

# Sum As You Go Hackerrank Solution



Sum as You Go is a popular problem that appears on platforms like HackerRank, designed to test a programmer's ability to manage and manipulate data structures effectively, particularly when working with cumulative sums. This problem challenges candidates to calculate the cumulative sum of a series of numbers while efficiently handling input and output operations. In this article, we will explore the Sum as You Go problem, discussing its requirements, providing a solution, and examining various approaches to optimize performance.

## Understanding the Problem Statement

The Sum as You Go problem typically requires the following:

1. Input Specification: You will receive a series of integers. The first integer indicates how many subsequent integers there will be.
2. Output Specification: For each integer provided, output the cumulative sum up to that point.

For instance, given the input:

```
5
1
2
3
4
5
...
```

The expected output would be:

```
1
3
6
10
15
```

```
```
```

Here's how the cumulative sums are calculated:

- For the first integer (1), the cumulative sum is 1.
- For the second integer (2), the cumulative sum is  $1 + 2 = 3$ .
- For the third integer (3), the cumulative sum is  $1 + 2 + 3 = 6$ , and so on.

## Breaking Down the Solution

To efficiently solve the Sum as You Go problem, we need to consider the following steps:

### 1. Input Handling

The first step is to read the input data. This can be done using standard input methods in various programming languages. Here's an example in Python:

```
```python
import sys

input_data = sys.stdin.read().splitlines()
```
```

In this example, we read all input lines at once, which can be efficient when dealing with multiple lines of input.

### 2. Initialize Variables

Before processing the input, we need to initialize a variable to keep track of the cumulative sum:

```
```python
cumulative_sum = 0
```
```

### 3. Processing the Input

Next, we iterate over the input data, converting the strings to integers and updating the cumulative sum accordingly. For each integer, we will print the current cumulative sum:

```
```python
for i in range(1, len(input_data)):
    number = int(input_data[i])
    cumulative_sum += number
    print(cumulative_sum)
```
```

This loop starts from the second line of input (index 1) because the first line indicates the number of integers.

## 4. Output the Results

As we calculate the cumulative sum, we print the result immediately. However, in some cases, it might be more efficient to store results in a list and output them at once:

```
```python
results = []
for i in range(1, len(input_data)):
    number = int(input_data[i])
    cumulative_sum += number
    results.append(cumulative_sum)

print("\n".join(map(str, results)))
```
```

This approach minimizes the number of print operations, which can be beneficial in terms of performance when dealing with large input sizes.

## Complexity Analysis

When analyzing the complexity of the Sum as You Go solution, we need to consider both time and space complexities:

### Time Complexity

- The time complexity of the solution is  $O(n)$ , where  $n$  is the number of integers provided (excluding the first integer that indicates the count). This is because we process each integer exactly once.

### Space Complexity

- The space complexity can be  $O(1)$  if we print the results immediately. However, if we store the results in a list before printing, the space complexity becomes  $O(n)$ , as we need to store the cumulative sums.

## Implementation in Different Languages

While the above example is written in Python, the Sum as You Go problem can be implemented in various programming languages. Here are examples in a few popular languages:

## Java

```
```java
import java.util.Scanner;

public class SumAsYouGo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int cumulativeSum = 0;

        for (int i = 0; i < n; i++) {
            cumulativeSum += scanner.nextInt();
            System.out.println(cumulativeSum);
        }
        scanner.close();
    }
}
```
```

## C++

```
```cpp
include
using namespace std;

int main() {
    int n;
    cin >> n;
    int cumulative_sum = 0;

    for (int i = 0; i < n; i++) {
        int number;
        cin >> number;
        cumulative_sum += number;
        cout << }

    return 0;
}
```
```

## JavaScript

```
```javascript
process.stdin.resume();
process.stdin.setEncoding('utf8');
```

```
let input = '';
process.stdin.on('data', chunk => {
  input += chunk;
});

process.stdin.on('end', () => {
  const lines = input.trim().split('\n');
  const n = parseInt(lines[0]);
  let cumulativeSum = 0;

  for (let i = 1; i <= n; i++) {
    const number = parseInt(lines[i]);
    cumulativeSum += number;
    console.log(cumulativeSum);
  }
});
`;
```

## Common Mistakes to Avoid

When solving the Sum as You Go problem, there are some common pitfalls that programmers should be aware of:

1. Incorrect Input Parsing: Ensure that you read the input correctly, especially when using buffered input methods.
2. Off-by-One Errors: Be careful with loop boundaries. Start your loop from the correct index based on how the input is structured.
3. Not Handling Edge Cases: Consider edge cases such as empty input or negative numbers. While the problem typically specifies positive integers, it's good practice to handle exceptions gracefully.
4. Performance Issues with Print Statements: Printing in loops can slow down the program. Batch printing or storing results can improve performance.

## Conclusion

In summary, the Sum as You Go problem is an excellent exercise for developing skills in cumulative calculations and input/output handling. By understanding the problem requirements and following a structured approach, programmers can efficiently solve this task using various programming languages. The concepts explored here not only apply to this specific problem but also serve as foundational skills for tackling similar challenges in competitive programming and technical interviews. With practice, programmers can enhance their problem-solving abilities and optimize their coding techniques.

