060 | 4070 | 4080 |
arr[142] → 4090 | 4100 | 4110 |
arr[143] → 4120 | 4130 | 4140 |
arr[144] → 4150 | 4160 | 4170 |
arr[145] → 4180 | 4190 | 4190 |
arr[146] → 4200 | 4210 | 4220 |
arr[147] → 4230 | 4240 | 4250 |
arr[148] → 4260 | 4270 | 4280 |
arr[149] → 4290 | 4300 | 4310 |
arr[150] → 4320 | 4330 | 4340 |
arr[151] → 4350 | 4360 | 4370 |
arr[152] → 4380 | 4390 | 4390 |
arr[153] → 4400 | 4410 | 4420 |
arr[154] → 4430 | 4440 | 4450 |
arr[155] → 4460 | 4470 | 4480 |
arr[156] → 4490 | 4500 | 4510 |
arr[157] → 4520 | 4530 | 4540 |
arr[158] → 4550 | 4560 | 4570 |
arr[159] → 4580 | 4590 | 4590 |
arr[160] → 4600 | 4610 | 4620 |
arr[161] → 4630 | 4640 | 4650 |
arr[162] → 4660 | 4670 | 4680 |
arr[163] → 4690 | 4700 | 4710 |
arr[164] → 4720 | 4730 | 4740 |
arr[165] → 4750 | 4760 | 4770 |
arr[166] → 4780 | 4790 | 4790 |
arr[167] → 4800 | 4810 | 4820 |
arr[168] → 4830 | 4840 | 4850 |
arr[169] → 4860 | 4870 | 4880 |
arr[170] → 4890 | 4900 | 4910 |
arr[171] → 4920 | 4930 | 4940 |
arr[172] → 4950 | 4960 | 4970 |
arr[173] → 4980 | 4990 | 4990 |
arr[174] → 5000 | 5010 | 5020 |
arr[175] → 5030 | 5040 | 5050 |
arr[176] → 5060 | 5070 | 5080 |
arr[177] → 5090 | 5100 | 5110 |
arr[178] → 5120 | 5130 | 5140 |
arr[179] → 5150 | 5160 | 5170 |
arr[180] → 5180 | 5190 | 5190 |
arr[181] → 5200 | 5210 | 5220 |
arr[182] → 5230 | 5240 | 5250 |
arr[183] → 5260 | 5270 | 5280 |
arr[184] → 5290 | 5300 | 5310 |
arr[185] → 5320 | 5330 | 5340 |
arr[186] → 5350 | 5360 | 5370 |
arr[187] → 5380 | 5390 | 5390 |
arr[188] → 5400 | 5410 | 5420 |
arr[189] → 5430 | 5440 | 5450 |
arr[190] → 5460 | 5470 | 5480 |
arr[191] → 5490 | 5500 | 5510 |
arr[192] → 5520 | 5530 | 5540 |
arr[193] → 5550 | 5560 | 5570 |
arr[194] → 5580 | 5590 | 5590 |
arr[195] → 5600 | 5610 | 5620 |
arr[196] → 5630 | 5640 | 5650 |
arr[197] → 5660 | 5670 | 5680 |
arr[198] → 5690 | 5700 | 5710 |
arr[199] → 5720 | 5730 | 5740 |
arr[200] → 5750 | 5760 | 5770 |
arr[201] → 5780 | 5790 | 5790 |
arr[202] → 5800 | 5810 | 5820 |
arr[203] → 5830 | 5840 | 5850 |
arr[204] → 5860 | 5870 | 5880 |
arr[205] → 5890 | 5900 | 5910 |
arr[206] → 5920 | 5930 | 5940 |
arr[207] → 5950 | 5960 | 5970 |
arr[208] → 5980 | 5990 | 5990 |
arr[209] → 6000 | 6010 | 6020 |
arr[210] → 6030 | 6040 | 6050 |
arr[211] → 6060 | 6070 | 6080 |
arr[212] → 6090 | 6100 | 6110 |
arr[213] → 6120 | 6130 | 6140 |
arr[214] → 6150 | 6160 | 6170 |
arr[215] → 6180 | 6190 | 6190 |
arr[216] → 6200 | 6210 | 6220 |
arr[217] → 6230 | 6240 | 6250 |
arr[218] → 6260 | 6270 | 6280 |
arr[219] → 6290 | 6300 | 6310 |
arr[220] → 6320 | 6330 | 6340 |
arr[221] → 6350 | 6360 | 6370 |
arr[222] → 6380 | 6390 | 6390 |
arr[223] → 6400 | 6410 | 6420 |
arr[224] → 6430 | 6440 | 6450 |
arr[225] → 6460 | 6470 | 6480 |
arr[226] → 6490 | 6500 | 6510 |
arr[227] → 6520 | 6530 | 6540 |
arr[228] → 6550 | 6560 | 6570 |
arr[229] → 6580 | 6590 | 6590 |
arr[230] → 6600 | 6610 | 6620 |
arr[231] → 6630 | 6640 | 6650 |
arr[232] → 6660 | 6670 | 6680 |
arr[233] → 6690 | 6700 | 6710 |
arr[234] → 6720 | 6730 | 6740 |
arr[235] → 6750 | 6760 | 6770 |
arr[236] → 6780 | 6790 | 6790 |
arr[237] → 6800 | 6810 | 6820 |
arr[238] → 6830 | 6840 | 6850 |
arr[239] → 6860 | 6870 | 6880 |
arr[240] → 6890 | 6900 | 6910 |
arr[241] → 6920 | 6930 | 6940 |
arr[242] → 6950 | 6960 | 6970 |
arr[243] → 6980 | 6990 | 6990 |
arr[244] → 7000 | 7010 | 7020 |
arr[245] → 7030 | 7040 | 7050 |
arr[246] → 7060 | 7070 | 7080 |
arr[247] → 7090 | 7100 | 7110 |
arr[248] → 7120 | 7130 | 7140 |
arr[249] → 7150 | 7160 | 7170 |
arr[250] → 7180 | 7190 | 7190 |
arr[251] → 7200 | 7210 | 7220 |
arr[25

How to declare 3D array
 → Java → $\text{int}[\text{ }][\text{ }][\text{ }] = \text{new int}[\text{ }][\text{ }][\text{ }]$
 $\rightarrow \text{CPP} \quad \text{v} \leftarrow \text{v} < \text{v} < \text{int} >> \text{arr}(\text{n}, \text{v} << \text{m}) < \text{m}, \text{v} << \text{p}, \text{o} >$
 v → type of vector

* Reality me 2D array
 Only 1D exists.



arr[1][2]

Java → Arrays Fill(-1).

ye pure Array me
 fill karte

Q1 = public static maximum(int[] arr){
 Q2 = minimum(int[] arr)
 Q3 = findData(int[] arr, int data)

last question

Q(1) Find max

Note urko minEle wt karte rkh & check or gate ki Array first thi karta h
For loop if style

Sol P.S. void

maximum (int arr)

→ // 1st element use aur m
Store bisko or aage
Check karte Jao bda h
koi yeh nhi

int maxEle = -(int)1e9;

↓

-10⁹ →

This loop Only
get the values
and always move kisi
forward

for loop → for (int ele : arr) {

h + 1
ke
step
2x

if (ele > maxEle) {

maxEle = ele;

maxEle bo
next kisi

}

}

why because mostly DP me
log use

INTEGER_MAX() → maximum
value.

use jere hi + 1 kaise rkh

Range se bhar value aage aur
-ve form me.

isliye max NIKALNA ho rkh
-(int)1e9.

② Day run

minEle = -10⁹

for (int ele : arr) {

if (ele < minEle) minEle = ele; }.

100	20	30	40	50	80	100	100	30
P	S	I	T	I	I	I	I	I

→ 10⁹

10⁹ se 100 chota h
yes

minEle = 10⁹ 100 40.
-10

some
foreach me
Index ko access
kone ki need nahi
hott direct no.
galt h

Q2

MPrismem → (Same as
other)

