

Practice > Algorithms > Warmup > Birthday Cake Candles

Birthday Cake Candles ☆

ProblemSubmissionsLeaderboardDiscussionsEditorial

You are in-charge of the cake for your niece's birthday and have decided the cake will have one candle for each year of her total age. When she blows out the candles, she'll only be able to blow out the tallest ones. Your task is to find out how many candles she can successfully blow out.

For example, if your niece is turning **4** years old, and the cake will have **4** candles of height **3, 2, 1, 3**, she will be able to blow out **2** candles successfully, since the tallest candle is of height **3** and there are **2** such candles.

Complete the function `birthdayCakeCandles` that takes your niece's age and an integer array containing height of each candle as input, and return the number of candles she can successfully blow out.

Input Format

The first line contains a single integer, *n*, denoting the number of candles on the cake.
The second line contains *n* space-separated integers, where each integer *i* describes the height of candle *i*.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq height_i \leq 10^7$

Output Format

Print the number of candles that can be blown out on a new line.

Sample Input 0

```
4
3 2 1 3
```

Sample Output 0




```
2
```

Explanation 0

We have one candle of height **1**, one candle of height **2**, and two candles of height **3**. Your niece only blows out the tallest candles, meaning the candles where *height* = **3**. Because there are **2** such candles, we print **2** on a new line.

Authorshashank21j
DifficultyEasy
Max Score10
Submitted By223636




NEED HELP?

-  [View discussions](#)
-  [View editorial](#)
-  [View top submissions](#)






RATE THIS CHALLENGE

☆☆☆☆

MORE DETAILS

-  [Download problem statement](#)
-  [Download sample test cases](#)
-  [Suggest Edits](#)



Current Buffer (saved locally, editable)   C++   

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 vector<string> split_string(string);
```

```
6
7 // Complete the birthdayCakeCandles function below.
8 int birthdayCakeCandles(vector<int> ar) {
9
10
11 }
12
13 int main()
14 {
15     ofstream fout(getenv("OUTPUT_PATH"));
16
17     int ar_count;
18     cin >> ar_count;
19     cin.ignore(numeric_limits<streamsize>::max(), '\n');
20
21     string ar_temp_temp;
22     getline(cin, ar_temp_temp);
23
24     vector<string> ar_temp = split_string(ar_temp_temp);
25
26     vector<int> ar(ar_count);
27
28     for (int i = 0; i < ar_count; i++) {
29         int ar_item = stoi(ar_temp[i]);
30
31         ar[i] = ar_item;
32     }
33
34     int result = birthdayCakeCandles(ar);
35
36     fout << result << "\n";
37
38     fout.close();
39
40     return 0;
41 }
42
43 vector<string> split_string(string input_string) {
44     string::iterator new_end = unique(input_string.begin(),
45     input_string.end(), [] (const char &x, const char &y) {
46         return x == y and x == ' ';
47     });
48     input_string.erase(new_end, input_string.end());
49
50     while (input_string[input_string.length() - 1] == ' ') {
51         input_string.pop_back();
52     }
53
54     vector<string> splits;
55     char delimiter = ' ';
56
57     size_t i = 0;
58     size_t pos = input_string.find(delimiter);
59
60     while (pos != string::npos) {
61         splits.push_back(input_string.substr(i, pos - i));
62
63         i = pos + 1;
64         pos = input_string.find(delimiter, i);
65     }
66
67     splits.push_back(input_string.substr(i, min(pos,
68     input_string.length()) - i + 1));
69
70     return splits;
71 }
```

Line: 1 Col: 1

 Upload Code as File ☐ Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)