

# Hướng dẫn học lập trình C++ cơ bản đến nâng cao

Dinh Xuan Minh

Ngày 17 tháng 7 năm 2025

## Mục lục

<b>1 ĐỀ HSG 9 THCS TỈNH HẢI DƯƠNG NĂM HỌC 2024-2025</b>	<b>1</b>
1.1 Bài 1: Số tam giác . . . . .	1
1.2 Bài 2: Chia phần . . . . .	3
1.3 Bài 3: Kiểm tra số chẵn lẻ . . . . .	4
1.4 Bài 4: Tích lớn nhất . . . . .	5
1.5 Bài 5: Đếm số cặp . . . . .	7

## Lời nói đầu

Cuốn sách này tổng hợp các đề thi và lời giải chi tiết các bài lập trình dành cho học sinh giỏi lớp 9. Mỗi bài gồm phần đề, phân tích, hướng dẫn giải và code mẫu.

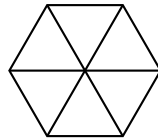
## 1 ĐỀ HSG 9 THCS TỈNH HẢI DƯƠNG NĂM HỌC 2024-2025

Thời gian làm bài: 150 phút

Độ khó:

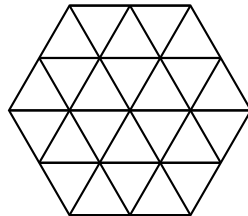
### 1.1 Bài 1: Số tam giác

**Đề bài:** Một lục giác đều với độ dài cạnh là số nguyên dương có thể được ghép bằng một số tam giác đều có độ dài cạnh bằng 1. Ví dụ dưới đây là hai hình lục giác đều được ghép bằng các tam giác đều cạnh 1.



Hình 1

Hình lục giác đều độ dài cạnh 1  
được ghép bằng 6 tam giác đều độ dài cạnh 1



Hình lục giác đều độ dài cạnh 2 được ghép bằng  
24 tam giác đều độ dài cạnh 1

**Yêu cầu:** Hỏi rằng số tam giác đều tối thiểu là bao nhiêu để ghép được  $n$  hình lục giác đều với độ dài các cạnh lần lượt là  $1, 2, \dots, n$ .

**Dữ liệu vào:**

Gồm  $T + 1$  dòng:

- Dòng đầu tiên chứa số nguyên dương  $T$  ( $1 \leq T \leq 10^5$ ) là số lượng test.
- Tiếp theo là  $T$  dòng, mỗi dòng tiếp theo chứa một số nguyên dương  $n$  ( $1 \leq n \leq 10^6$ ) mô tả một bộ dữ liệu.

**Dữ liệu ra:** Gồm  $T$  dòng, mỗi dòng in ra số tam giác đều tối thiểu cần thiết cho bộ dữ liệu tương ứng.

**Ràng buộc dữ liệu:**

- Subtask 1 (60 điểm):  $T \leq 10$ ,  $n \leq 1000$ .
- Subtask 2 (40 điểm):  $10 < T \leq 10^5$ .

**Ví dụ:**

Input
3
1
2
3
Output
6
30
84

**Hướng dẫn giải:**

**Solution C++:**

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int T;
5     cin >> T;
6     while (T--) {
7         int n;
8         cin >> n;
9         long long result = 3 * n * (n + 1);
10        cout << result << endl;
11    }
12    return 0;
13 }

```

## 1.2 Bài 2: Chia phần

**Đề bài:** Cho dãy số nguyên dương  $a_1, a_2, \dots, a_n$ . Chia dãy này thành hai phần

- Phần thứ nhất gồm các số  $a_1, a_2, \dots, a_k$ .
- Phần thứ hai gồm các số còn lại.

**Yêu cầu:** Gọi  $T_1$  và  $T_2$  lần lượt là tổng các số trong phần thứ nhất và phần thứ hai. Hãy tìm giá trị nhỏ nhất của  $|T_1 - T_2|$ . **Dữ liệu vào:**

Gồm hai dòng:

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $2 < n \leq 10^6$ ) là số lượng phần tử trong dãy.
- Dòng thứ hai chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $|a_i| \leq 10^9, \forall i = 1, 2, \dots, n$ ) cách nhau bởi dấu cách.

**Dữ liệu ra:** Gồm một dòng duy nhất chứa giá trị nhỏ nhất của  $|T_1 - T_2|$ .

**Ràng buộc dữ liệu:**

- Subtask 1 (75 điểm):  $n \leq 5000$ .
- Subtask 2 (25 điểm):  $n > 5000$ .

**Ví dụ:**

**Input**

5  
1 2 3 4 5

**Output**

3

**Hướng dẫn giải:** Dùng toán tử điều kiện hoặc hàm so sánh.

**Solution C++:**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int n;
5     cin >> n;
6     long long a[n];
7     long long sum = 0;
8     for (int i = 0; i < n; i++) {
9         cin >> a[i];
10        sum += a[i];
11    }
12
13    long long min_diff = LLONG_MAX;
14    long long T1 = 0;
15
16    for (int i = 0; i < n - 1; i++) {
17        T1 += a[i];
18        long long T2 = sum - T1;
19        min_diff = min(min_diff, abs(T1 - T2));
20    }
21
22    cout << min_diff << endl;
23    return 0;
24 }
```

### 1.3 Bài 3: Kiểm tra số chẵn lẻ

**Đề bài:** An có  $n$  đoạn thẳng. Cậu ta nhận thấy rằng một số đoạn thẳng cùng chiều dài nên có thể xếp thành những hình vuông.