## Frequently Asked Questions

## **What is the 'Sum as You Go' problem on HackerRank?**

The 'Sum as You Go' problem requires you to calculate the cumulative sum of a sequence of integers and output the sum after each addition.

## **What programming languages can I use to solve the 'Sum as You Go' problem on HackerRank?**

You can use a variety of programming languages such as Python, Java, C++, JavaScript, and Ruby to solve the 'Sum as You Go' problem on HackerRank.

## **What is the expected input format for the 'Sum as You Go' problem?**

The expected input format typically includes an integer 'n' followed by 'n' integers, representing the numbers you need to sum.

## **How do you handle input and output in the 'Sum as You Go' solution?**

You can use standard input functions to read the numbers and print the cumulative sum after processing each number.

## **What is a simple approach to implement the 'Sum as You Go' solution?**

A simple approach is to initialize a variable for the cumulative sum, iterate through the list of integers, update the cumulative sum, and print it in each iteration.

## **Are there any edge cases to consider while solving 'Sum as You Go'?**

Yes, consider edge cases such as an empty input list, negative integers, and very large integers that could affect the cumulative sum.

## **Can you provide a sample code snippet for the 'Sum as You Go' problem?**

Sure! Here's a simple example in Python:

```
```python
n = int(input())
sum = 0
for _ in range(n):
    num = int(input())
    sum += num
print(sum)
```
```

## What are common mistakes to avoid when solving 'Sum as You Go'?

Common mistakes include not resetting the cumulative sum, mismanaging input/output operations, and not accounting for the size limits of the data.

## How can I optimize my solution for the 'Sum as You Go' problem?

The problem is generally straightforward, but ensure you're using efficient input methods for large datasets, such as reading all input at once and then processing it.

## Where can I find discussions or solutions for 'Sum as You Go' on HackerRank?

You can find discussions and solutions in the HackerRank community forums, Stack Overflow, or by searching for tutorials related to the problem.

Find other PDF article:

<https://soc.up.edu.ph/45-file/Book?docid=XO155-7873&title=past-and-present-tense-verbs-worksheet-s.pdf>

## Sum As You Go Hackerrank Solution

**excel** 0 \_

Jul 10, 2024 · SUM SUM 5. Excel ...

**excel** -

Sep 26, 2016 · =sum (D4:D13) D4:D13

**in sum** to sum up \_

Aug 12, 2024 · "in sum" "to sum up" "in sum" ...

**SUM** + **IF** -

Jul 1, 2020 · SUM + IF SUM IF

**excel** sum -

Feb 15, 2017 · SUM SUM SUM SUM (number1, ...

**SUBTOTAL** (9,F2:F12) 9 \_

Apr 2, 2015 · SUBTOTAL (9,F2:F12) 9 SUBTOTAL 1-11 9 SUM SUBTOTAL ...

Excel SUBTOTAL 0 - 0

Aug 15, 2019 · sum D1 =SUM (D3:D32) 1929 ...

SUM -WPS

May 10, 2020 · SUM =SUM (A2:A10)=SUM (A2:A10,C2:C10) ...

excel sum -

Nov 26, 2018 · SUM B1:G1 B1 G1

Excel -

Jun 17, 2019 · =SUM A1 B1

excel 0 \_

Jul 10, 2024 · SUM SUM 5 Excel 0

excel -

Sep 26, 2016 · =sum (D4:D13) D4:D13

**in sum** **to sum up** \_

Aug 12, 2024 · "in sum" "to sum up" "in sum" "It's a great idea, in sum" "I need a good job, in sum, to challenge myself" ...

**SUM** **+IF** -

Jul 1, 2020 · SUM +IF SUM IF

**excel** **sum** -

Feb 15, 2017 · SUM SUM SUM SUM SUM number1, [number2],... number1 1 number2,... ( ) ...

**SUBTOTAL (9,F2:F12)** 9 \_

Apr 2, 2015 · SUBTOTAL (9,F2:F12) 9 SUBTOTAL 1-11 9 SUM SUBTOTAL SUM 1 sum 2 subtotal SUM =SUM (A1:A10) A1 ...

Excel SUBTOTAL 0 - 0

Aug 15, 2019 · sum D1 =SUM (D3:D32) 1929

SUM -WPS

May 10, 2020 · SUM =SUM (A2:A10)=SUM (A2:A10,C2:C10) SUM (number1, [number2...,WPS WPS 1000+Office WPS (Excel) WPS Word WPS (PPT) ...

excel sum -

Nov 26, 2018 · SUM B1:G1 B1 G1



**Excel** 100 Days of Excel - Day 10

Jun 17, 2019 · `=SUM` 100 Days of Excel - Day 10  
100 Days of Excel - Day 10

Unlock the 'sum as you go' HackerRank solution with our comprehensive guide. Enhance your coding skills and tackle challenges effectively. Learn more!

[Back to Home](#)