Hello, GCPC!

Alice is the new General Code Performance Caretaker (GCPC) at her company. Her job is to check the code written by the developers for bad performance and improve it if possible. The code is often strange since a company-wide policy is to write every piece of code in three programming languages. This is to ensure that the code runs on as many platforms as possible although Alice doubts that this is the best way to do it. Either way, she found the following code written in C++, Java, and Python and wants to improve it.

C++:

```cpp
#include <algorithm>
#include <iostream>
#include <vector>

using namespace std;

int main() {
  int n;
  cin >> n;

  vector<long long> a(n);
  for(int i = 0; i < n; i++) cin >> a[i];

  vector<long long> p;
  for(int i = 0; i < n; i++) {
    for(int j = 0; j < n; j++) {
      for(int k = 0; k < n; k++) {
        p.push_back(a[i]*a[j]*a[k]);
      }
    }
  }

  sort(p.begin(), p.end());

  cout << p[p.size()-1] << endl;
  return 0;
}
```

Java:

```java
import java.util.Collections;
import java.util.LinkedList;
import java.util.Scanner;

public class HelloGCPC {
  public static void main(String[] args) {
    Scanner s = new Scanner(System.in);

    int n= s.nextInt();

    long[] a = new long[n];
    for(int i = 0; i < n; i++) a[i] = s.nextLong();

    LinkedList<Long> p = new LinkedList<>();
    for(int i = 0; i < n; i++) {
      for(int j = 0; j < n; j++) {
        for(int k = 0; k < n; k++) {
          p.add(a[i]*a[j]*a[k]);
        }
      }
    }

    Collections.sort(p);

    System.out.println(p.get(p.size()-1));
  }
}
```

Python:

```
n = int(raw_input())
a = map(long, raw_input().split())

p = []
for i in range(n):
  for j in range(n):
    for k in range(n):
      p.append(a[i]*a[j]*a[k])
p.sort()

print p[len(p)-1]
```

Can you help her rewrite the code in one of the languages such that it prints the same result for all inputs but always runs within a second?

Input

The input consists of:

- one line with an integer n ($1 \leq n \leq 10^4$), where n is the number of integers read by the code;
- one line with n integers $a_1, \ldots, a_n$ ($0 \leq a_i \leq 10^6$ for all $1 \leq i \leq n$), where $a_1, \ldots, a_n$ are the numbers used by the code above.

Output

Output the same as the code above does.

Sample Input 1	Sample Output 1
3 1 2 3	27

Sample Input 2	Sample Output 2
5 344 823 3950 167 4784	109489762304