Q1. To merge two arrays of the same size sorted in descending order

**ANSWER:-**

Step 1:- Start.

Step 2:- Declare variable which takes the size of the array.

Step 3:- Declare the array A [], B [], C [].

Step 4:- Initialize the array.

Step 5:- declare and initialize ia=0,ib=0,ic=0

Step 6:- if iath element of A is greater or equal to ibth element of B and ia is smaller than array size, set icth element of C to ibth element of B and increase ia by 1.

Step 7:- else set icth element of C to ibth element of B and increase ib by 1.

Step 8:- Increase ic by 1

Step 9:- Repeat Step 6,7,8 until ic is less than resulting array size.

Step 10:- Display resulting array

Step 11:- Stop

**DRY RUN (N=3)**

A[3] = {10,8,5}

B[3] = {11,7,3}

C[6] = {11,10,8,7,5,3}

step 6: 0<3 and A[0]>=B[0] false

step 7: C[0] = 11,ib = 1

step 8: ic=1

step 6: 0<3 and A[0]>=B[1] true gives C[1]=10,ia=1

step 8: ic=2

step 6: 1<3 and A[1]>=B[1] true gives C[2]=8,ia = 2

step 8: ic=3

step 6: 2<3 and A[2]>=B[1] false

step 7: C[3]=7,ib=2

step 8: ic=4

step 6: 2<3 and A[2]>=B[2] true gives C[4]=3,ia=3

step 8: ic=5

step 9: 3<3 and A[3]>=B[2] false

step 7: C[5]=3,ib=3

step 8: ic=6

step 11: stop (since 6 is not less than array size=6)

Q2. To print all unique elements in an array.

**ANSWER**:-

Step 1:- Start.

Step 2:- Read and store the entered array size into the variable n.

Step 3:- Read the elements and store into the array a[]

Step 4:- Find the repeated elements in the array using for loop iterates from i=0 i<n,c=1

a) If the a[i]!=-1 then compare each element with remaining elements of the array as a[i]==a[j],using for loop from j=i+1 to j<n.

If any element is equal to other elements of the array then increase the c value by 1.Repeat the comparison until j<n.

b) Store the c value into b[i].Repeat until i<n. The array b[] contains the count values of the array elements.

Step 5: The count value represents how many times the element repeated in the array. The count 1 means the element repeated only once in the array which is unique.

Step 6: Print the unique elements which are having the count value is 1 using for loop from i=0 to i<n.

**DRY RUN:**

a[4] = {1,2,2,3};

"1,2,3"

Step 4a:c=1,a[0]=1 != -1 then loop from j=1 to 3

step 4a: j=1 ,a[0] == a[1] means 2==1,false

step 4a: j=2, 2==1,false

step 4a: j=3, 3==1,false

step 4b: store c=1 into b means b[0]=1.

step 4a: c=1,a[1]=2 != -1 then loop from j=2 to 3

step 4a: j=2,a[1]==a[2] true c=2

step 4a: j=3,3==2 false

step 4b: store c=2 into b means b[1]=2.

step 4a: c=1,a[2]=3 != -1 then loop from j=3 to 3

step 4a: j=3,a[2]==a[3] false

step 4b: store c=1 into b means b[2]=2.

step 5: loop through array

step 6: Print values from a who has corresponding value 1 in b.

Q3. To separate odd and even integers in separate arrays.

**ANSWER**:-

Step 1:- Start.

Step 2:- Declare variable which takes the size of the array.

Step 3:- Declare the array A1 [].

Step 4:- Initialize the array.

Step 5:- Then create other two array A2 [], A3 [] which stores odd and even numbers respectively.

Step 6:- Check every elements of the given array whether it is divisible by 2 or not.

Step 7:- If it is divisible by 2 then store that element in A3 [] array and if not then store in a2 [] array.

Step 8:- Display odd and even array elements.

Step 9:- Stop.

**DRY RUN:-**

N=10

A1 [5] = {1,3,10,2,5}

A2 [3] = {1,3,5}

A3 [2] = {10,2}

step 5:- i

step 6:- a[0]= 1 which isnt multiple of 2 so add into A2 and increase i

step 6:- a[1]= 3 which isnt multiple of 2 so add into A2 and increase i

step 6:- a[2]= 10 which is multiple of 2 so add into A3 and increase i

step 6:- a[3]= 2 which is multiple of 2 so add into A3 and increase i

step 6:- a[4]= 5 which isnt multiple of 2 so add into A2 and increase i

step 8: Display result

Q4. To accept two matrices and check whether they are equal.

**ANSWER**:-

Step 1:- Start.

Step 2:- Declare the array A1 [] [], A2 [] [].

Step 3:- Initialize both the arrays.

Step 4:- Calculate the number of rows and columns in array A1 [] [] and store them in variables R1 and C1 respectively.

Step 5:- Calculate the number of rows and columns in array A2 [] [] and store them in variables R2 and C2 respectively.

Step 6:- Initialize the Flag to true.

Step 7:- Check if the size of both the arrays are same or not. If it is not same then display “Matrixes are not equal”.

