Prepare > C > Functions > Sorting Array of Strings

Exit Full Screen View

Submissions

Leaderboard

Discussions

Editorial

Constraints

- $1 \leq$ No. of Strings ≤ 50
- $1 \leq$ Total Length of all the strings ≤ 2500
- You have to write your own sorting function and you cannot use the inbuilt *qsort* function
- The strings consists of lower-case English Alphabets only.

Output Format

The locked code-stub will check the logic of your code. The output consists of the strings sorted according to the four comparsion functions in the order mentioned in the problem statement.

Sample Input 0

```
4
wkue
qoi
sbv
fekls
```

Sample Output 0

```
fekls
qoi
sbv
wkue
```

Change Theme Language: C

```

    char** arr;
    arr = (char**)malloc(n * sizeof(char*));

    for(int i = 0; i < n; i++){
        *(arr + i) = malloc(1024 * sizeof(char));
        scanf("%s", *(arr + i));
        *(arr + i) = realloc(*(arr + i), strlen(*(
    }

    string_sort(arr, n, lexicographic_sort);
    for(int i = 0; i < n; i++)
        printf("%s\n", arr[i]);
    printf("\n");

    string_sort(arr, n, lexicographic_sort_reverse
    for(int i = 0; i < n; i++)
        printf("%s\n", arr[i]);
    printf("\n");

    string_sort(arr, n, sort_by_length);
    for(int i = 0; i < n; i++)
        printf("%s\n", arr[i]);
    printf("\n");

    string_sort(arr, n, sort_by_number_of_distinct_
    for(int i = 0; i < n; i++)
        printf("%s\n", arr[i]);
    printf("\n");
}
```