

## DSP - Array - 40 codes

1. Largest element in array.

$$arr = [1, 2, 3, 5, 4]$$

```
max = arr[0];
for (j = 0; j < size; i++) {
    if (arr[j] > max)
        max = arr[j];
print(max)
```

2. Second largest element in an array  
without

```
max = arr[0];
for (i = 0; i < n; i++) {
    max = max(max, arr[i]);
}
max2 = 0;
for (i = 0; i < n; i++) {
    if (max2 < arr[i] && max2 != max)
        max2 = arr[i];
}
print(max2)
```



3.

check the array is sorted.

```
a = [1, 2, 3, 5, 6, 7]
for (i=0; i<n; i++) {
    if (a[i] <= a[i+1])
        return 0;
    else {
        print("array is not sorted.");
        break;
    }
}
print("array is sorted.");
print("array")
```

4.

Remove duplicate from sorted array

```
a = [1, 2, 3, 4]
for (i=0; i<size-1; i++)
```

```
    if (a[i] == a[i+1]) { while (i < size) {
```

```
        a[i+2] = a[i+1];
        i++;
    }
```

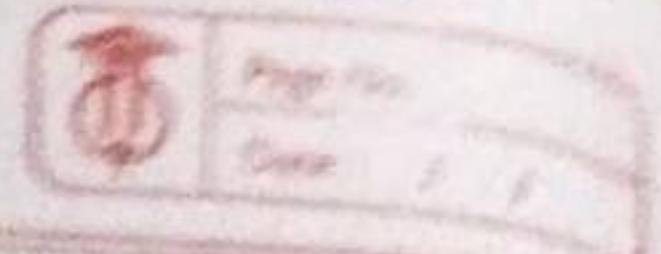
```
    if (i=0, size=1)
```

```
        if (a[i] == a[i+1]) {
```

```
            for (j=i+1; j<n-1; j++)
```

```
                a[j] = a[j+1];
```

swap( $a$ ) =  $01101010 \oplus 11111111$



5 Left rotate an array by one place.

$a = [1, 2, 3, 4, 5]$   $q_p = [2, 3, 4, 5, 1]$   
 $n = a[0]$   
for ( $i = 0; i < size; i++$ ) {

~~rotation~~  
 $a[i] = a[i+1];$

$a[size-1] = n;$

6 Left rotate an array by D places

~~left(0, 1) / for ( $i = 0; i < size; i++$ ) {  
    ~~for ( $i = 0; i < size - i + d$ ;~~  
    ~~$n = a[i]$~~   
    ~~~~rotate(a, i, size - i + d);~~~~~~

$D = 2$   
 $[1, 2, 3, 4, 5]$   
 $[2, 3, 4, 5, 1]$   
 $[3, 4, 5, 1, 2]$

~~~~rotate(a, i, size - i + d);~~~~

<del



7. Shift all zeroes to end of the array

$$c1 = [1, 2, 0, 3, 0, 4]$$

$$\text{After } a = [1, 2, 3, 4, 0, 0]$$

$$n = \text{size}$$

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

```
    if (a[i] == 0)
```

```
        for (j=i; j<n; j++) {
```

```
            a[i] = a[j+1]
```

```
        }
```

```
        a[n] = 0;
```

```
}
```

```
print(a);
```

union of two sorted arrays

$a[4] = [9, 2, 3, 4, 5]$  size n

$b[4] = [1, 2, 3, 4]$  size m

new int a3[ $n+m$ ];

for (int i=0; i<n; i++) {

    for (int j=0; j<m; j++) {

        if ( $a1[i] > a2[j]$ ) {

$a3[i] = a2[j];$

        }

    else if ( $a1[i] < a2[j]$ ) {

$a3[i] = a1[i];$

}

    afr 3 - print ]

for (int i=0; i<n; i++) {

    bool = false;

    for (int j=0; j<m; j++) {

        if ( $a2[i] == a1[j]$ )

            bool = true;

        break;

}

    if (!bool) {

        print(a2[i]);

}



Q.

Q13. Longest subarray with given sum k (C++)

$$a = \{1, 0, 5, 1, 2, 7, 1, 9\} \quad k = 15$$

$\underbrace{(1, 0, 5, 1)}_{\text{OP}} \quad 2, 7, 1, 9$

(A)

```
int sum = 0;  
int count = 0; int result;
```

```
for (int i = 0; i < size; i++) {
```

sum += a[i];  
if (sum >= 15)

{

~~if (sum == 15)~~; if (sum == 15)  
result = c;

()

else

c++;

}

else

sum = a[i];

c = 1;



medium:

Ques 2 sum problem.

$$2+7=9$$

$$a = \{ \begin{matrix} 2 \\ 7 \\ 11 \\ 15 \end{matrix} \}$$

$$\text{target} = 9$$

$$\text{ans} = 0, 1$$

$$g = [3, 2, 4]$$

$$t = 6$$

$$a = \{ 3, 3 \}$$

```
for (int i = 0; i < size; i++) {
```

```
    for (j = 0; j < n; j++) {
```

```
        if ((a[i] + a[j]) == target) {
```

```
            return i, j;
```

```
            break;
```

```
}
```



Median - find

Given sorted array of 0's, 1's & 2's

Ex:  $\{2, 1, 0, 1, 1, 0\} = \text{arr},$

$\Delta \text{Q/P} = [0, 0, 1, 1, 2, 2]$

$\text{mid} = 0;$

$\text{arr}_2 = [2, 0, 1]$

$\text{Q/P} = [0, 1, 2]$

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

$\Delta \left\{ \begin{array}{l} \text{if } (\text{a}[i] > \text{a}[i+1]) \\ \text{swap } (\text{a}[i], \text{a}[i+1]) \end{array} \right\}$

int low = 0, mid = 0, high = n - 1

while ( $\text{mid} > \text{high}$ ) {

    if ( $\text{a}[\text{mid}]$ ) {

        Case 0:

            swap ( $\text{a}[\text{mid}], \text{a}[\text{low}]$ )

~~mid++~~ low++ mid++

        ② break;

    Case 1:

        mid++;

    ③ break;

Case 2:

    swap ( $\text{a}[\text{high}], \text{a}[\text{mid}]$ )

        high--;

3



Ques

majority element :

$$ar = [2, 2, 1, 1, 1, 2, 2]$$