Q3

FindData

p. S. Bodhan

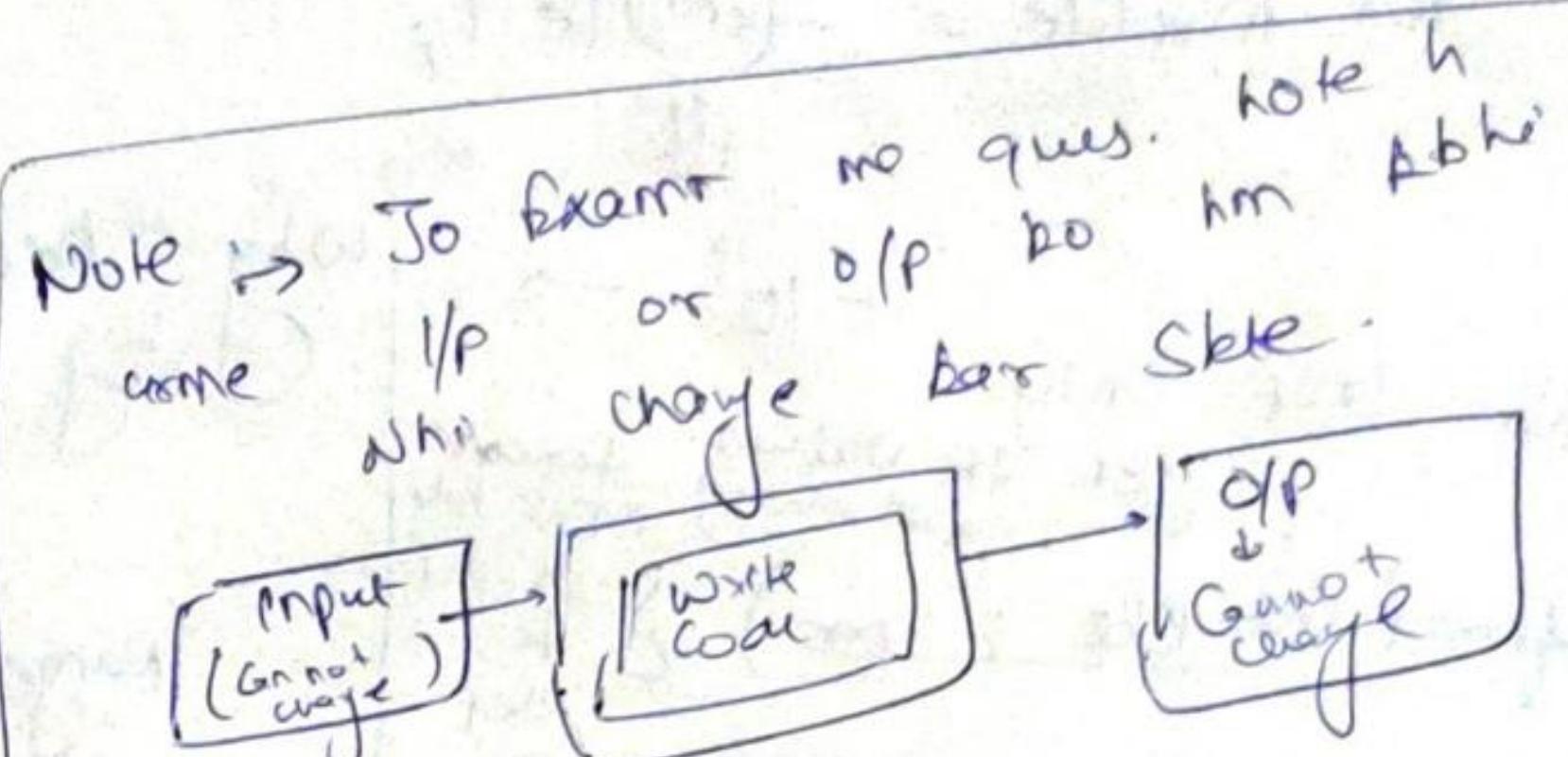
findData (int arr, int data) {

means for loop baage or

dekhoge
element
otherwise

agr

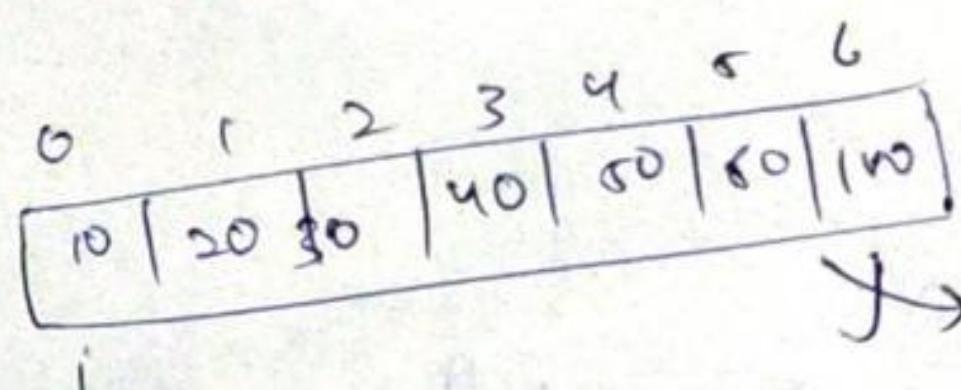
mil gaya toh true
Rider



{ agr position change ho
toh normal for loop use
karte h }

Q4
=
=

Reverse of an array



→

Reverse kare ke liye
& pointer phlego
i or j or
swap j exchange
++i, j--

swap

int temp = arr[i]

arr[i] = arr[j]

arr[j] = temp

temp = arr[i]

100 90 80 70 60 50

Temp

perle 0, 1, 2, 3, 4, 5, 6
Jese hi complete ho jega
and

→ swap(arr, i++, e--)

forloop (arr, 0, 4) → perle yeh
line chlegi

i = 0
e = 4
2) Inc.
order
ho jega

Q. Rotate An array (without extra space)

(a) store per
toggie ja

Carer
chotti

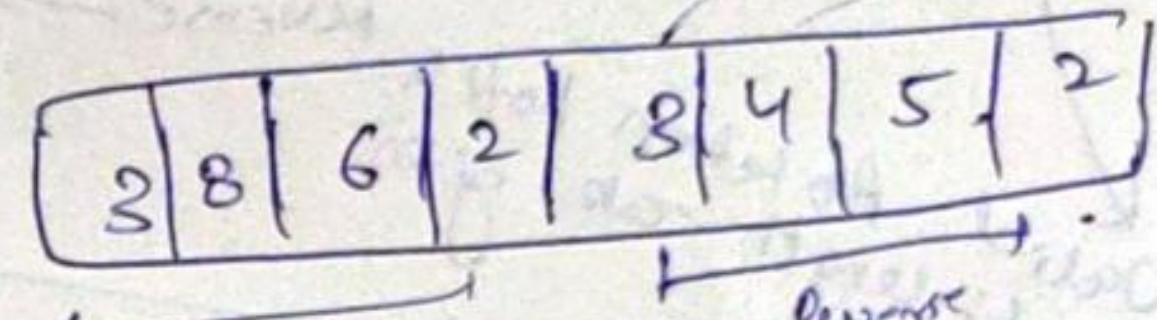
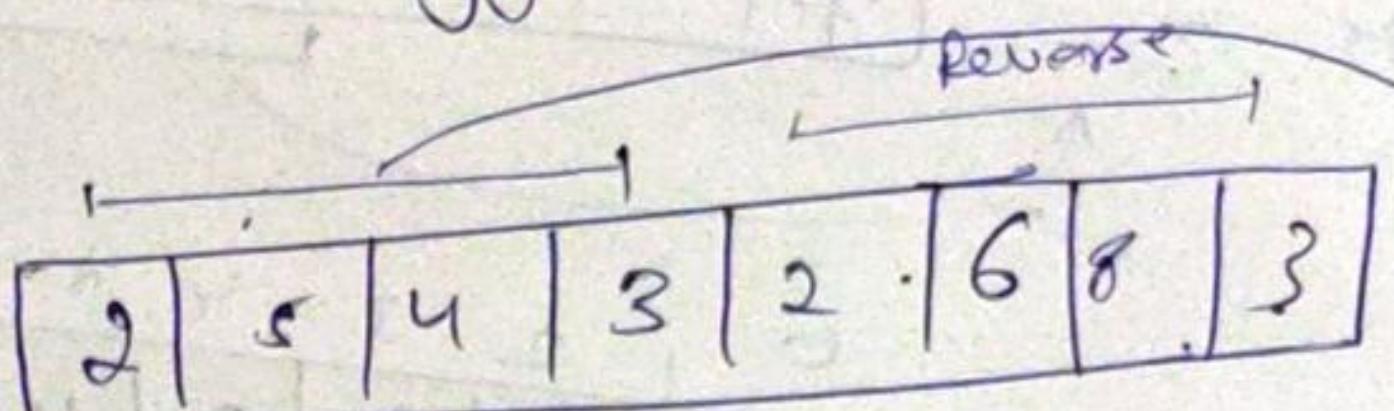
handle

space

Krige k value bali

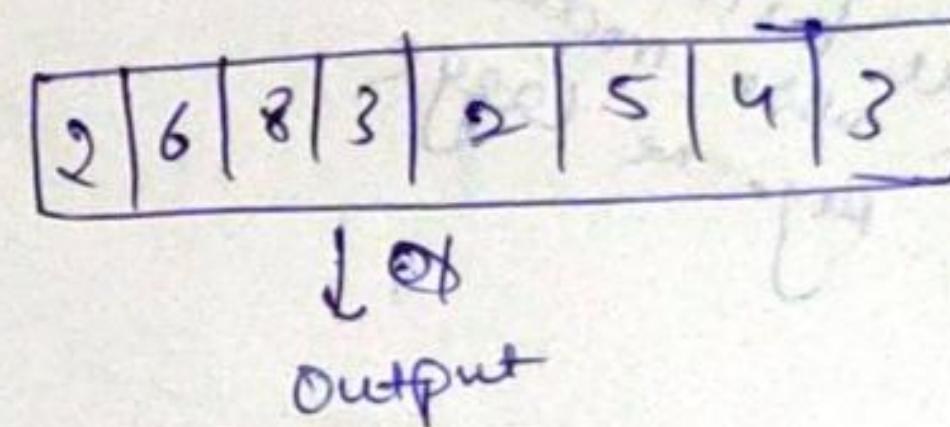
+4 rotate

1st 4 last
me agarne
chuye



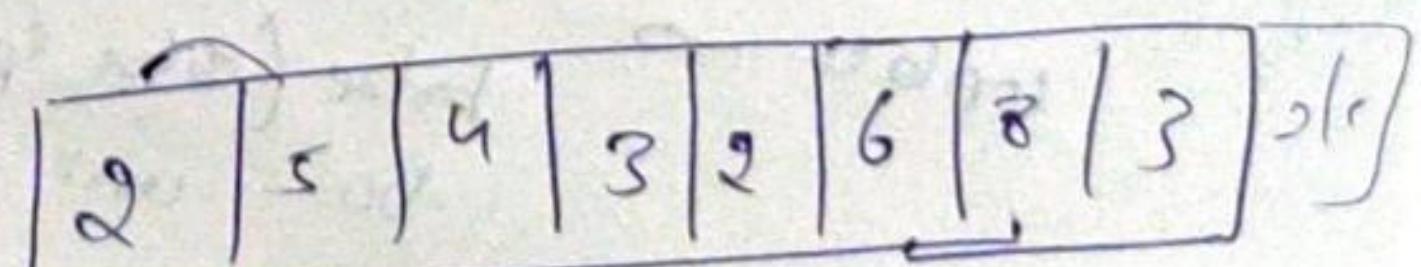
To K given h

Take $(0, K-1)$ take reverse kriye
 $(K, n-1)$ " "



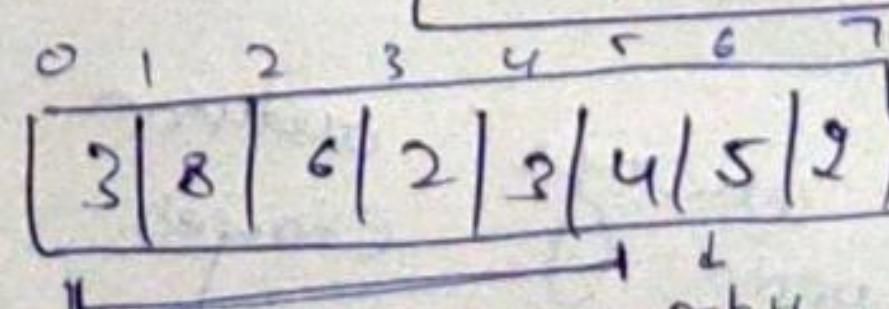
CPP →

```
void rotate (No char <int> & arr, int k) {
    int n = arr.size();
    k = (k % n + n) % n;
    reverse (arr, 0, n - 1); ① 1st pure array to
    reverse (arr, 0, k - 1);  reverse by
    reverse (arr, k, n - 1);
```