**Yêu cầu:**

Hỏi rằng số hình vuông nhiều nhất An có thể xếp được là bao nhiêu? **Dữ liệu vào:** Gồm hai dòng:

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $1 \leq n \leq 3 \times 10^5$ ) là số lượng đoạn thẳng.
- Dòng thứ hai chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^{18}$ ) là độ dài của các đoạn thẳng.

**Dữ liệu ra:** Gồm một số nguyên duy nhất là số lượng hình vuông nhiều nhất An có thể xếp được. **Ràng buộc dữ liệu:**

- Subtask 1 (30%):  $n \leq 2000, a_i \leq 10^6$ .
- Subtask 2 (30%):  $n > 2000, a_i \leq 10^6$ .
- Subtask 3 (40%):  $n > 2000, a_i \leq 10^{18}$ .

**Ví dụ:**

## Input

```
9
2 2 2 9 2 2 2 2 2
```

## Output

```
2
```

**Hướng dẫn giải:** Kiểm tra  $n$  chia hết cho 2 hay không.

**Solution C++:**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int n;
5     cin >> n;
6     long long a[n];
7     for (int i = 0; i < n; i++) {
8         cin >> a[i];
9     }
10
11     map<long long, int> count_map;
12     for (int i = 0; i < n; i++) {
13         count_map[a[i]]++;
14     }
15
16     long long squares = 0;
17     for (auto& pair : count_map) {
18         squares += pair.second / 4;
19     }
20
21     cout << squares << endl;
22
23     return 0;
24 }
```

## 1.4 Bài 4: Tích lớn nhất

**Đề bài:** Cho một dãy số nguyên dương  $a_1, a_2, \dots, a_n$ .

**Yêu cầu:** Hãy tính giá trị lớn nhất của biểu thức  $a_i \times a_j \times a_k$  với  $1 \leq i < j < k \leq n$ .

**Dữ liệu vào:** Gồm hai dòng:

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $3 \leq n \leq 3 \times 10^5$ ) là số lượng phần tử trong dãy.
- Dòng thứ hai chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $|a_i| \leq 10^6$ ) cách nhau bởi dấu cách.

**Dữ liệu ra:** Gồm một số nguyên duy nhất là giá trị lớn nhất của biểu thức  $a_i \times a_j \times a_k$ .

**Ràng buộc dữ liệu:**

- Subtask 1 (40%):  $n \leq 100$ .
- Subtask 2 (60%):  $n > 100$ .

Ví dụ:

Input

6  
5 2 10 1 3 2

Output

150

Hướng dẫn giải:

Solution C++:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7     long long max_product = 0;
8     long long a[3];
9
10    for (int i = 0; i < n; i++) {
11        long long x;
12        cin >> x;
13        if (i < 3) {
14            a[i] = x;
15        } else {
16            if (x > a[0]) {
17                a[0] = x;
18            }
19            if (x > a[1]) {
20                a[1] = a[0];
21                a[0] = x;
22            } else if (x > a[2]) {
23                a[2] = x;
24            }
25        }
26    }
27
28    max_product = a[0] * a[1] * a[2];
29    cout << max_product << endl;
30
31    return 0;
32 }
```

## 1.5 Bài 5: Đếm số cặp

**Đề bài:** Cho một dãy số nguyên dương  $a_1, a_2, \dots, a_n$ . **Yêu cầu:** Hãy đếm số cặp  $(i, j)$  với  $1 \leq i < j \leq n$  thỏa mãn tính chất: Số  $a_i \times a_j$  là một số chính phương (số nguyên dương  $x$  được gọi là chính phương nếu tồn tại một số nguyên dương  $y$  sao cho  $x = y^2$ ). **Dữ liệu vào:** Gồm hai dòng:

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $n \leq 10^6$ ) là số lượng phần tử trong dãy.
- Dòng thứ hai chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^6, \forall i = 1, 2, \dots, n$ ) cách nhau bởi dấu cách.

**Dữ liệu ra:** Gồm một số nguyên duy nhất là số lượng cặp  $(i, j)$  thỏa mãn tính chất đã nêu. **Ràng buộc dữ liệu:**

- Subtask 1 (40%):  $n \leq 2000$ .
- Subtask 2 (40%):  $n > 2000, a_i \leq 10^4$ .
- Subtask 3 (20%):  $n > 2000, a_i \leq 10^6$ .

**Ví dụ:**

Input

5  
2 8 3 75 27

Output

4

**Hướng dẫn giải: Solution C++:**

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int n;
5     cin >> n;
6     vector<int> a(n);
7     for (int i = 0; i < n; i++) {
8         cin >> a[i];
9     }
10
11     unordered_map<int, int> freq;
12     for (int i = 0; i < n; i++) {
13         freq[a[i]]++;
14     }
15
16     long long count = 0;
17     for (auto& pair1 : freq) {
18         for (auto& pair2 : freq) {
19             if (pair1.first <= pair2.first) {
20                 int product = pair1.first * pair2.first;

```

```
21         int root = sqrt(product);
22         if (root * root == product) {
23             if (pair1.first == pair2.first) {
24                 count += (long long)pair1.second * (pair1.
25                     second - 1) / 2;
26             } else {
27                 count += (long long)pair1.second * pair2.
28                     second;
29             }
30         }
31     }
32
33     cout << count << endl;
34
35     return 0;
36 }
```