■ HackerRank | Prepare   Certify   Compete   Apply   Q Search   ▢ 🔔² Ⓗ ⌄

All Contests > CSE-g-week2 > Printing Symbols

Printing Symbols

Problem	Submissions	Leaderboard	Discussions

Structure symbol has a character, string and an integer field. Dynamically allocate an array of 'n' structure symbol elements. Accept the character, its name and number of dots for the 'n' elements in the order given in input format. Pass the structure array to the sort function. Sort the array in alphabetical order for the character names. Return the sorted array. Display each character of the sorted array, after printing the dots(periods or full stop) as per the given number. Free the dynamically allocated memory.

In case of the same symbol names, print in the same order of entry as in input. Note that each set of periods (dots) and symbol should be followed by a newline character, even the last character

Input Format

number of elements
number_of_dots symbol symbol_name

Constraints

Number of elements<100

Output Format

number_of_dots times dot(.) symbol

Sample Input 0

```
3
6 # hash
9 @ at
4 ! exclamation
```

Sample Output 0

```
.........@
....!
......#
```

Explanation 0

According to alphabetical order at comes first and it has 9 dots followed by the symbol @
Next comes exclamation with 4 dots followed by !
Next hash with 6 dots followed by #

Sample Input 1

```
3
6 # hash
6 @ at
2 # hash
```

Sample Output 1

```
......@
......#
```

..#

Explanation 1

Alphebetically at comes first followed by hash. Here since there are two hashes the order that is manitained is the same as in input.

f y in

Contest ends in a day

Submissions: 14
Max Score: 6
Difficulty: Medium

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

C

```c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  typedef struct sym {
6      char c;
7      char name[50];
8      int dot;
9  } SYMBOL;
10
11 SYMBOL *sort(SYMBOL *arr, int n);
12 void display(SYMBOL *arr, int n);
13 void swap(SYMBOL *a, SYMBOL *b);
14
15 int main() {
16     int n;
17     scanf("%d", &n);
18
19     SYMBOL *arr=(SYMBOL*)malloc(n*sizeof(SYMBOL));
20
21     for(int i=0;i<n;i++){
22         scanf("%d",&arr[i].dot);
23         scanf(" %c",&arr[i].c);
24         scanf("%s",arr[i].name);
25     }
26
27
28     sort(arr,n);
29     display(arr,n);
30
31     free(arr);
32
33     return 0;
34 }
35
36 SYMBOL *sort(SYMBOL *arr, int n) {
37
38     for(int i=0; i<n;i++){
39         for(int j=i;j<n;j++){
40             if(strcmp(arr[j].name,arr[i].name)<0){
41                 SYMBOL temp=arr[i];
42                 arr[i]=arr[j];
43                 arr[j]=temp;
44             }
45         }
46     }
47     return arr;
48 }
49
50 void display(SYMBOL *arr, int n) {
51     for(int i=0;i<n;i++){
```

```
52  ▾        for(int j=0;j<arr[i].dot;j++){
53               printf("%c",'.');
54           }
55  ▾        printf("%c\n",arr[i].c);
56       }
57  }
```

Line: 1 Col: 1

⬆ Upload Code as File   ☐ Test against custom input   Run Code   Submit Code

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |