

SET-A

Question 1 :

School XYZ is planning to conduct an Arts and Literary competition this academic year. A students can participate in more than one event. Given the list of students registered for English speech, Hindi speech, and debate; you need to perform the following tasks.

1. List the students registered for at least one of the speech events.
2. List the students registered only for debate.

Design :

main()

1. Read the no of students in English , Hindi and Debate into n1, n2, n3
2. Read the roll numbers into 3 arrays E[0,.....,n1-1], H[0,.....,n2-1], D[0,.....,n3-1]
3. for i = 0 to n1-1 // printing the roll numbers registered in English speech
 - a. print(E[i])
4. for i = 0 to n2-1 // printing the roll numbers registered only in Hindi speech
 - a. if (search (E , H[i] , n1))
 - i. continue
 - b. else
 - i. print(H[i])
5. flag =0 // flag to check if any student enrolled only in Debate
6. for i = 0 to n3-1
 - a. if(search(E , D[i], n1) or search(H , D[i],n2))
 - i. Continue
 - b. else
 - i. print(D[i])
 - ii. flag =1
7. if(flag == 0)
 - a. print(-1)

Evaluation criteria : [4 marks]
--

0.5 marks for line 3

1.5 marks for line 4

1.5 marks for line 6

0.5 marks for line 7

search(A, key, n):

//returns true if roll number is found in array A else returns false

1. for i = 0 to n-1
 - a. if(key == A[i])
 - i. return true
2. return false

Evaluation criteria : **[2 mark]**

Question-2

The event coordinators have decided to conduct the final round of the debate event. For this, they collected the scores of all the students participated in the preliminary round from different judges and consolidated the scores. To select the students for the final round, the coordinators need to sort the entire scores in non-increasing order. It was observed that in the consolidated list, there were sequences of scores in non-increasing order. The coordinators decided to prepare the final sorted list of scores.

Design :

Declare global variables start , end

print_scores(A, i, j):

- a. for k = i to j
 - i. print(A[k]) // add spaces while printing

Evaluation criteria : [0.5 mark]

longest_sorted_sequence(A, n):

//finds the longest leftmost sorted sequence in non increasing order from the array and updates global variables start and end.

1. max_length = 1
2. start = 0 // global variable
3. end = 0 // global variable
4. length = 1
5. index = 0
6. For i=1 to n-1
 - a. if(A[i-1] >= A [i])
 - i. length ++
 - b. else
 - i. if(length > max_length)// to consider the leftmost sequence
 1. max_length = length
 2. start = index
 3. end = start + max_length-1
 - ii. length = 1
 - iii. index = i
7. if(length > max_length)
 - a. max_length = length
 - b. start = index
 - c. end = start + max_length-1

Evaluation criteria : [1 mark]

sort_scores(A, n):

// sort function to sort the scores of the given array in non-increasing order (students are allowed to use any sorting algorithm)

1. score_sort(A, 0, n-1)

score_sort(A, l, r):

// merge sort for sorting a given array

1. if $l < r$
 - a. $q = (l + r) / 2$
 - b. score_sort (A, l, q)
 - c. score_sort (A, q+1, r)
 - d. merge (A, l, q, r)

merge(A, p, q, r):

// merge procedure for merge sort in non increasing order

1. $n1 = q - p + 1$
2. $n2 = r - q$
3. let $L[0.....n1]$ and $R[0.....n2]$ be new arrays
4. for $i = 0$ to $n1 - 1$
 - a. $L[i] = A[p + i]$
5. for $j = 0$ to $n2 - 1$
 - a. $R[j] = A[q + j + 1]$
6. $L[n1] = \infty$
7. $R[n2] = \infty$
8. $i = 0$
9. $j = 0$
10. For $k = p$ to r
 - a. if $L[i] \geq R[j]$
 - i. $A[k] = L[i]$
 - ii. $i = i + 1$

- b. else
 - i. $A[k] = R[j]$
 - ii. $j = j+1$

Evaluation criteria : **[1 marks]**

three_way_merge(A, d1, d2, n):

// merge procedure extension for merging 3 sorted arrays in non increasing order

1. $n1 = d1 + 1$
2. $n2 = d2 - d1$
3. $n3 = n - d2 - 1$
4. let $L[0.....n1]$, $M[0.....n2]$, $R[0.....n3]$ be new arrays
5. for $i = 0$ to $n1 - 1$
 - a. $L[i] = A[i]$
6. for $j = 0$ to $n2 - 1$
 - a. $M[j] = A[d1 + j + 1]$
7. for $k = 0$ to $n3 - 1$
 - a. $R[k] = A[d2 + k + 1]$
8. $L[n1] = \infty$
9. $M[n2] = \infty$
10. $R[n3] = \infty$
11. $i = 0$
12. $j = 0$
13. $k = 0$
14. for $l = 0$ to $n-1$
 - a. if ($R[k] \geq M[j]$)
 - i. If ($R[k] \geq L[i]$)
 1. $A[l] = R[k]$
 2. $k = k + 1$
 3. print (3)

- ii. else
 - 1. $A[l] = L[i]$
 - 2. $i = i + 1$
 - 3. print (1)
- b. else
 - i. if $M[j] \geq L[i]$
 - 1. $A[l] = M[j]$
 - 2. $j = j + 1$
 - 3. print (2)
 - ii. else
 - 1. $A[l] = L[i]$
 - 2. $i = i + 1$
 - 3. print(1)

Evaluation criteria : **[1.5 marks]**

No marks shall be awarded if two way merge is used twice to merge the given three arrays