Full array
reverse

Reverse $(0, n-1)$ →



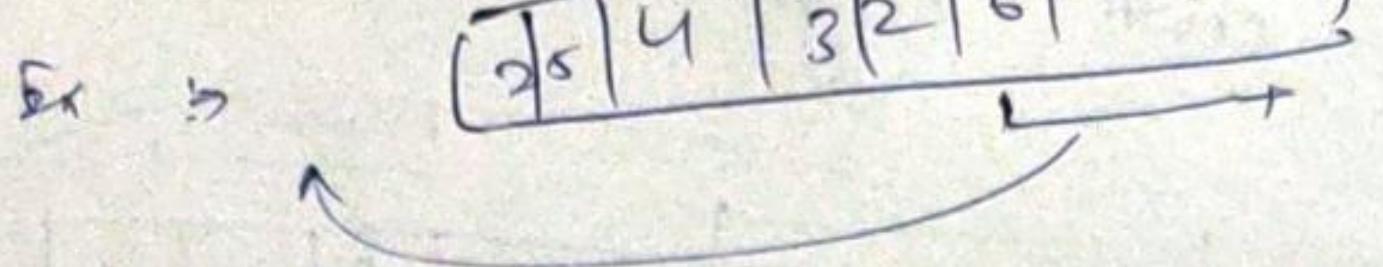
Reverse $(0, n-1)$

Reverse $(n-k+1, n)$

$n=8$
 $k=2$
 $8-2$

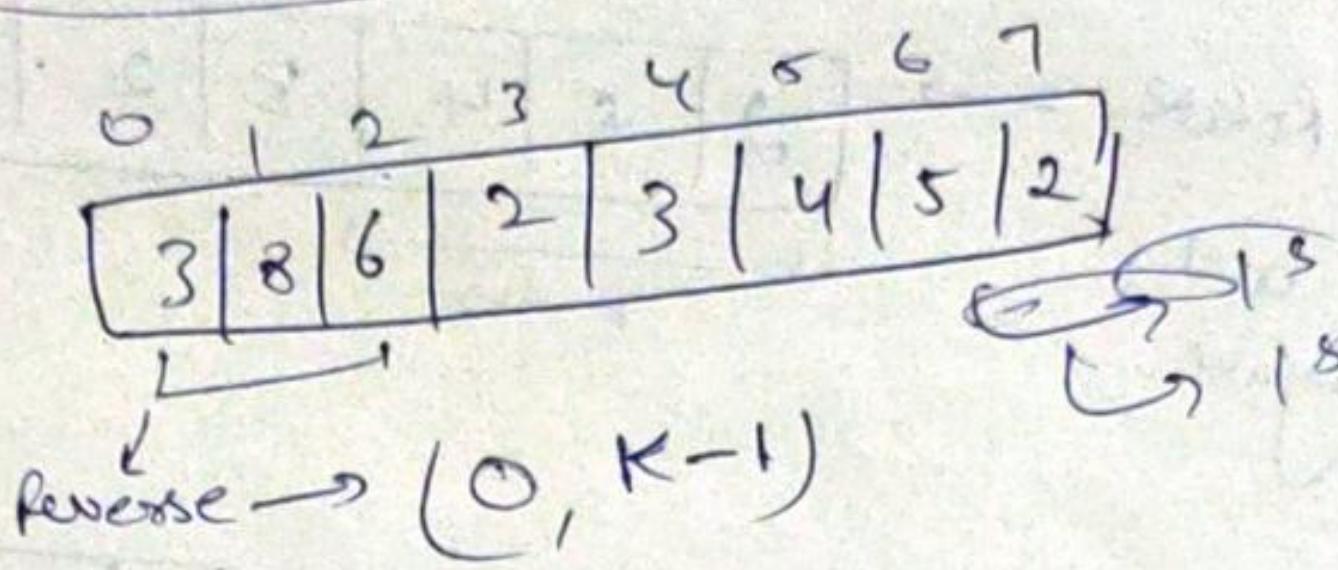
0-6

age Elements size len change in arr.

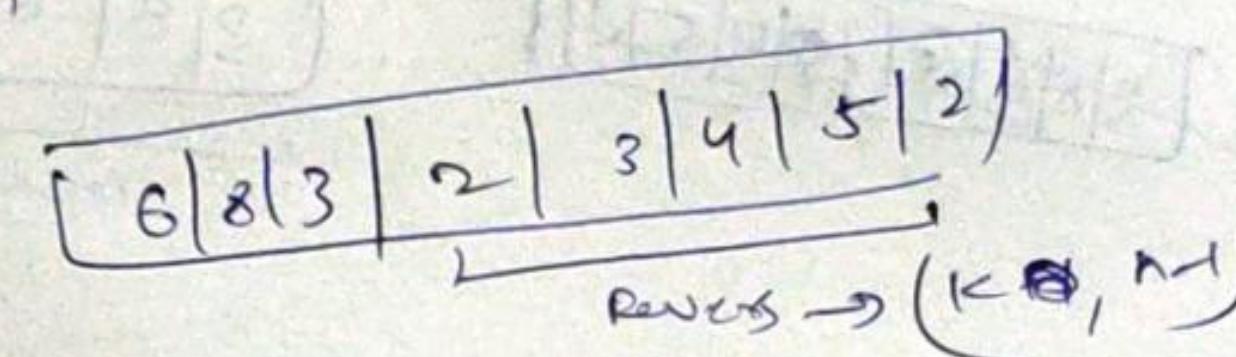


$$k = +3$$

(i) Reverse :-



$k \leftarrow$ Element reverse



$$k = (K \% n + m) \% m; \rightarrow \text{get -ve be true me shata n.}$$

$$-a \% m$$

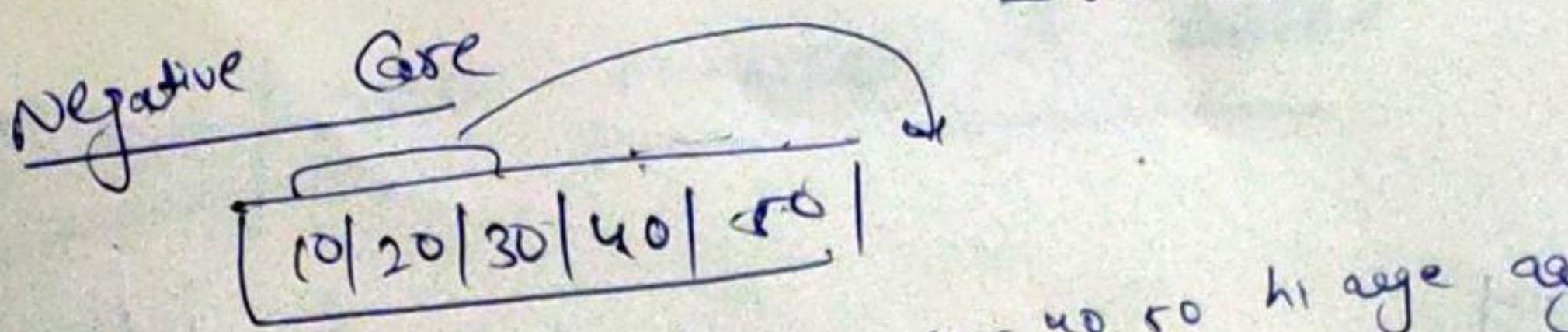
$$\Rightarrow (-a \% m + m) \% m$$

Imp :- Array or replace mean
Jo Array me change kore h
Vohi pure change hata h
print (out) nhi hata

(i) Array baki baki
Inplace → baki baki for extra space work
kore 'X'

Inplace → means to input data h ceri me work
kore (extra space rough hoga ya nhi). and Jo O/P hoga
vohi user me change hoga.

(ii) Array always return array or to input array hata h
vohi allowed nhi h



-ve to +ve.

$$K = -3$$

arr agr no, so hi arrge arrge
 $K = +2$

$(K < 0) \Sigma$
 $(K + = n)$

$$40, 50, 10, 20, 30$$

$$-3 + 5 = 2$$

rotate ↴

$$40, 50, 10, 20, 30$$

agr \rightarrow -ve me k ki bdi value
 note coh

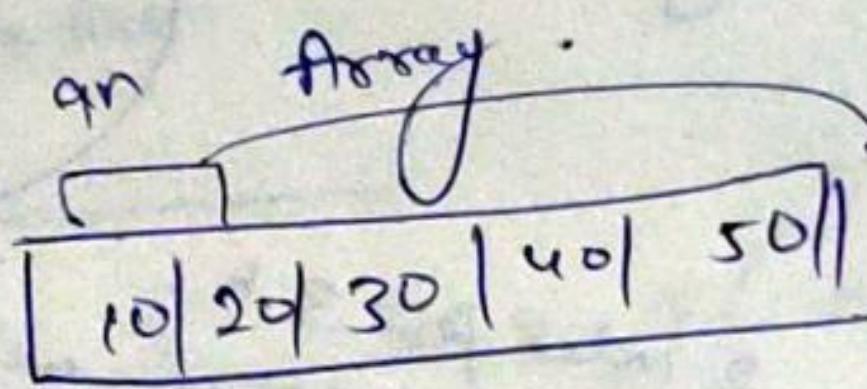
$$\text{eg} = K = -23\%$$

$$K = K \% n$$

$$= -23\% / 5$$

$\Rightarrow -1 \rightarrow$ isko karein
ta hi sketch

Repeat
Reverse of arr



$$K = 2 \quad i \quad 1 \quad 2 \quad 3 \quad 4$$

arr \rightarrow [30|40|50|10|20]

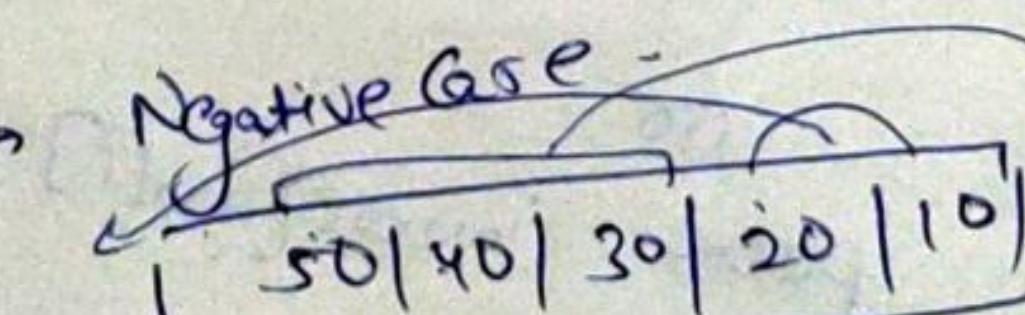
$$(i) \text{ Reverse} \rightarrow \begin{matrix} 0 & 1 & 2 & 3 & 4 \\ 50 & 40 & 30 & 20 & 10 \end{matrix}$$

reverse ↴ reverse ↴

reverse(0, K-1) reverse(K, n-1)

$$K = 2$$

② Case B \rightarrow



$$K = -3$$

$$20, 10, 50, 40, 30$$

$$K = m - m \quad -5 + 5 = 0$$

$$K = 2 \rightarrow R + S^n$$

$$20, 10, 50, 40, 30$$

$$K + = n$$

$$-3 + 3 = 2$$

Q. Inverse of an Array

1	4	3	2	0
0	1	2	3	4

→ 30 bhi values and they are distinct.

