Practice > Algorithms > Warmup > Simple Array Sum

Simple Array Sum ☆

Problem Submissions Leaderboard Discussions Editorial

Given an array of integers, find the sum of its elements.

Input Format

The first line contains an integer, n, denoting the size of the array.

The second line contains n space-separated integers representing the array's elements.

Output Format

Print the sum of the array's elements as a single integer.

Sample Input

```
6
1 2 3 4 10 11
```

Sample Output

```
31
```

Explanation

We print the sum of the array's elements: $1 + 2 + 3 + 4 + 10 + 11 = 31$.

Author shashank21j
Difficulty Easy
Max Score 10
Submitted By 764326

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++

```cpp
#include <bits/stdc++.h>

using namespace std;

vector<string> split_string(string);

/*
 * Complete the simpleArraySum function below.
 */
int simpleArraySum(vector<int> ar) {
    /*
     * Write your code here.
     */

}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    int ar_count;
    cin >> ar_count;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    string ar_temp_temp;
    getline(cin, ar_temp_temp);
```

```
27
28      vector<string> ar_temp = split_string(ar_temp_temp);
29
30      vector<int> ar(ar_count);
31
32      for (int ar_itr = 0; ar_itr < ar_count; ar_itr++) {
33          int ar_item = stoi(ar_temp[ar_itr]);
34
35          ar[ar_itr] = ar_item;
36      }
37
38      int result = simpleArraySum(ar);
39
40      fout << result << "\n";
41
42      fout.close();
43
44      return 0;
45  }
46
47  vector<string> split_string(string input_string) {
48      string::iterator new_end = unique(input_string.begin(),
    input_string.end(), [] (const char &x, const char &y) {
49          return x == y and x == ' ';
50      });
51
52      input_string.erase(new_end, input_string.end());
53
54      while (input_string[input_string.length() - 1] == ' ') {
55          input_string.pop_back();
56      }
57
58      vector<string> splits;
59      char delimiter = ' ';
60
61      size_t i = 0;
62      size_t pos = input_string.find(delimiter);
63
64      while (pos != string::npos) {
65          splits.push_back(input_string.substr(i, pos - i));
66
67          i = pos + 1;
68          pos = input_string.find(delimiter, i);
69      }
70
71      splits.push_back(input_string.substr(i, min(pos,
    input_string.length()) - i + 1));
72
73      return splits;
74  }
75
```

Line: 1 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code