

CSES Problem Set**Distinct Numbers**[TASK](#) | [SUBMIT](#) | [RESULTS](#) | [ANALYSIS](#) | [STATISTICS](#) | [TESTS](#) | [QUEUE](#)**Submission details**

Task: [Distinct Numbers](#)
 Sender: josueMamani
 Submission time: 2025-10-08 02:57:59 +0300
 Language: C++ (C++17)
 Status: READY
 Result: ACCEPTED

Test results ▲

test	verdict	time	
#1	ACCEPTED	0.00 s	»
#2	ACCEPTED	0.00 s	»
#3	ACCEPTED	0.00 s	»
#4	ACCEPTED	0.04 s	»
#5	ACCEPTED	0.09 s	»
#6	ACCEPTED	0.11 s	»
#7	ACCEPTED	0.10 s	»
#8	ACCEPTED	0.10 s	»
#9	ACCEPTED	0.00 s	»
#10	ACCEPTED	0.10 s	»
#11	ACCEPTED	0.08 s	»
#12	ACCEPTED	0.06 s	»
#13	ACCEPTED	0.03 s	»
#14	ACCEPTED	0.03 s	»
#15	ACCEPTED	0.10 s	»

Compiler report ▲

```
input/code.cpp: In function 'int main()':
input/code.cpp:29:19: warning: 'sum' may be used
uninitialized in this function [-Wmaybe-
uninitialized]
29 |         cout<<sum-1;
|             ^

```

Code ▲

```
1 #include<iostream>
2 #include<vector>
```

Sorting and SearchingDistinct Numbers Apartments Ferris Wheel Concert Tickets Restaurant Customers Movie Festival Sum of Two Values Maximum Subarray Sum

...

Your submissions2025-10-08 02:57:59 2025-10-08 02:57:13

```

3 #include<algorithm>
4 using namespace std;
5
6 int main(){
7     int numero, sum;
8
9     cin>> numero;
10    //definimos al vector
11    vector <int> vec;
12
13    //agregamos datos
14    for(int i=0; i<numero;i++){
15        int valor;
16        cin>>valor;
17        vec.push_back(valor);
18    }
19
20    //ordenamos los elementos del vector
21    sort(vec.begin(), vec.end());
22
23    for(int j=0; j<numero; j++){
24        if(vec[j]!=vec[j+1]){
25            sum=sum+1;
26        }
27    }
28
29    cout<<sum-1;
30
31
32    return 0;
33 }
```

SHARE CODE TO OTHERS 

Test details ▲

Test 1

Verdict: ACCEPTED

input
10 1 1 1 1 1 1 1 1 1 1

correct output
1

user output
1

Test 2

Verdict: ACCEPTED

input
10 7 4 10 9 6 1 8 2 5 3

correct output
10

user output
10

Test 3

Verdict: ACCEPTED

input
10 5 9 5 5 10 9 3 1 8 8

correct output
6

user output
6

Test 4

Verdict: ACCEPTED

input
200000 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...

correct output
1

user output
1

Test 5

Verdict: ACCEPTED

input

200000

150502 195680 66111 105526 183...

**correct output**

200000

**user output**

200000

**Test 6**

Verdict: ACCEPTED

input

200000

265840313 559159566 749760055 ...

**correct output**

199973

**user output**

199973

**Test 7**

Verdict: ACCEPTED

input

200000

1 107898 215795 323692 431589 ...

**correct output**

200000

**user output**

200000

**Test 8**

Verdict: ACCEPTED

input

200000

1 300001 600001 900001 1200001...



correct output		
200000		

user output		
200000		

Test 9

Verdict: ACCEPTED

input		
4 1 2 8 9		

correct output		
4		

user output		
4		

Test 10

Verdict: ACCEPTED

input		
200000 199999999 107898 215795 323969...		

correct output		
200000		

user output		
200000		

Test 11

Verdict: ACCEPTED

input		
199999 199996 199997 149999 117797 19...		

correct output		
-----------------------	--	--

199999

**user output**

199999

**Test 12**

Verdict: ACCEPTED

input

107896

107897 215794 323691 431588 53...

**correct output**

107896

**user output**

107896

**Test 13**

Verdict: ACCEPTED

input

100000

65537 6 31 156 781 3906 19531 ...

**correct output**

32770

**user output**

32770

**Test 14**

Verdict: ACCEPTED

input

100000

65537 2 3 4 5 6 7 8 9 10 6 7 8...

**correct output**

19012



user output	
19012	

Test 15

Verdict: ACCEPTED

input	
200000 1 172934 345867 518800 691733 ...	

correct output	
200000	

user output	
200000	