Q. agr original array A par Q tha.
So Inv. me 1 par 0 hoga

② original array me 1 index ke upr 4 tha
so Inv. array me 4 par 1 index
and similarly

③ 0.A me 0 ke upr 0 tha, Inv.
me 0 ke upr 0 h.

Inverse array

Ready bina h
oo vo bhi size ke hi bnege
utne

10	0	3	2	1
0	1	2	3	4

Sol) store pehle kisi value pta lgte h.
Jo arr[0] m store h.

0	1	2	3	4
1	4	3	2	0

O.A. → loop for (int i=0; i < arr.length-1; i++) {

old position ← int pos = arr[i];
inverse[pos] = i; → new position for
butor s[i] related

3.
first pos = arr[1]
inv[pos] = 0

pos → inverse[arr[i]] = 0;

original array →	1 4 3 2 0
inverse	0 1 2 3 4

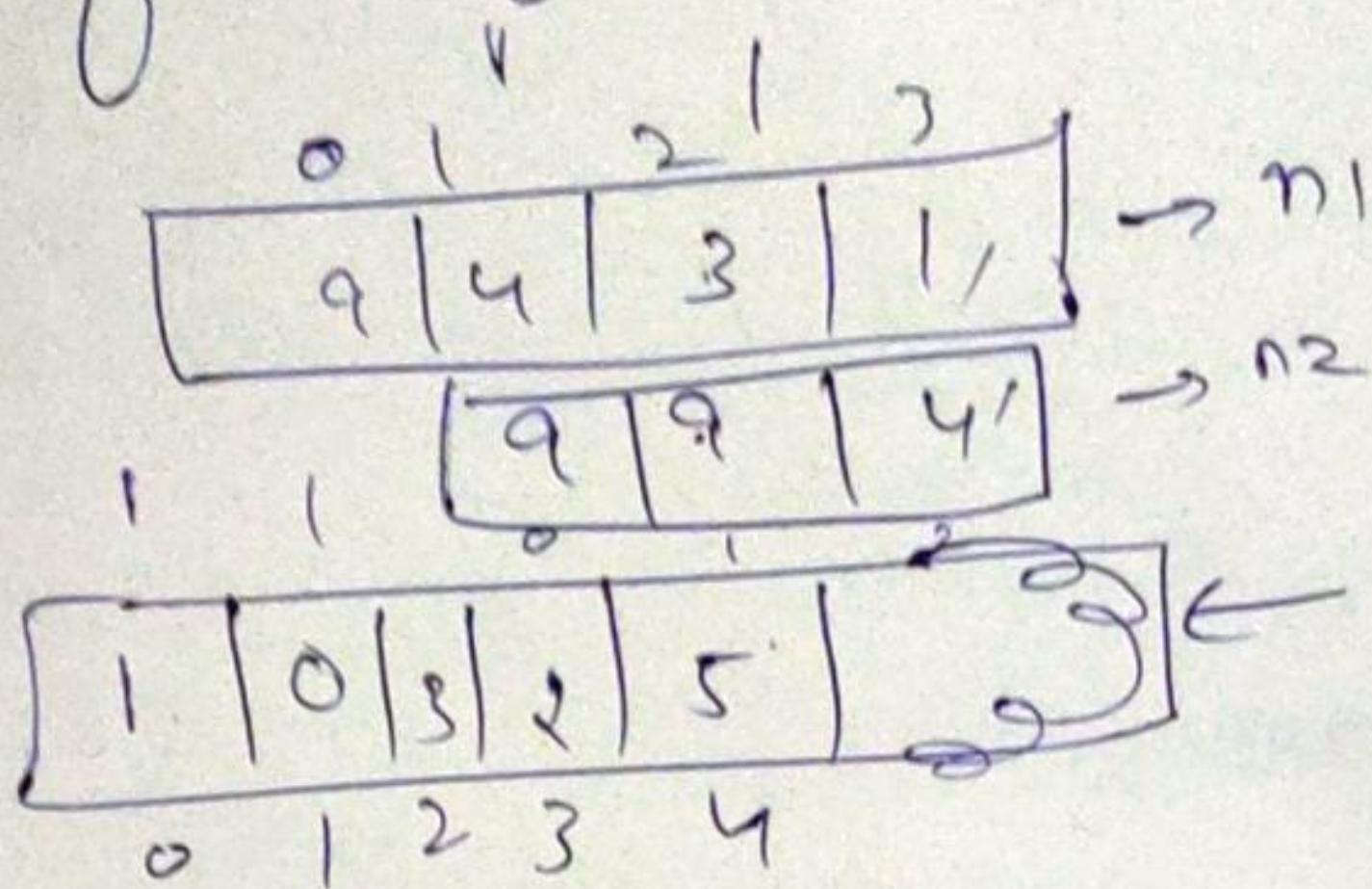
0.	1.	2.	3.	4
0.	1.	2.	3.	4

30th Dec / 2020

FUNCTION & ARRAY

Q1) sum of array.

Ex :-



new array
Math. max(n1, n2) + 1

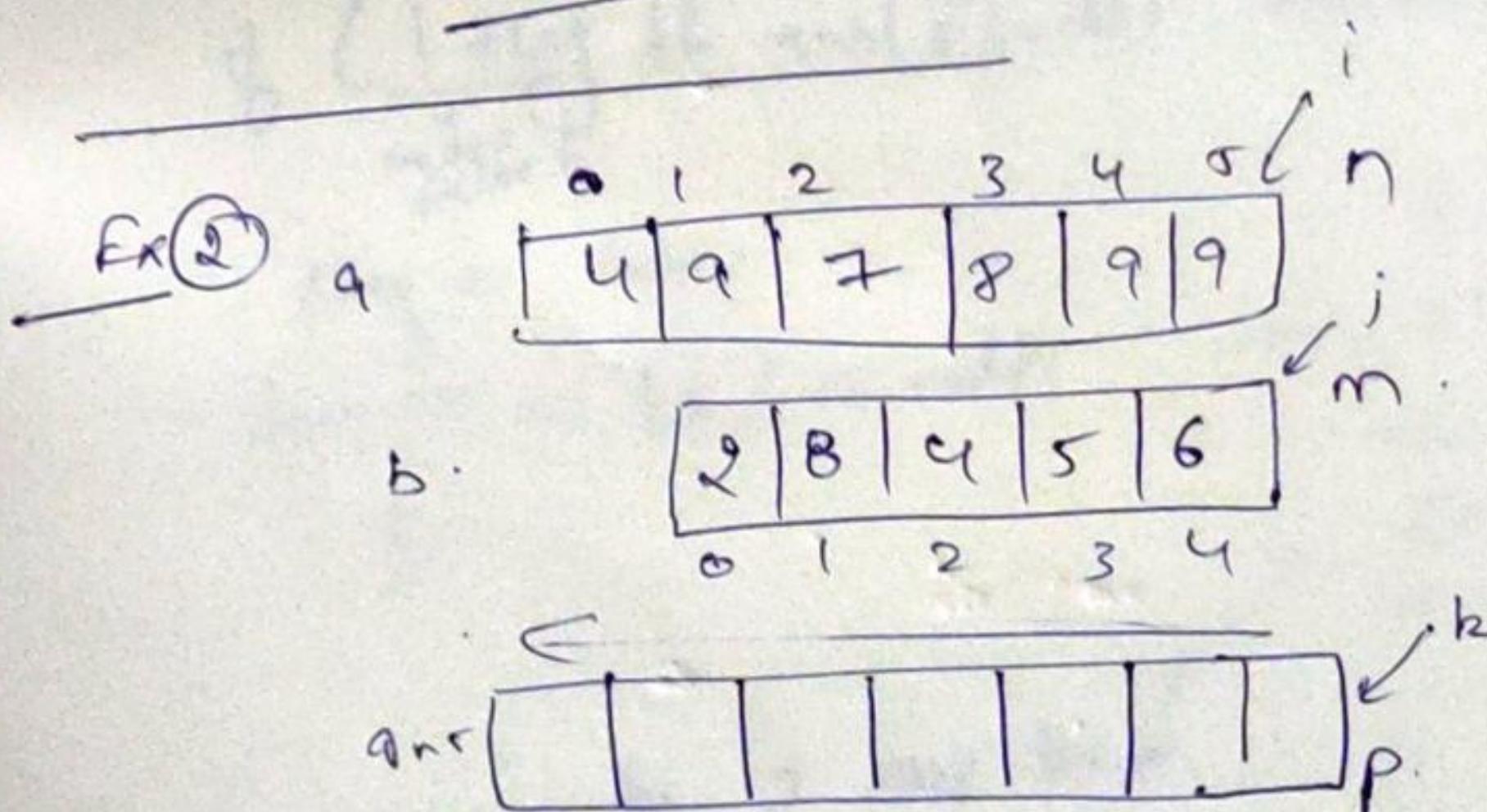
sum = Garry + n1 + n2.

sum = % 10.

sum /= 10; → Garry

$$\text{Sum} = \frac{12}{2} \quad \text{quotient ie Garry} \\ \text{Rem}$$

$$\text{Sum} = \frac{13}{3} \quad (1) \rightarrow \text{Garry} \\ \text{Rem}$$



int i = n-1, j = m-1, k = p-1
while (a[i] != 0 || j != 0 || Garry != 0)
or
while (k > 0) {

