Solution: Add (P1) — Sum of Two Integers

ANNAFORCES

Problem Description

Read two integers a and b, and print their sum $a + b$.

Simple Answer

Read the two numbers and output their sum. This is a direct implementation of integer addition.

Detailed Explanation

Approach

The operation requires three logical steps:

1. **Input Acquisition:** Read integers a and b from standard input.
2. **Computation:** Compute $s = a + b$ using the language's integer addition operator.
3. **Output Presentation:** Print s to standard output followed by a newline.

This is an $O(1)$ time and $O(1)$ additional space algorithm (only constant extra memory is used).

Constraints and Data Types

Constraints: $|a|, |b| < 10^{17}$. The sum may be as large as $2 \times (10^{17} - 1)$ in magnitude, which fits within the signed 64-bit integer range (`long long` in C/C++). Python's native integers are arbitrary-precision and are safe as well.

Complexity

- **Time Complexity:** $O(1)$ — only a fixed number of operations are performed.
- **Space Complexity:** $O(1)$ — only a few variables are used.

Language-Specific Implementations

Below are sample implementations in Python, C++, and C. Each program reads two integers from standard input and prints their sum.

Python (solution.py)

```python
# solution.py
# Reads two integers and prints their sum

a, b = map(int, input().split())
print(a + b)
```

C++ (solution.cpp)

```cpp
// solution.cpp
// Uses long long to safely handle values up to 10^17
#include <iostream>

int main() {
    long long a, b;
    std::cin >> a >> b;
    std::cout << (a + b) << std::endl;
    return 0;
}
```

C (solution.c)

```c
/* solution.c */
#include <stdio.h>

int main() {
    long long a, b;
    if (scanf("%lld %lld", &a, &b) == 2) {
        printf("%lld\n", a + b);
    }
    return 0;
}
```

Notes and Verification

The sample cases are simple arithmetic checks:

- Input: `1 1` -> Output: `2`
- Input: `2 2` -> Output: `4`

These follow directly from the definition of addition.