CODECHEF
An Educational Initiative

Hello **pushpitha2001** ▾ ▤ 8⁺ Ⓞ f Logout

▸ PRACTICE ▸ COMPETE ▸ DISCUSS ▸ COMMUNITY ▸ HELP ▸ ABOUT

Home » Pushpitha P

Pushpitha P ✏ f 👥

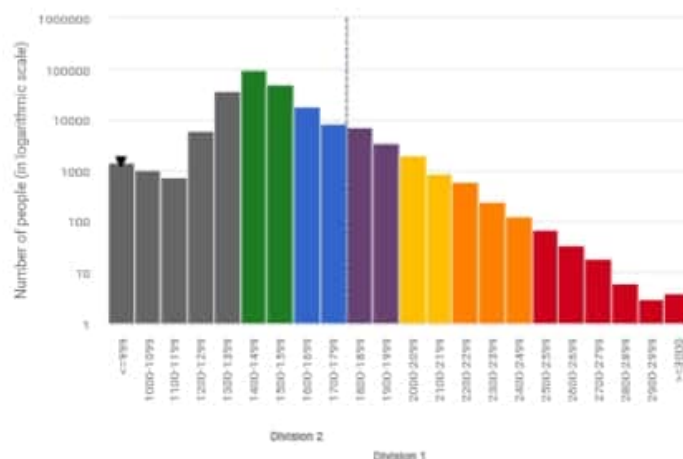| Username:	pushpitha2001
Country:	🇮🇳 India
State:	Karnataka
City:	Mysuru
Student/Professional:	Student
Institution:	Alvas Institute of Engineering and Technology Karnataka, India
Teams List:	List of teams by Pushpitha P
Team Invites:	Click here to check team invites. 🔄

Rating Graphs

CodeChef	Long Challenge	Cook Off	Lunch Time

CodeChef Rating Distribution

0
★

CodeChef Rating
(Highest Rating 0)

NA | **NA**
Global Rank | Country Rank

Contests	Rating	Global Rank	Country Rank
Long Challenge	0	NA	NA
Cook-off	0	NA	NA
Lunch Time	0	NA	NA

Recent Activity

Date/Time	Problem	Result	Lang
	No Recent Activity		

CODECHEF
An Educational Initiative

pushpitha2001@gr ············   Login   New User
Forgot Password

▸ PRACTICE ▸ COMPETE ▸ DISCUSS | ▸ COMMUNITY ▸ HELP ▸ ABOUT

Home ▸ IDE

Code, Compile & Run

Ide   ✕   +

C++14 (gcc 6.3)   ▼   ⓘ ⬇ ↗ ⚙

```
1   #include <stdio.h>
2   void distinct_element(int a[],int n);
3   int main()
4 ▾ {
5       int size_array,i,arr[20];
6       scanf("%d",&size_array);
7       for(i=0;i<size_array;i++)
8 ▾     {
9           scanf("%d",&arr[i]);
10      }
11      distinct_element(arr,size_array);
12      return 0;
13  }
14  void distinct_element(int a[],int n)
15 ▾ {
16      int i,j;
17      for(i=0;i<n;i++)
18 ▾     {
19          for(j=0;j<i;j++)
20 ▾         {
21              if(a[i]==a[j])
22              break;
23          }
24          if(i==j)
25          printf("%d",a[i]);
26      }
27  }
```

24:26   ⟳

Open File   ✓ Custom Input   Run

Custom Input
```
5
1 2 3 4 4
```

Status Successfully executed   Date 2020-06-07 06:47:33   Time 0 sec   Mem 15.232 kB   ✕

Input
```
5
1 2 3 4 4
```

Output
```
1234
```

Algorithm of a program to execute distinct count.

Step 1 :- Start

Step 2 :- For every index i from 0 to len _ array (n) - k, i.e n-k, traverse the array from i to i+k. This is the window.

Step 3. Traverse the window, from i to that index and check. if the element is present or not.

Step 4. If the element is not present in the prefix of the array i.e no duplicate element is present from i to index -1, then increase the count.

Step 5 :- Print the count

Step 6 :- Stop.

Flowchart

```
( Start )
   |
   v
( int main() )
   |
   v
+------------------------------+
|  int n, k;                   |
|  int ctr = 0;                |
|  int i;                      |
|  int arr1[] = {5,2,3         |
|     7,6,4,9,8};              |
|  n = size of (arr1)/size     |
|     of (arr1[0]);            |
|           k = 5              |
+------------------------------+
   |
   v
+------------------------------+
| printf(" The given array     |
|     is : \n"); i=0           |
+------------------------------+
   |
   v
  < i < n? >-------------------- True
   | False                        |
   v                              v
+------------------------------+  +-----------------------------+
| printf("\n")                 |  | printf("%d ", arr1[i]);     |
| qsort(arr1, n, size of(int), |  +-----------------------------+
|   compfunc);                 |              |
|  int l = 0;                  |              v
|  int r = 0;                  |          +--------+
| printf("The distinct pairs   |          |  i++   |
|  for difference %od are:",k); |         +--------+
+------------------------------+
   |
   v
  < r < n >
   | False
   v
+------------------------------+
| printf("\n number of distinct |
| pairs for difference %d are   |
| %d", k, ctr return 0;        |
+------------------------------+
   |
   v
 [ End ]
```

printf("[%d,%d]"
arr1[r], arr1[l]);
ctr++;
l++;
r++;

  < arr1[r] - arr1[l] == k ? > ---- True ---- ... ---- False
            | True |
            v   v
        ...   < arr1[r]-arr1[l] > k ? >
                                                         | False
                                                         v
                                                     [r++;] False

[i++;] <---- True