Assignment 7: Missing Card

7.1 Algorithm

An algorithm is a finite step of execution to find the answer to a problem. There can be several algorithms for the same problem. The following code shows two ways to calculate the sum of natural numbers up to 10:

```c
#include <stdio.h>

int sum1(int n) {
    int s = 0;
    for (int i = 1; i <= n; i++)
        s += i;
    return s;
}

int sum2(int n) {
    return n * (n + 1) / 2;
}

int main() {
    int n = 10;
    printf("1 + ... + %d = %d\n", n, sum1(n));
    printf("1 + ... + %d = %d\n", n, sum2(n));
}
```

7.2 Programming Assignment 7: missing.c

You have n cards ($n > 0$). The cards are numbered by 1, 2, ..., n. But, one of the cards is missing from the set. Enumerating the remaining card, you have a sequence of numbers. Given a sequence of numbers, find the number on the missing card.

Having ten cards initially, if the enumeration of the remaining cards makes the following sequence of numbers:

> 7, 4, 2, 1, 9, 3, 6, 8, and 10,

the number on the missing card is 5. If the given sequence of numbers is:

> 1, 3, 2, 5, 4, and 6,

there is no missing card from 1 to 6. Therefore, the missing card has the number 7.

The input consists of a single line containing $(n - 1)$ natural numbers in standard input ($0 < n < 100$). The numbers are separated by space. Your program should print the number on the missing card.

Additional requirements for bonus points

- Define and use the function `max` accepting the array and returning the maximum in the array.

Input	**Output**
`7 4 2 1 9 3 6 8 10`	`5`
`1 3 2 5 4 6`	`7`