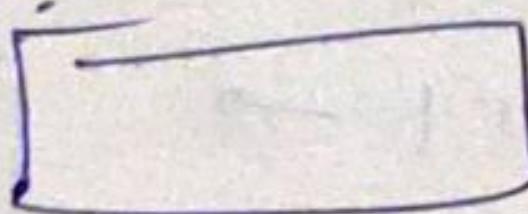
int sum = Garry +

Continue; → means niche line execute
Urke Jo bhi hongi Vo nhif hongi

Code :- Ternary Operator ^{on yeh college} Condition ? ^{True} S1 : S2 → else

Q. Subtraction of 2 Arrays. Always n27n1

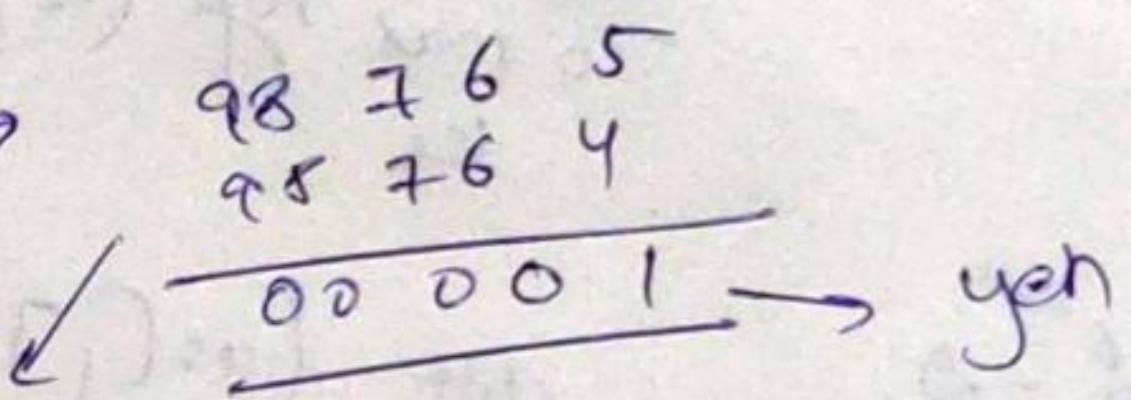
$$92 \rightarrow \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline 3 & 1 & 9 \\ \hline 9 & 1 & 4 \\ \hline 0 & 1 & \\ \hline \end{array}$$

ans 

$$\begin{array}{r} \cancel{6} \cancel{2} \cancel{3} \\ - \cancel{4} \cancel{8} \\ \hline \end{array}$$

$$0 + 3 - 8 = \cancel{-5} + 10.$$

some 0 wall problem
aaegi n Nhi

One handle →  base handle break.

boolean flag = false;
for (int i=0; i< ans.size(); i++) {

if (!flag && ans[i] == 0) continue;

flag = true;

cout << ans[i] << endl;

3

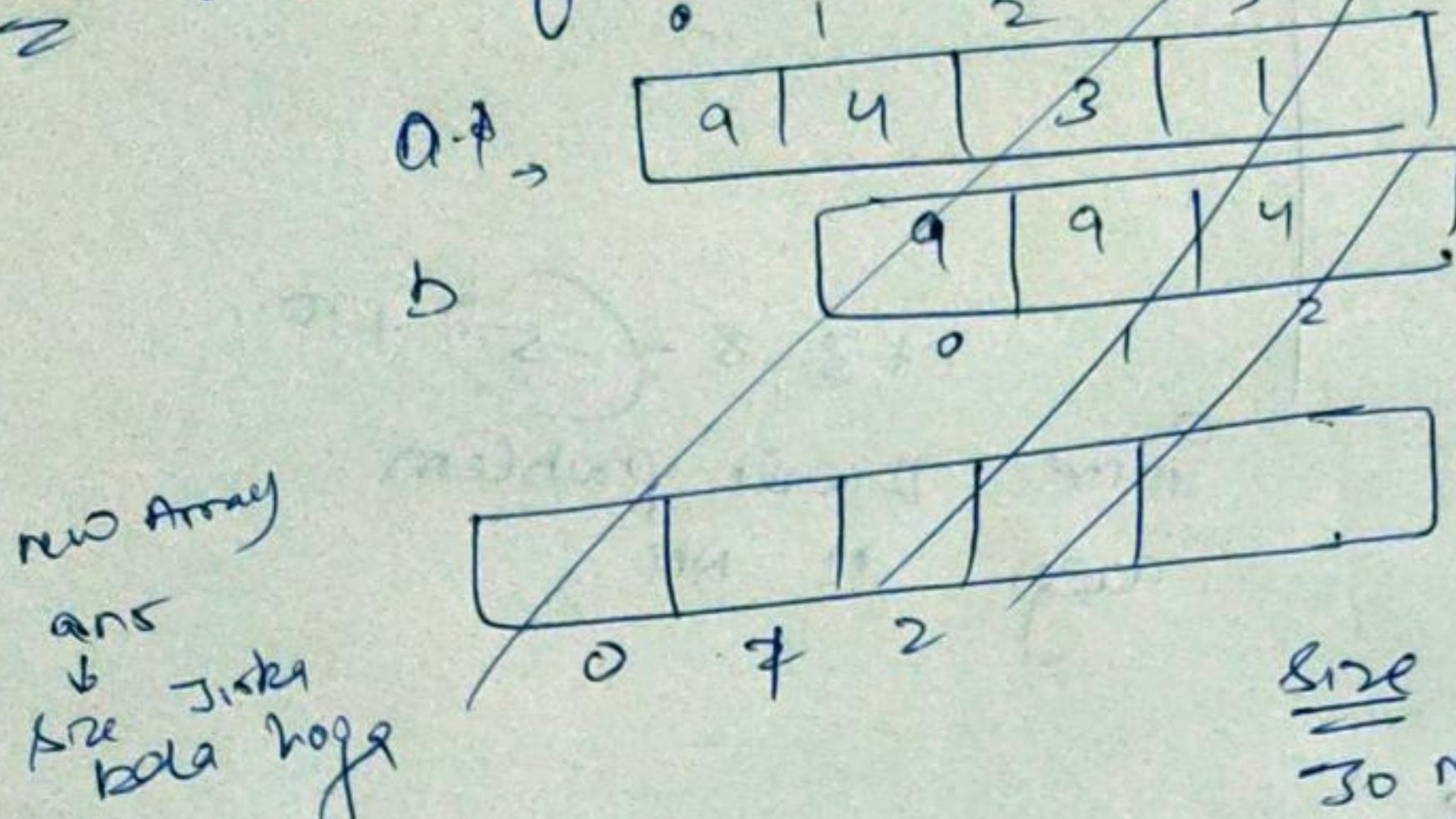
startly be 3 skip base
kooge

Continue means next digit aklege
next iteration krega ab yeh

$$\begin{array}{r} 987654321 \\ - 987554321 \\ \hline 0001000000 \\ \downarrow \\ \text{0/p} \rightarrow 100000. \end{array}$$

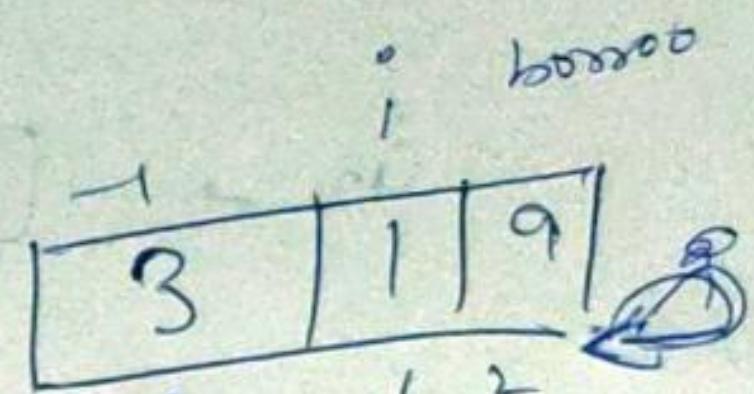
Repetit Difference

O. Sum of 2 Arrays

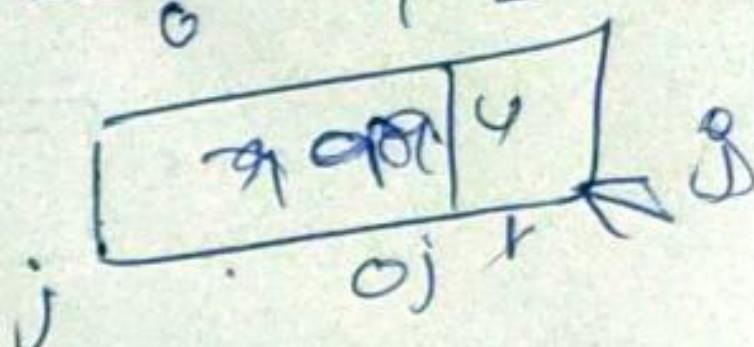


$n_2 > n_1$

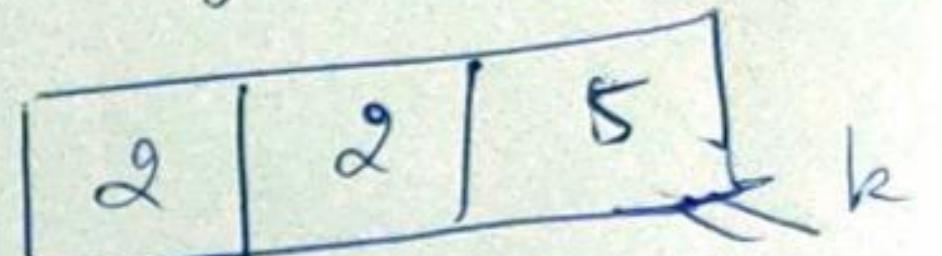
$a_2 \rightarrow$



$a_1 \rightarrow$



ans



size
30 max hogya (a_2, a_1)

$$diff = 9 - 4 = 5$$

$$diff = 1 - 9 = -8 + 10 = 2$$

int i = n-1, j = m-1, k = ans length - 1;

int borrow = 0;

while ($borrow \neq 0$) {

 int diff = $a_2[j] - a_1[i] + borrow$;

