

Revision

- Digit Traversal \rightarrow find last digit ($\% 10$)
- Number Theory \rightarrow remove last digit ($/ 10$)
- Arrays

for (0, 1, n, i++)
i % 10

num / 10

Arrays \rightarrow collection of same data type

- \rightarrow static
- \rightarrow continuous

int \rightarrow 0

char \rightarrow null

double \rightarrow 0.0

boolean \rightarrow false

Syntax for declaration

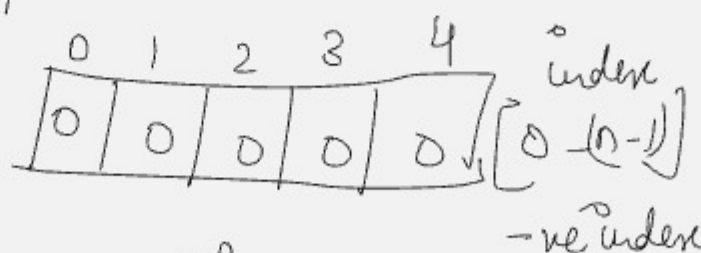
int n = s.nextInt();

int arr[] = new int [5];

size of array
n = 5

int \rightarrow 4 byte
5 x 4 = 20

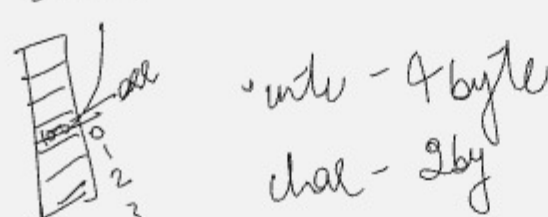
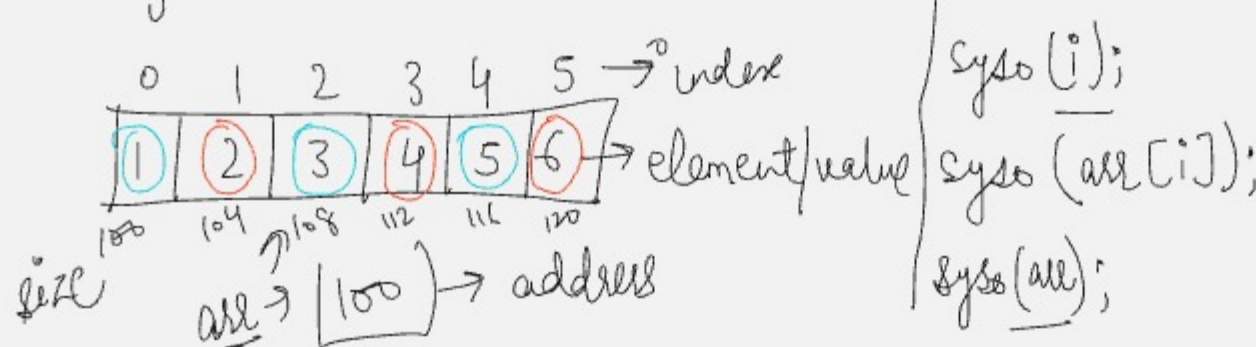
i/p, o/p \rightarrow loops



for (int i = 0; i < n; i++) {
int num = s.nextInt(); // i/p
}

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

arr[2];



arr[i] % 2 == 0
syso(i)

syso(arr[i])

syso(arr);

[1, 2, 3, 4, 5, 6]

test cases

scan

\rightarrow take Input

for ()

\rightarrow print()

for ()

print(arr)

syso ()

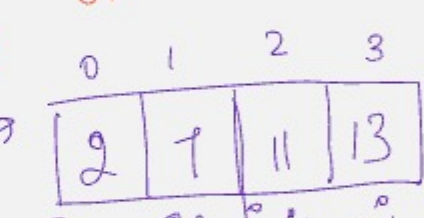
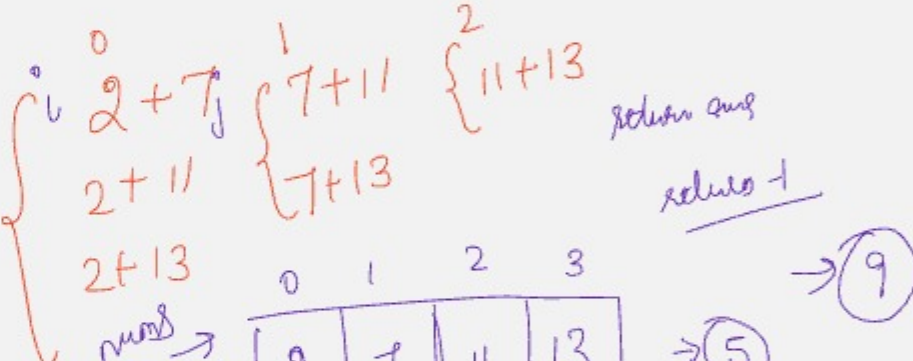
Two Sum

given \rightarrow array, target
nums = [2, 1, 11, 13] target = 9

$$x + y = t$$

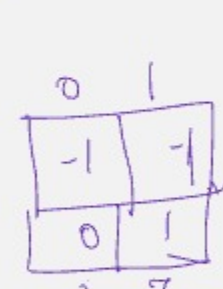
int arr[] = new int[n];
arr[0] = arr[1] = -1;

for (int i = 0; i < n; i++) {
for (int j = i + 1; j < n; j++) {
if (nums[i] + nums[j] == target) {
arr[0] = i;
arr[1] = j;
return arr;
}
}
return arr;



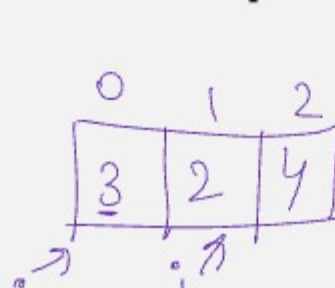
int arr[] = new int[2];
arr[0] = arr[1] = -1;

o/p = [0, 1]



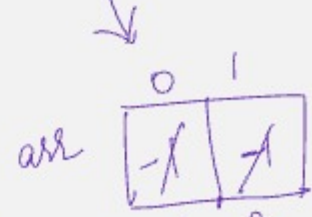
arr = [0, 1] [0, 0] [2, 1] [-1, -1] [0, 2] [0, 1]

Input: nums = [3, 2, 4], target = 6



i = 0
j = 1
3 + 2 = 5
3 + 4 = 7

return arr [1, 2]



2 + 4 = 6
6 = 6
arr[0] = 1
arr[1] = 2