

C20CS100

ADITYA RAJ

Algorithm :

(a)

1. start

2. Input size and elements in two array and store separately in two array variable.

```
int arr1[size1], arr2[size2], merge arr[size3],  
int size1, size2, merge size;
```

3. Creating another array which will store merge array with size.

mergeSize = size1 + size2.

4. Now, Initializing two variable index1 = 0 and index2 = 0.

5. We will run loop from 0 to mergeSize..

for (mergeIndex = 0; mergeIndex < mergeSize; mergeIndex++).

6. Now, we will check smallest element in two input array inside the loop.

if (arr1[index1] < arr2[index2])  
then assigning element of first element to merge array. i.e.,

mergeArray[mergeIndex] = arr1[index1]  
with increment index1.

else { mergeArray [mergeIndex] = arr2[index2];  
index2++;  
}.

9) After loop merge the remaining array element.

10) stop.

Dry run :

Input

Input first array elements: 2, 4, 8, 9, 12

Input second array elements: 3, 5, 6, 10

Output

merged array in descending order = 12, 10, 9, 8, 6, 5, 4,  
3, 2.



- ②
- 1> start
  - 2> Read the elements and store into the array  $a[]$  as  $\text{scanf}("%d", \&a[i])$  using for loop  
 $\text{for}(i=0; i < n; i++)$ .
  - 3> Find the repeated elements in the array.  
 for loop iterates from  $i=0, i < n, c=1$ .
  - 4> If  $a[i] \neq -1$ , then compare each element with remaining elements of the array.  
 $a[i] == a[j]$ , using for loop  
 $\text{for}(j = i+1; j < n; j++)$   
~~for~~
  - 5> If any element is equal to other element of array, then,  
 ~~$c = c + 1$~~   
 $\text{if}(a[i] == a[j])$   
 $\{ c++ ;$   
 $a[j] = -1 ;$   
 $\}$
  - Repeat till  $j < n$ .
  - 6> store value of  $c$  into  $b[i]$ .  
 Repeat  $i < n$ .



7) Print unique element which are having count value is 1, using for loop.

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

8) STOP.

Org run

Enter size of array : 5

Enter element in array : {1, 2, 2, 4, 3}

unique no. in array : {1, 4, 3}

8) 1) Start

2) Read and store array elements in array `a[]`, and count total using `scanf` and for loop.

```
for (i=0, i<n, i++)
```

```
{  
    scanf("%d", &a[i]);
```

```
    if (a[i] % 2 == 1)
```

```
        c++;
```

3) Arrange the arrays in ascending order.  
using for loop.

```
for (i=0, i<n-1, i++)
```

4) ~~then inner for.~~  
Compare  $a[i]$  and  $a[i+1]$ ,

5) If  $a[i]$  is highest than  $a[i+1]$  then swap the element.

6) Repeat the until all iteration of  $i$ ,  
followed by  $i < n-1$

7) ~~initialise~~ separating odd and even number.  
by initialising  $k = 0$  &  $j = n-1$ .

8) for  $(i = 0; i < n; i++)$

8) If  $a[i]$  is even num, ~~if~~  $k < n-c$ ,  
 $b[k++] = a[i];$

If  $a[i]$  is odd num, if  $j < n$ ,  
 $b[j++] = a[i];$

Now,  
9) ~~the~~ array  $b[]$  contains even num from  
 $i = 0$  to  $n-c-1$ , and  
odd num from  $n-c$  to  $n-1$

or element of  
10) moving  $b[]$  to  $a[]$ .

11) Now, Print using for loop.

~~for~~  $(i = 0; i < n; i++)$ .



Dry run

Enter size of the array : 10

Enter array : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Array after separating odd and even :

{0, 2, 4, 6, 8, 3, 1, 5, 7, 9}



start

2) ~~Declare~~ <sup>Read</sup> arr1[ ][ ], arr2[ ][ ] and to see it accept 2x2 matrix and check it is equal by using for loop.

for (i = 0; i < r1; i++)

{ for (j = 0; j < c1; j++)

}

and,

for (i = 0; i < r2; i++)

{ for (j = 0; j < c2; j++)

}

3) Arranging the matrices array of matrix using for loop.