 if (diff < 0) {

$$diff = (a_2[j] - borrow) - a_1[i]$$

$$(3-1) - 9[-1]$$

Index k
value
-ve nhi
noskti h
Range me hogya tb hi

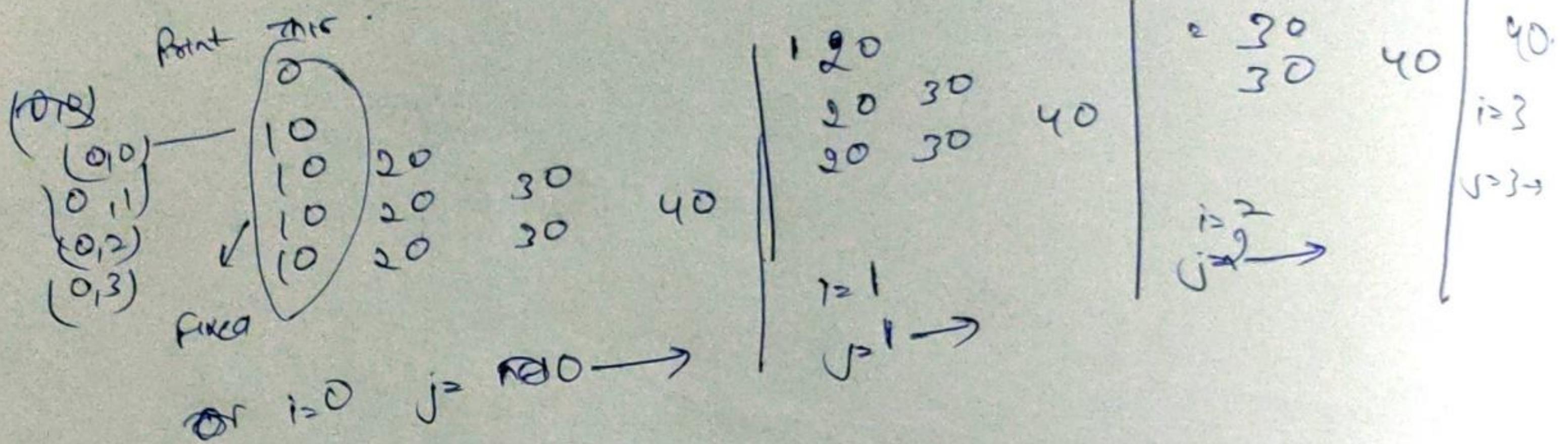
($a_2[j] - a_1[i] + borrow$)

TIME COMPLEXITIES (DS)

- ① Shifting in Array $\rightarrow O(n)$.
- ② Shifting in LL Doubly LL $\rightarrow O(1)$.
- ③ HashMap \rightarrow Shifting & all op's in $O(1)$.

Q. Subarray Problem

0	1	2	3
10	20	30	40



Q2) Given array

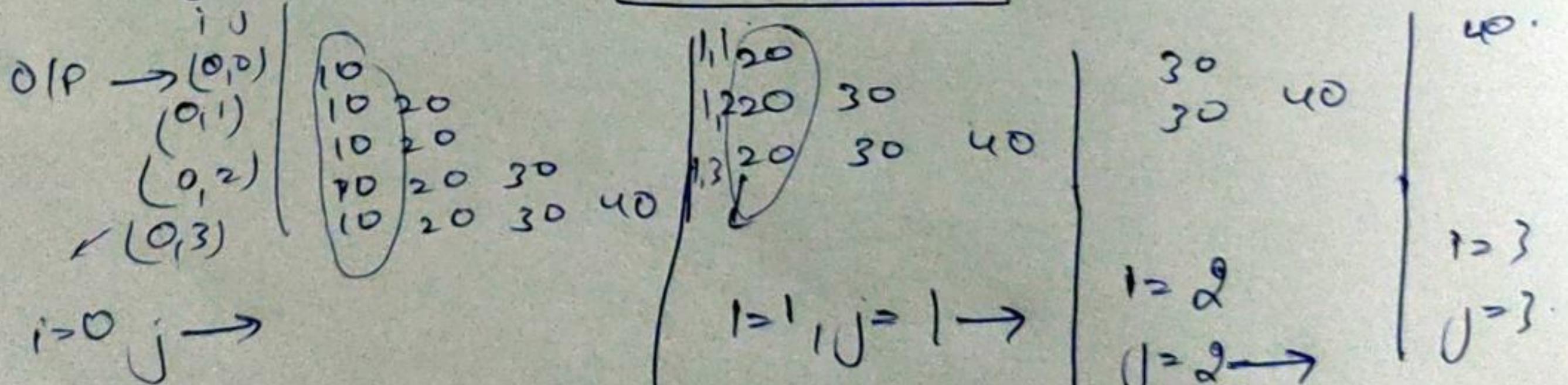
Given queries $\rightarrow \begin{cases} (3,6) \text{ ka sum karo} \\ (4-10) \text{ " " " } \\ \vdots \end{cases}$

\Rightarrow yet sb queries
resolve kon h.

Constraints
 $1 < \text{arr. length} \leq 10^6$.
 $1 < \text{query} \leq 10^6$.

Q3 Subarray Problem

0	1	2	3
10	20	30	40



for (int i=0; i<arr.length; i++) {
 for (int j=i; j<arr.length; j++) {
 cout << arr[j] << " ";
 }
}

~~for~~ (cout << arr[j] << " ";
 }
}

means 3 loop gege

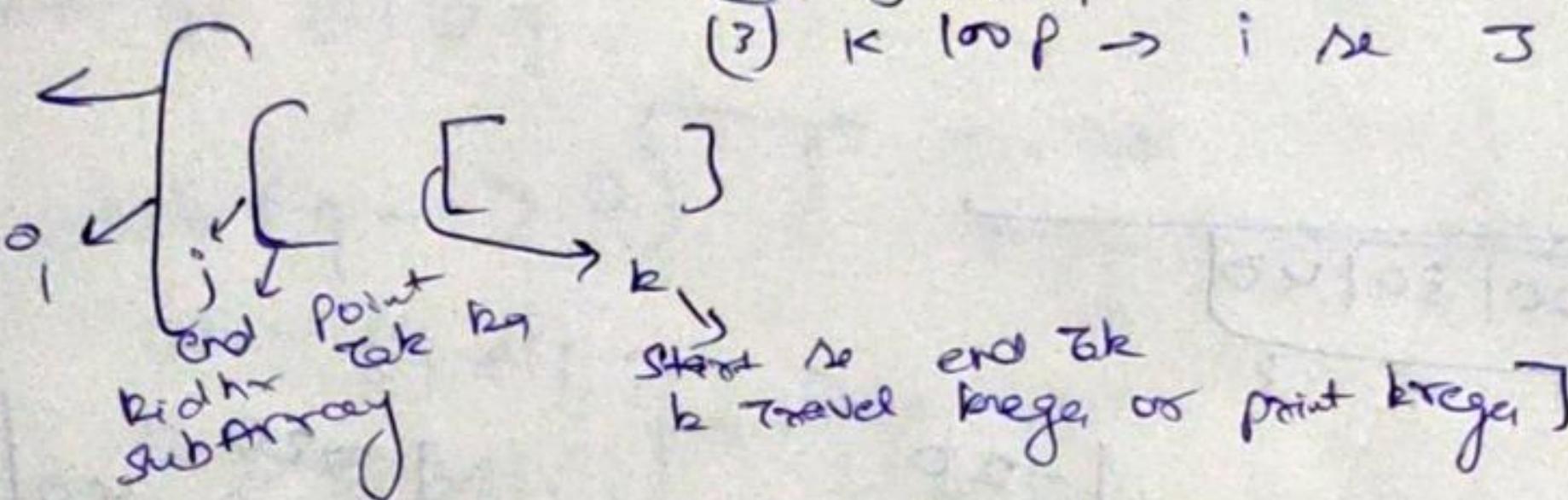
$i=0 \ j=\phi \ K^30$ $\boxed{10 \ 20 \ 30 \ 40} \rightarrow$ 2 for loop which will print how we don't want - so we need more for loop

```
for (int i=0; i<arr.length; i++) {
    for (int j=i; j<arr.length; j++) {
        for (int k=j; k<=j; k++) {
            cout << arr[k] << " ";
        }
    }
}
```

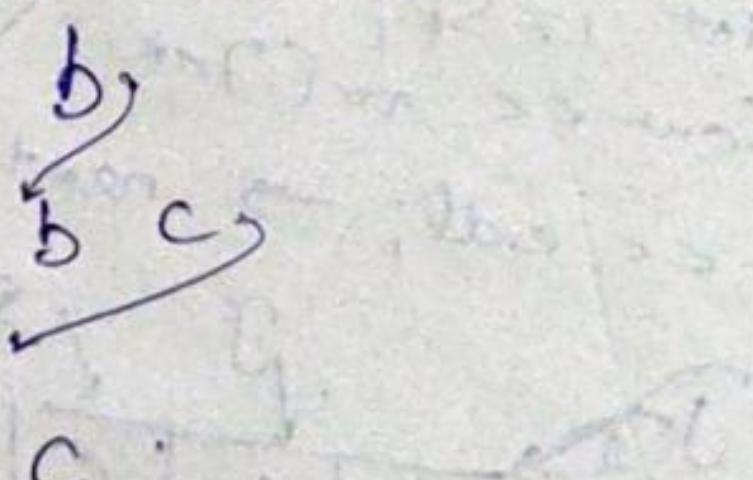
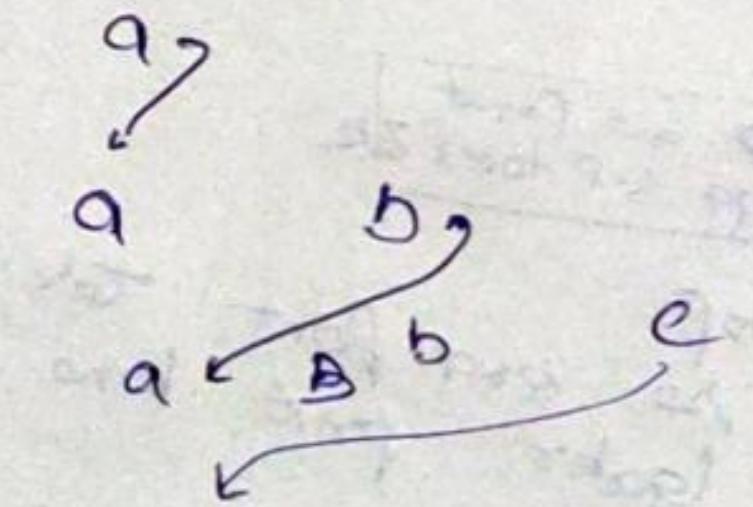
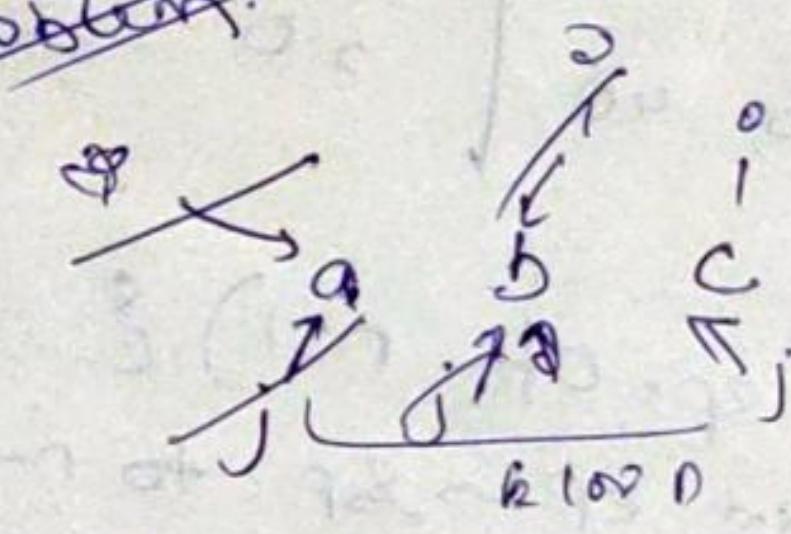
cout << endl;

