Simple Array Sum

Editorial by shashank21j

In this problem you have to read N numbers and sum them. The first line of input is the number of elements. That is read in and converted to an integer if necessary. A value is created to hold the sum. Next, a `for` loop uses that value to terminate reading in the array elements, one integer per line. As lines are read, the sum is accumulated, and finally printed when the loop exits.

Note that some languages, such as C, allow you to specify the data type as it is being read. Others, such as Python, read strings by default and the values need to be converted to their necessary types.

Featured Solutions

C

```
#include <stdio.h>

int main(){
    int number_of_elements;
    int i;
    scanf("%d", &number_of_elements);
    int array[number_of_elements];
    int sum_of_array = 0;

    for(i = 0; i < number_of_elements; i++){
       scanf("%d", &array[i]);
       sum_of_array += array[i];
    }

    printf("%d\n", sum_of_array);
    return 0;
}
```

C++

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    int number_of_elements;
    cin >> number_of_elements;
    vector <int> array(number_of_elements);
    int sum_of_array = 0;

    for(int i = 0; i < number_of_elements; i++){
       cin >> array[i];
       sum_of_array += array[i];
    }

    cout << sum_of_array;
    return 0;
}
```

Java

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int numberOfElements = in.nextInt();
        int array[] = new int[numberOfElements];
        int sumOfArray = 0;
```

STATISTICS

Difficulty:	Easy
Time Complexity:	
Required Knowledge:	for loops
Publish Date:	Jun 20 2016

This is a Practice Challenge

NEED HELP?

View discussions

View top submissions

```
        for(int i = 0; i < numberOfElements; i++) {
            array[i] = in.nextInt();
            sumOfArray += array[i];
        }
        System.out.println(sumOfArray);
        in.close();
    }
}
```

Javascript

```
process.stdin.resume();
process.stdin.setEncoding('ascii');

var input_stdin = "";
var input_stdin_array = "";
var input_currentline = 0;

process.stdin.on('data', function (data) {
    input_stdin += data;
});

process.stdin.on('end', function () {
    input_stdin_array = input_stdin.split("\n");
    main();
});

function readLine() {
    return input_stdin_array[input_currentline++];
}

/////////////// ignore above this line ////////////////////

function main() {
    var number_of_elements = parseInt(readLine());
    array = readLine().split(' ');
    array = array.map(Number);
    var sum_of_array = 0;
    for(var i = 0 ; i < number_of_elements ; i++){
        sum_of_array += array[i];
    }
    console.log(sum_of_array);
}
```

PHP

```
<?php
$handle = fopen ("php://stdin","r");
fscanf($handle, "%d", $number_of_elements);
$array = explode(" ", fgets($handle));
array_walk($array, 'intval');
echo array_sum($array);
?>
```

Python 2

```
#
# map(function, iterable) returns an array of results where each element
# is the result of applying 'function' to an element in the iterable.  In this case
# the iterable is created when the raw_input() is split().  Each element is converted to
# an int.
#
number_of_elements = int(raw_input())
array = map(int, raw_input().split())
print sum(array)
```

Ruby

```
number_of_elements = gets.strip.to_i
array = gets.strip.split(' ').map(&:to_i)
print array.inject(0, :+)
```

Feedback

Was this editorial helpful?

Yes No