JAVA

CODING

Arrays & Strings.

» All Questions needed are solved by me in C++ - logic given in previous pages.

Extra pages left for practice too.

» Syntax differences Index

① `cout<<x<<y<<" ";` → `System.out.println(x+y+" ")`
skips line ;

② `int t;`
`cin>>t;` → `Scanner sc = new Scanner(System.in);`
`int t = sc.nextInt();`

③
```
int a[10,000];
for(---){
    cin>>a[i];
}
```
→
```
int[] a = new int[10,000];
for(---){
    a[i] = sc.nextInt();
}
```

④ obj formation not req for calling fn
`x = fnname(--);` → `ClassN objName = new ClassN();`
`x = objName.fnname(--);`

⑤ memset → Arrays.fill.

⑥ swap is same but swap(listname, i, j) format.

⑦ `max(a,b)` → `Math.max(a,b);`
`x = INT_MIN;` → `x = Integer.MIN_VALUE;`

→ Method 1 →

```
temp = arr[end];
arr[end] = arr[start];
arr[start] = temp;

start++;
end--;
```

even two for loops can be used.

→ Method 2 → Using StringBuilder.

→ Method 3 → Using toCharArray & adding to a new string in reverse loop.

→ Method 4 → Reversing a number (Given before see once.)

Problem 2 → Min & Max in an array

```
for (...){
        if (arr[i] > temp){
                temp = arr[i];}
        if (arr[i] < temp2){
        temp2 = arr[i];}
}
```

temp0 = INT_MAX

temp1 = INT_MIN

```
pair out = new pair(min, max);
return out;
```

Problem 3 → Printing a array & Kth smallest element.

① → sort & k-1 element is Kth smallest.

```
Arrays.sort(arrname);
```

① →
```
for(....){
    System.out.println(i);
}
```

② →
```
for (int x : arrname){
    System .....(x);
}
```

③
```
System... (Arrays.toString(arrname));
```
↳ will print as a List/string.

Problem 4 → Sort 0s, 1s & 2s without any algo sort

M1 → 3 for loops,
```
for (....){
    if(arr[i]==0){
        a[i] = 0
        i++ }
}
return a[i];
```
3 times. (with 1 & 2 too).

M2 → count number of 0,1 & 2s & print using for loop.

M3 → Maintain high, low & medium (like in Dutch national flag problem).

Problem 5. → More all neg num to one side.

M1 → Sort

M2 → Last problem approach (or) if (i<0)
2 lists.

Without extra space

↳ Bubble sort but instead of arr[i] < arr[i+1]

do arr[i] < 0;

Problem 6 → Union of 2 arrays (& intersection)

M1 → Convert to set use HashSet

M2 → Convert to set (optional) use Hashmap

M3 → while (i<n ~~&&~~ j<m) → using 1 while & 2 loops.

```
{  if(a[i] > b[j]){ j++; }
else if( b[j] > a[i]){ i++; }
   else{( System.out...(a[i])); }
{.
```

NumberFormatException

Integer.ParseInt(String) → for string to int.

s.add(b[i]); → adding array to set.

s.toString → set or int to string.

Problem 7 → Rotating an array

```
(int i=1 . . . . . . ){
    arr[i] = arr[i-1] }
        arr[0] = arr[n];
```

Problem 8 → Largest sum Contiguous Subarray.

(Kadan's Algo). V.Imp

```
int check=0;
int max = Integer.MIN_VALUE;

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

    if (check < arr[i]){
            check = arr[i];
    }
    if( max < check){
        max = check;
    }
}
return max;
```

Can also be written as Math max (check, check + arr[i])

||

check

Problem 9 → Minimise max difference b/n heights

Q. Adding (or) sub k with each ele of array then returning max - min.

```
1) Sort Array.
2) var1 = arr[n-1] - arr[0];
3) smallest → arr[0]+k;
4) largest → arr[n-1] - k;
5) int min=0, max=0;
6) for ( - - - - ) {
       min = ~~Math~~ min (smallest, arr[i+1] -k);
       max = ~~Math~~.max ( largest, arr[i] + k);
           if (min < 0) { continue; }
           ans = Math.min (ans, max-min).
   }
7) return ans;
```

Problem 10 → Min no. of jmps to reach end of array

Rough

i = 3; x = 1

jump = 1

= 3

i = 9

n = 11

i = 3

n = 11-3-1

= 7

i = 3+1 = 4

x = 4

→

Problem 11 → **find duplicate** (pigeon hole principle n+1 members & n distinct).

→ 2 loops → naive approach. ($O(n^2)$)

→ Hashset & Hashmap. (contains)

→ Sort, and arr[i]==arr[i+1]. O(logn)

→ Binary search. (Given in leetcode, see once).