}

Eg: $\boxed{a \ b \ c \ d}$ $j = 10$
 To print array for
 (1) 1st loop \rightarrow i.e. $a \ b \ c \ d$ all elements to be printed
 (2) 2nd loop \rightarrow i.e. $a \ b \ c \ d$ all elements to be printed
 (3) 3rd loop \rightarrow i.e. $a \ b \ c \ d$ all elements to be printed



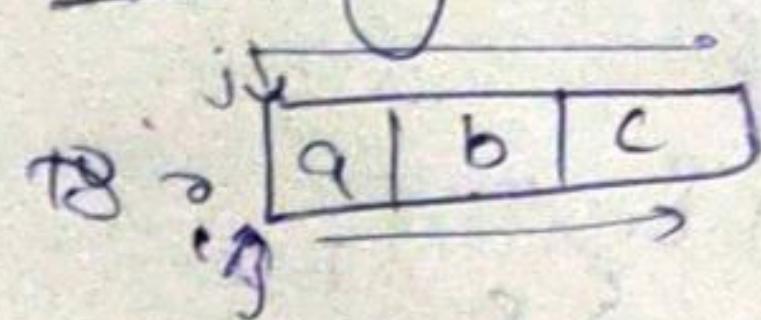
Q5 Given queries Problem.



Repeat Dry Run

Subarray Problem

Ex :-



1st for loop \rightarrow ? To travel the array

① (int i=0; i<arr.length(); i++) {

2nd for loop \rightarrow To print the element. (int j=i; j < arr.length(); j++) { \rightarrow Only these print counts print arr[j] a b c.

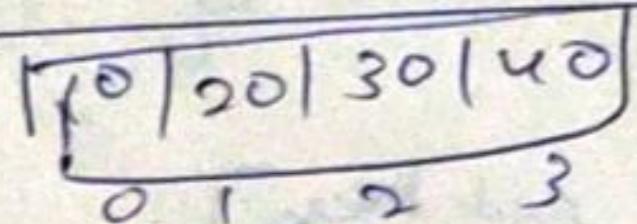
To

3rd loop \rightarrow To travel & print
(int k=i, k <= (j); k++) {
cout <<

generalize

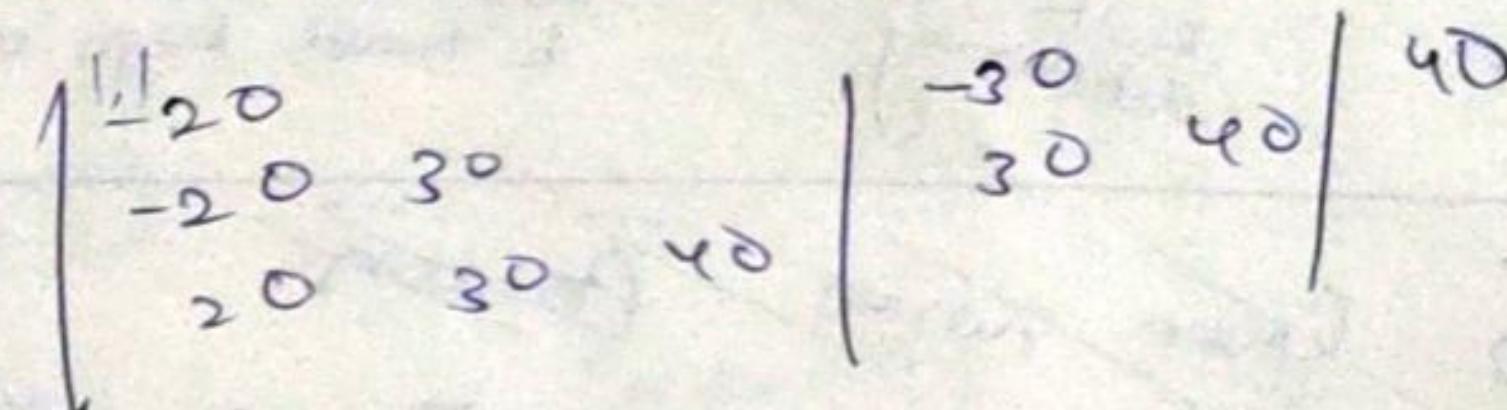
chalg

i a ab



SP. EP.

0 - 0 (0-3)	$\leftarrow 10$
0 - 1	$\leftarrow 10 \ 20$
0 - 2	$\leftarrow 10 \ 20 \ 30$
0 - 3	$\leftarrow 10 \ 20 \ 30 \ 40$



for (int i = sp = 0; i < n-1) {
 for (int ep = sp + 1; ep < n) {

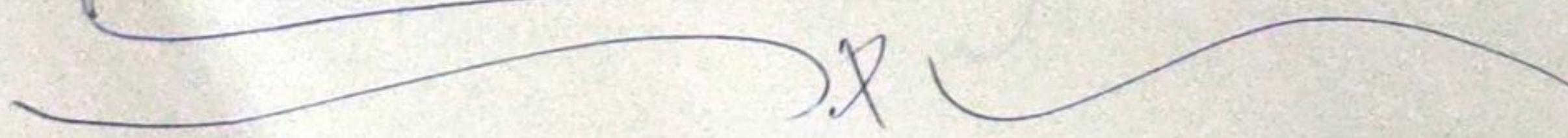
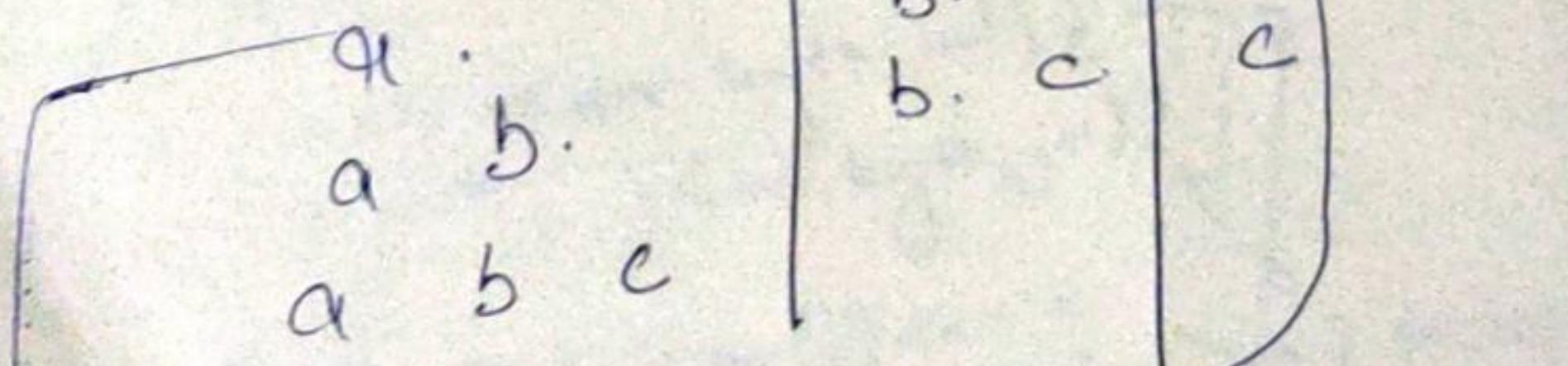
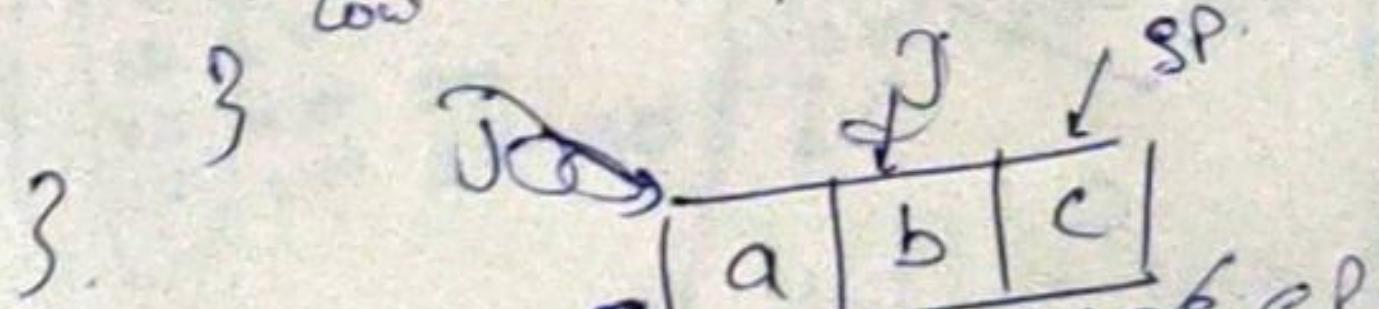
Inse simple array ki
increasing point for
chalg but
hme array ki
chenon point hne
so 1 more for loop

loop and print
chalg SP to EP

3rd loop:
SP as end point Take 1st
gather point from h

for (int i = SP; i <= EP; i++) {
 cout << arr[i] << " ";
}

} cout << endl; \rightarrow next subarray.