Step 8:- If it is same then using for loop check all the elements of both the arrays.

Step 9:- If all the elements are same then display “Matrixes are equal”.

Step 10:- If any of the corresponding elements are not same then set the Flag to false and break the loop and display ”Matrixes are not equal”.

**DRY RUN:-**

A1 [] [] = {{9, 0, 3}, {1, 0, 2}, {6, 8, 2}}

A2 [] [] = {{9, 0, 3}, {1, 0, 2}, {6, 8, 2}}

Row1=3, Col1=3, Row2=3, Col2=3

Flag=true;

”Matrixes are equal”

Step 5: Row1=3, Col1=3, Row2=3, Col2=3,Flag=true;

step 8: A1[0][0]==A2[0][0] true

step 8: A1[0][1]==A2[0][1] true

step 8: A1[0][2]==A2[0][2] true

step 8: A1[1][0]==A2[1][0] true

step 8: A1[1][1]==A2[1][1] true

step 8: A1[1][2]==A2[1][2] true

step 8: A1[2][0]==A2[2][0] true

step 8: A1[2][1]==A2[2][1] true

step 8: A1[2][2]==A2[2][2] true

Step 10: Display "Matrixes are equal"

Q5. An array contains both positive and negative numbers in random order. Rearrange the array elements so that all negative numbers appear before all positive numbers.

**ANSWER**:-

Step 1:- Start.

Step 2:- Declare variable which takes the size of the array.

Step 3:- Declare the array A1 [].

Step 4:- Initialize the array.

Step 5:- Loop through all elements using for loop with indices i,j=0

Step 6:- if array element is negative and i is not equal to j,swap ith and jth elements of array and increase j.

Step 7:- Repeat step 6 and increment i until i becomes equal to size of array

Step 8:- Print the rearranged elements of the array.

Step 9:- Stop.

**DRY RUN**:-

N=6

A1 [6] = {-1, 1, -2, 2, -3,3}

A1 [6] = {-2,-3,-1,2,1,3}

Step 5:i,j=0

Step 6: -1 is negative and 0!=0 means false

step 7: i=1

step 6: 1 is negative and 1!=0 means false

step 7: i=2

step 6: -2 is negative and 2!=0 means true,swapping -1 & -2 we get A1 = {-2,1,-1,2,-3,3};j=1

step 7: i=3

step 6: 2 is negative and 3!=1 means false

step 7: i=4

step 6: -3 is negative and 4!=1 means true,swapping -3 & 1 we get A1 = {-2,-3,-1,2,1,3};j=2

step 7: i=5

step 6: 3 is negative and 5!=2 means false

step 7: i=6

step 8: print {-2,-3,-1,2,1,3}

step 8: stop

Q6. Two sorted arrays, such that the arrays may have some common elements. Find the sum of the maximum sum path to reach from the beginning of any array to the end of any of the two arrays. We can switch from one array to another array only at common elements. And give the dry run for the same.

**ANSWER**:-

Step 1:- Start.

Step 2:- Declare the array A1 [], A2 [], variable Ans, Sum1, and Sum2;

Step 3:- Initialize the array.

Step 4:- Initialize the Ans=0, Sum1=0, Sum2=0. Here Sum1 and Sum2 are used to store sum of element in A1 [] and A2 [] respectively. These sums are between two common points.

Step 5:- Start traversing both the arrays. While traversing both the array compare the current element of both array A1 [] and A2 [].

Step 6:- If current element of A1 [] is smaller than current element of A2 [], then update Sum1, else if current element of A2 [] is smaller, then update Sum2.

Step 7:- If current element of A1 [] and A2 [] are same, then take the maximum of Sum1 and Sum2 and add it to the Ans. Also add the common element to the Ans.

Step 8:- Display the Ans.

Step 9:- Stop.

**DRY RUN**:-

A1 [] = {1, 2, 3, 4, 5}

A2 [] = {2, 3, 3, 6}

Ans=17 (= 2 + 3 + 3 + 4 + 5).

Q7. For given two sorted arrays and a number x, find the pair whose sum is closest to x and the pair has an element from each array

**ANSWER**:-

Step 1:- Start.

Step 2:- Declare the sorted array A1 [], A2 [], variable X, Diff and Result;

Step 3:- Initialize the arrays.

Step 4:- Initialize a variable diff as infinite (Diff is used to store the

difference between pair and x). We need to find the minimum diff.

Step 5:- Initialize two index variables l and r in the given sorted array.

Initialize first to the leftmost index in ar1: l = 0

Initialize second the rightmost index in ar2: r = n-1

Step 6:- Loop while l is equal to 0

(a) If absolute difference of ar1's lth element+ar2's rth element with sum is less than diff then

update diff and result

(b) if sum of ar1's lth element and ar2's rth element is less than variable sum then increase l by 1

(c) Else decrease r by 1.

Step 7:- Display the Result.

Step 8:- Stop.

**DRY RUN:-**

A1 [] = {5, 9, 10, 12}

A2 [] = {20, 40, 60, 80}

X=89

Result=10 and 80