

## Problem C. Random Shuffle

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Prof. Pang is selecting teams that advance to the world final contest. As the regionals are cancelled, he uses random shuffle to rank the teams. There are  $n$  teams in total. His code is as follows:

```
uint64_t x;//uint64_t represents 64-bit unsigned integer
int n;
uint64_t rand() {//this is a xor-shift random generator
    x ^= x << 13;
    x ^= x >> 7;
    x ^= x << 17;
    return x;
}
int main() {
    cin >> n;
    cin >> x;
    for (int i = 1; i <= n; i++) {//random shuffle [1, 2,..., n]
        a[i] = i;
        swap(a[i], a[rand() % i + 1]);
    }
    for (int i = 1; i <= n; i++) {//print the result
        cout << a[i] << (i == n ? '\n' : ' ');
    }
}
```

He compiled and ran his code and then entered  $n$  and some special nonnegative integer  $x$ . He printed the result on paper.

One day later, Prof. Pang forgot his choice for  $x$ . You are given the result of the code and the integer  $n$ . Please recover the number  $x$  that Prof. Pang had entered.

### Input

The first line contains a single integer  $n$  ( $50 \leq n \leq 100000$ ) – the number of teams.

The next line contains  $n$  integers – the result printed by Prof. Pang's code. It is guaranteed that the result is correct, i.e., there exists an integer  $x$  ( $0 \leq x \leq 2^{64} - 1$ ) that leads to the result.

### Output

Output the integer  $x$  ( $0 \leq x \leq 2^{64} - 1$ ) Prof. Pang had entered. If there are multiple possible  $x$ 's, print any one.

### Example

standard input
50
36 22 24 21 27 50 28 14 25 34 18 43 47
13 30 7 10 48 20 16 29 9 8 15 3 31 12
38 19 49 37 1 46 32 4 44 11 35 6 33 26
5 45 17 39 40 2 23 42 41
standard output
16659580358178468547

---

## Note

Note that the second line of the sample input is wrapped to fit in the width of page.