Out que resolveQuery() {
→ array ba input $\text{len} < \text{30 Array}$
dys h or fir kar kar
query input ke rhege

int n, cin >> n;
vector<int> arr(n, 0);

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

vector<int> psum = prefixSumArray(arr); → ab yeh milgye

out qi cin >> q;
so ab query input len 1.

while (q--) { } → jb tak
query one by one.

int si, ei; } → Range Input
cin >> si >> ei; } → milgyi so apna

answer
input know.

cout << psum[ei] - psum[si] << endl;

}.

[0, 1, 2, 3]

[1, 2, 3]

[2, 3]

[3]

[4]

[5]

[6]

[7]

[8]

[9]

[10]

[11]

[12]

[13]

[14]

[15]

[16]

[17]

[18]

[19]

[20]

[21]

[22]

[23]

[24]

[25]

[26]

[27]

[28]

[29]

[30]

[31]

[32]

[33]

[34]

[35]

[36]

[37]

[38]

[39]

[40]

[41]

[42]

[43]

[44]

[45]

[46]

[47]

[48]

[49]

[50]

[51]

[52]

[53]

[54]

[55]

[56]

[57]

[58]

[59]

[60]

[61]

[62]

[63]

[64]

[65]

[66]

[67]

[68]

[69]

[70]

[71]

[72]

[73]

[74]

[75]

[76]

[77]

[78]

[79]

[80]

[81]

[82]

[83]

[84]

[85]

[86]

[87]

[88]

[89]

[90]

[91]

[92]

[93]

[94]

[95]

[96]

[97]

[98]

[99]

[100]

[101]

[102]

[103]

[104]

[105]

[106]

[107]

[108]

[109]

[110]

[111]

[112]

[113]

[114]

[115]

[116]

[117]

[118]

[119]

[120]

[121]

[122]

[123]

[124]

[125]

[126]

[127]

[128]

[129]

[130]

[131]

[132]

[133]

[134]

[135]

[136]

[137]

[138]

[139]

[140]

[141]

[142]

[143]

[144]

[145]

[146]

[147]

[148]

[149]

[150]

[151]

[152]

[153]

[154]

[155]

[156]

[157]

[158]

[159]

[160]

[161]

[162]

[163]

[164]

[165]

[166]

[167]

[168]

[169]

[170]

[171]

[172]

[173]

[174]

[175]

[176]

[177]

[178]

[179]

[180]

[181]

[182]

[183]

[184]

[185]

[186]

[187]

[188]

[189]

[190]

[191]

[192]

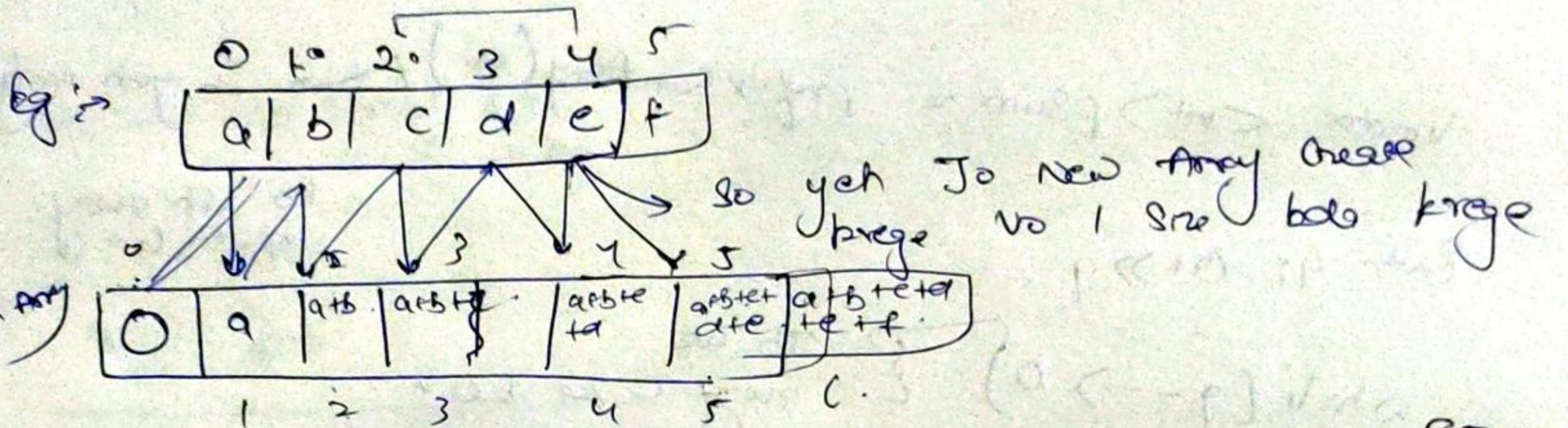
[193]

</

Q Given query problem

Q. given array \rightarrow

given queries $\rightarrow (3, 6)$ be Samedad
 $(4, 9)$ "



$$\text{psum}[i+1] = \text{psum}[i] + \text{arr}[i]$$

queries
 \downarrow
 $(2, 5)$ in be bich kg
 addition.
 $a+b+c+d$.

$$a+b=9.$$

$$a+b+c=a+b+d$$

$$a+b+c+d+e - a+b.$$

$$\text{psum}[e+1] - \text{psum}[s].$$

Eg $\rightarrow (1, 5)$

$$\begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix}$$

$$(2, 3) \rightarrow \text{psum}[4] = P[2] = a+b+c+d - a+b. \quad \text{cd}.$$

more sum ie 4
 wale kg
 mehr h.

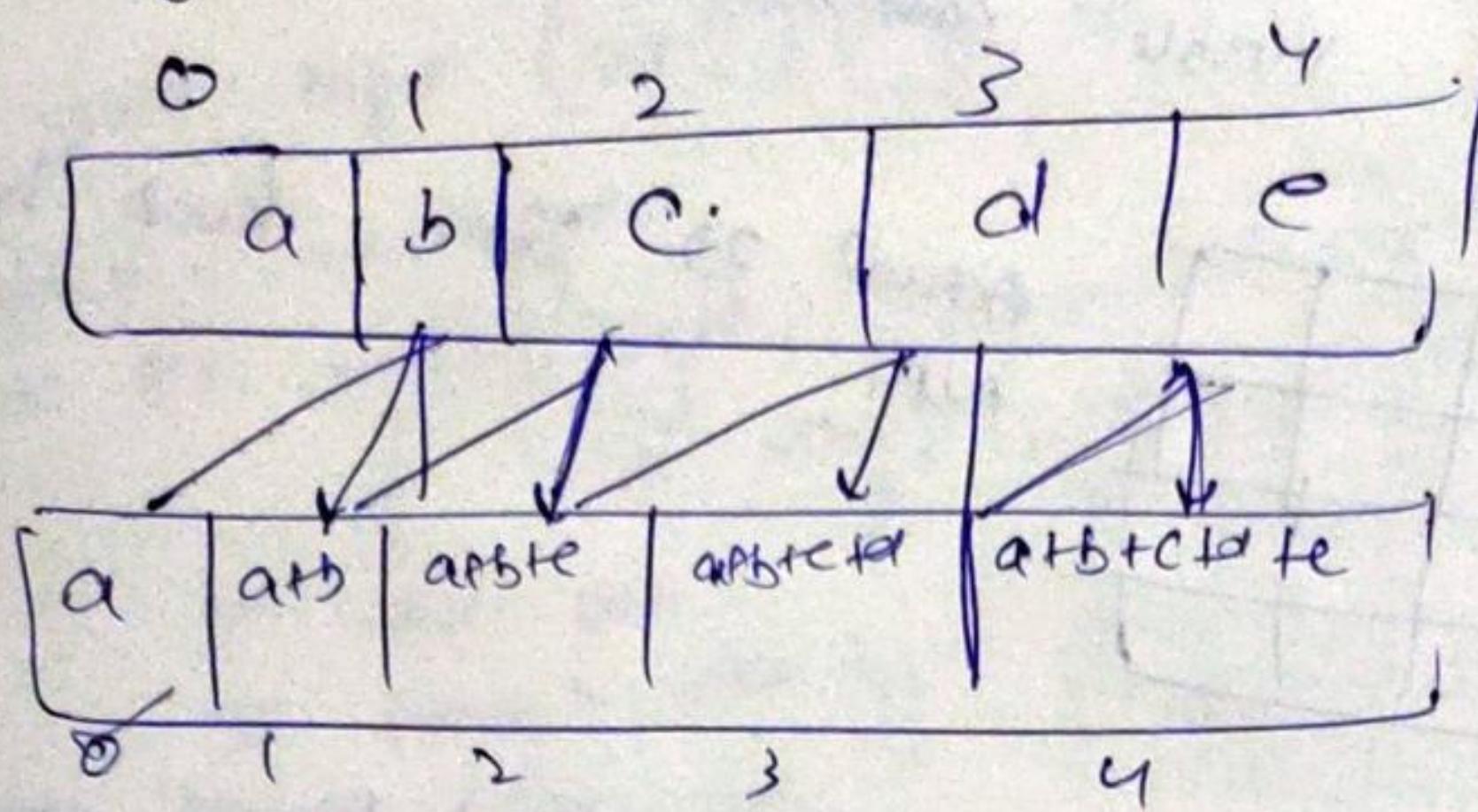
$$(1, 5) \rightarrow \text{psum}[6] - P[1]$$

run sum. 6th for h.
 $a+b+c+d+e+f - a$
 Antrag ke dekkoje

$$b c d e f \rightarrow \text{ans}.$$

$$0, 5 \rightarrow (a+b+c+d+e+f) - (0)$$

Q. Why start with 0? new Array?



Sum of
2, 4 → ie c, d, e
 $pSum[4] - pSum[2-1]$
~~ab/btstate~~ - ~~ab~~
 \boxed{cde} .

But $(0, 4) \rightarrow$ ~~dp~~ \rightarrow abcde
 $pSum[4] - \boxed{pSum[0-1]}$ → Index is Out of Bound

$$abcde - 0 = ab$$