```
int n = size/2
for (int i=0; i < size; i++) {
    int c=0;
    for (int j=0; j < size; j++) {
        if (ar[i] == ar[j])
            c++;
    }
    if (c > n)
        return ar[i];
}
return -1;
```

Ques

## Rearrange Array Elements by sign

C++:- arr = [3, 1, -2, -5, 2, -4]

Op: [3, -2, 1, -5, 2, -4]  
+ - + - + -

for (i=0; i<n; i++) {  
if (a[i] >= 0) {

if (a[i] >= 22 || a[i+1] < 0) {  
c++;

new array

int p=0;  
int n=1;

for (int i=0; i<n; i++) {  
if (a[i] >= 0) {  
newarray[p] = a[i];  
p = p+2;

}

else {

newarray[n] = a[i];  
n = n+2;

3 3

1234 4323 1423  
1432  
1324

6 =  
6 = 11

ans: next permutation

$$arr = [1, 2, 3]$$

$$opp = [3, 2]$$

$$ans = [1, 3, 2]$$

ans: leaders in an array.

$$a = [4, 7, 1, 0]$$

$$opp = [3, 1, 0]$$

$$a = [10, 22, 12, 31, 6]$$
  
$$= 22, 12, 6$$

bool ans;

for (i) {

for (j) {

$$a[i] > a[j]$$

ans[~~i~~] = true;

}

if (~~ans~~ ans == true)

newarray[i] = a[i];

}



Que. Longest consecutive sequence.

a = [100, 4, 200, 1, 3, 2]

ans = 4

exp: - (1, 2, 3, 4)

1st step :- find minima.

c = 1; max = 0;

for (i) {

    for (j) {

        if (a[i] == a[j])

            c++;

    max = max(max, c);

int long = 1;

int c = 1;

for (i=1) {

    sort(a.begin(), a.end());

    if (a[i] == a[i-1]) {

        if (a[i] == a[i-1+1])

            c++;

    else {

        long = max(long, c);

        c = 1;

    }

}

return max (long);



Ques. Set matrix zero.

$$\text{ex:- } \begin{matrix} & j & j & j \\ \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} & = & \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

make entire row, & column  
zero when you find  
zero.

$a[i][j]$

$a[2][2] = 0$

$j=2$

if ( $a[i][j] = -0$ ) {

$a[i][0] = 0$

$a[0][j]$

$a[i][k] = 0$

$a[1][j]$

$a[i][l] = 0$

$a[2][j] = 0$

Using four for loops.

for (int i=0) {

    for (int j=0) {

        for (k=0) {

~~$a[i][k] = a[i][k] = 0$~~   $a[i][k] = 0$  ?  $0 :- 1$

}

        for (K=0) {

~~$a[k][j] = a[k][j] = 0$~~   $a[k][j] = 0$  ?  $0 :- 1$

    }

}

    for (j=0) {

        if ( $a[i][j] = -1$ )

$a[i][j] = 0$

}

}



Ques. Rotate 90 degree the matrix.

ex:-

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

i> transpose:

for (int i=0) {

    for (j) {

        swap (a[i][j], a[j][i]);

}

Now matrix is:

$$\begin{array}{ccc|cc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \quad \text{size-1} \quad \text{swap}$$

reverse the row.

int a=0;

int b=size-1;

for (int i=0) {

    for (int j=0) {

        if (a[i][j] < a[b][j]) i=a[i], j=

Swap<sup>mat</sup> mat[i][j], mat[b][j])

    a++; b--;

}

while(a < b) {



Ques: Spiral matrix.

ex:-  $i=0 \begin{bmatrix} 1 \rightarrow & 2 \rightarrow 3 \downarrow \\ \downarrow & 4 \rightarrow 5 & 6 \downarrow \\ i=2 \begin{bmatrix} 7 \rightarrow 8 \leftarrow 9 \\ \end{bmatrix} \end{bmatrix}$  op:- [1, 2, 3, 6, 9, 8, 7, 4, 5]

$n = \text{size};$

~~while ( $\bullet \neq 1 = 0$ ) {  
for ( $j = 0$ ) {  
if ( $j = \text{size}$ ) {~~

Ques: Count subarray with given sum.

ex:- arr = [1, 2, 3],  $\leftarrow \text{sum}$ ,  $K = 3$

int c = 0; // count

for (int i = 0) {  
 int sum = 0;  
 for (int j = 0) {  
 sum += arr[j];  
 if (sum == k) c++;

3 3

print(c);

Hard

Ques find majority element ( $n/3$  times)

Code is just like  $n/2$  times.

new[i];

int n = size/3;

for (int i=0) {

int c=0;

for (int j=0) {

if (a[i]==a[j]) (++)

}

if (c > n).

print(a[i]); or new[i] = a[i];

}

bool ans = false.

for (K) {

if (new[K] == a[i]) {

ans = true;

break;

}

if ((n && !ans))

new[i] = a[i];

}

}

}

to check  
whether

element  
is already

present in  
new arr

or  
removing  
duplicates