4) Comparing the <sup>two</sup> matrices for equality. using if for loop.

if ( $r_1 == r_2$  &&  $c_1 == c_2$ ).

for ( $i=0$ ;  $i < r_1$ ;  $i++$ )

{ for ( $j=0$ ;  $j < c_2$ ;  $j++$ ). }

5) If the matrices of arr1  $[i][j] !=$  arr2  $[i][j]$  then flag = 0 and break.

6) If the If (flag == 1).

7) Print "The two matrices are equal".  
else Print "But, two matrices are not equal".

Dry run.

Input element in 1st matrix -  
matrix -  $[0], [0] : 1$

" -  $[0], [1] : 2$

" -  $[1], [0] : 3$

" -  $[1], [1] : 4$

Q

Input element in 2nd matrix -

matrix -  $[0], [0] : 1$

$[0], [1] : 2$

$[1], [0] : 3$

$[1], [1] : 4$



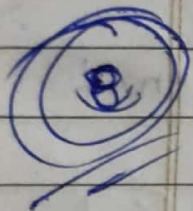
The first matrix is:

1	2
3	4

The 2nd matrix is:

1	2
3	4

two  
The <sup>two</sup> matrices can be compared are equal:



1) start

2) Declare an array, arr[] and input the array element <sup>has</sup> with size n.

3) Start traversing the array. taking i & j as a for loop.

$j = 0$   
for ( $i = 0; i < n; i++$ ).

4) If the current element is negative.

if ( $arr[i] < 0$ ).

if ( $i \neq j$ ).

5) Swap the current element with first positive element. and cont. traversing until all element have been encountered.



temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

j++;

⑤ ~~Cont. to Rearrange array using~~ wi

⑥ Rearrange and Print the array using for loop.  
for (i = 0; i < n; i++)

⑦ stop

Dry run

Enter the number of element : 6

Enter

Input the array : -1, 1, -2, 2, -3, 3.

The Rearranged array is : -1, -2, -3, 2, 1, 3.

(f) start

Declare

2) Read array  $x[]$ ,  $y[]$  and create variable  $m, n$ ,  $sum_x$ ,  $sum_y$  result.

3) transverse the array using taking  $i, j$  for looping.

while ( $i < m$  &  $j < n$ ).

④ handle the duplicate element by using while loop for both  $x$  &  $y$ .

⑤ if the Current element of  $y$  is less than Current element of  $x$ .

if ( $y[j] < x[i]$ ).

}

$sum_y += y[j];$

$j++;$

apply ~~same~~ same condition with  $x$  element.

⑥

else - Consider

if ( $x[i] == y[j]$ )

$sum += \max(sum_x, sum_y) + x[i]$

$i++, j++;$

and least both sum.

$sum_x = 0, sum_y = 0;$



⑦ Process the remaining elements of  $x$  &  $y$  if any.

⑧ Print the max sum <sup>with</sup> after the running the main code.

Dry Run .

arr  $x$  [] = {3, 6, 7, 8, 10, 12, 15, 18, 100} ;

arr  $y$  [] = {1, 2, 3, 5, 7, 9, 10, 11, 15, 16, 18, 25, 30}

~~$m = \text{size of } (x) / \text{size of } (x[0]) ;$~~

~~$n = \text{size of } (y) / \text{size of } (y[0]) ;$~~

The max sum is = 199

199 is sum of 1+2+3+6+7+9+10+12+15  
+ 16+18+100 .

⑨ → start

1) Declare the <sup>sorted</sup> array arr[] .  $x$ , diff. & result.

3) Initialize the ~~arrays~~ variable diff .  
as infinite :  $\infty$

4) Initialize two index variable  $i$  and  $j$  in the given sorted array.

5) Initialize first & second most index.

$l = 0, r = n - 1, diff = Int \cdot \max$

6) while there are element between  $l$  &  $r$ ,  
use while loop.

while,  $(r > l)$

7) If  $abs(arr[l] + arr[r] - sum) < diff$ .  
then update  $diff$  & result.

8) If  $arr[l] + arr[r] < sum$  then  
++  
else,  $r--$

Dry + sum

Input :  $arr[] = \{10, 29, 28, 39, 80, 88\}$   
sum = 58

output : 29 and 80