Technical Training(online)

Started on	Monday, 20 April 2020, 12:18 PM
State	Finished
Completed on	Monday, 20 April 2020, 1:47 PM
Time taken	1 hour 29 mins
Marks	17.90/21.00
Grade	**25.57** out of 30.00 (**85**%)

Correct

Mark 10.00 out of 10.00

Example:

```
Input:
9
-2 1 -3 4 -1 2 1 -5 4
Output: 6
Explanation: [4,-1,2,1] has the largest sum = 6.
```

For example:

Test	Input	Result
1	9 -2 1 -3 4 -1 2 1 -5 4	6

Answer: (penalty regime: 10, 20, ... %)

Language c

```c
#include<stdio.h>
#include<stdlib.h>
int largestSum(int *arr, int n) {
    int i, j, k, largest = 0, sum;
    for(i=0;i<n;i++) {
        if(arr[i] > largest)
            largest = arr[i];
    }
    for(k=2;k<n+1;k++) {
        for(i=0;i<n-k+1;i++)
        {
            sum=0;
            for(j = i; j < i+k; j++)
                sum = sum + arr[j];
            if(sum > largest)
                largest = sum;
        }
    }
    return largest;
}
int main() {
    int n, i, *arr, largest;
    scanf("%d",&n);
    arr = (int *) malloc (n * sizeof(int));
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    largest = largestSum(arr, n);
    printf("%d", largest);
    return 0;
}
```

	Test	Input	Expected	Got	
✔	1	9 -2 1 -3 4 -1 2 1 -5 4	6	6	✔
✔	2	9 1 2 3 4 5 6 1 2 3	27	27	✔
✔	3	6 -10 -2 -3 1 -100 -2	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 10.00/10.00.

Correct

Mark 7.00 out of 10.00

if there're duplicates in `arr`, count them seperately.

Example 1:

```
Input:
3
1 2 3
Output: 2
Explanation: 1 and 2 are counted cause 2 and 3 are in arr.
```

Example 2:

```
Input:
8
1 1 3 3 5 5 7 7
Output: 0
Explanation: No numbers are counted, cause there's no 2, 4, 6, or 8 in arr.
```

Example 3:

```
Input:
6
1 3 2 3 5 0
Output: 3
Explanation: 0, 1 and 2 are counted cause 1, 2 and 3 are in arr.
```

Example 4:

```
Input:
4
1 1 2 2
Output: 2
Explanation: Two 1s are counted cause 2 is in arr.
```

Answer: (penalty regime: 10, 20, ... %)

Language c

```c
#include<stdio.h>
#include<stdlib.h>
int count(int *arr, int n) {
    int cnt = 0, i, j;
    for  (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            if (arr[i]+1 == arr[j]) {
                cnt++;
                break;
            }
        }
    }
    return cnt;
}
int main() {
    int n, i, *arr, cnt;
    scanf("%d",&n);
    arr = (int *) malloc (n * sizeof(int));
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    cnt = count(arr, n);
    printf("%d",cnt);
    return 0;
}
```

	Test	Input	Expected	Got	
✔	1	3 1 2 3	2	2	✔
✔	2	8 1 1 3 3 5 5 7 7	0	0	✔
✔	3	6 1 3 2 3 5 0	3	3	✔
✔	4	4 1 1 2 2	2	2	✔

Correct

Marks for this submission: 10.00/10.00. Accounting for previous tries, this gives **7.00/10.00**.

Question **3**

Correct

Mark 0.90 out of 1.00

Given an array `nums`, write a function to move all `0`'s to the end of it while maintaining the relative order of the non-zero elements.

Example:

```
Input:
5
0 1 0 3 12
Output: 1 3 12 0 0
```

For example:

Test	Input	Result
1	5 0 1 0 3 12	1 3 12 0 0

Answer: (penalty regime: 10, 20, ... %)

Language c

```c
#include<stdio.h>
#include<stdlib.h>
int* move0s(int *arr, int n) {
    int cnt = 0, i, *arr1;
    arr1 = (int *) malloc (n * sizeof(int));
    for  (i = 0; i < n; i++) {
        if (arr[i]!=0){
            arr1[cnt++] = arr[i];
        }
    }
    while (cnt < n) {
        arr1[cnt++] = 0;
    }
    return arr1;
}
int main() {
    int n, i, *arr;
    scanf("%d",&n);
    arr = (int *) malloc (n * sizeof(int));
    for (i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    arr = move0s(arr, n);
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

	Test	Input	Expected	Got	
✔	1	5 0 1 0 3 12	1 3 12 0 0	1 3 12 0 0	✔
✔	2	4 1 2 3 4	1 2 3 4	1 2 3 4	✔
✔	3	6 1 0 2 0 3 0	1 2 3 0 0 0	1 2 3 0 0 0	✔
✔	4	4 0 0 0 0	0 0 0 0	0 0 0 0	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.90/1.00**.

Jump to...

SECTIONS

1

2

3

Follow Us

Powered by Technical Hub