

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

Dinh Xuan Minh

Ngày 21 tháng 7 năm 2025

Mục lục

1	ĐỀ HSG 9 THCS TỈNH HẢI DƯƠNG NĂM HỌC 2024-2025	2
1.1	Bài 1: Số tam giác	2
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
2	ĐỀ HSG 9 THCS TỈNH ĐIỆN BIÊN NĂM HỌC 2024-2025	8
2.1	Bài 1: Tính tiền	8
2.2	Bài 2: Số cùng nhau	9
2.3	Bài 3: Thu nhập	10
2.4	Bài 4: Đếm dây con liên tiếp	12
3	ĐỀ HSG 9 THCS TỈNH NGHỆ AN NĂM HỌC 2024-2025	13
3.1	Bài 1: Số chính phương	13
3.2	Bài 2: Cây thông	15
3.3	Bài 3: Trồng cây	17
3.4	Bài 4: Giáng sinh	19
4	ĐỀ HSG 9 THCS TỈNH THANH HÓA NĂM HỌC 2024-2025	21
4.1	Bài 1: Diện tích	21
4.2	Bài 2: Số lớn	22
4.3	Bài 3: Hộp số	24
4.4	Bài 4: Bộ số chung nhỏ nhất	25

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.

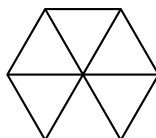
1 ĐỀ HSG 9 THCS TỈNH HẢI DƯƠNG NĂM HỌC 2024-2025

Thời gian làm bài: 150 phút

Độ khó:

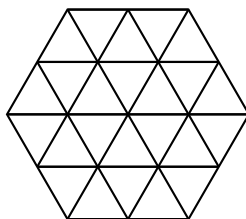
1.1 Bài 1: Số tam giác

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     }
34
35     cout << count << endl;
36
37     return 0;
38 }

```

2 ĐỀ HSG 9 THCS TỈNH ĐIỆN BIÊN NĂM HỌC 2024-2025

Thời gian làm bài: 150 phút Độ khó:

2.1 Bài 1: Tính tiền

Đề bài: Trong đợt Hội chợ thương mại Điện Biên năm 2024. Để kích cầu một doanh nghiệp đã đưa ra chương trình khuyến mãi. Theo đó, nếu tổng giá trị hóa đơn lớn hơn hoặc bằng 2,000,000 đồng, khách hàng sẽ được giảm giá 15% trên tổng giá trị hóa đơn. Nếu không đạt điều kiện trên, khách hàng sẽ phải thanh toán toàn bộ giá trị hóa đơn mà không được giảm giá.

Yêu cầu:

Viết chương trình để tính số tiền thực tế khách hàng phải thanh toán dựa trên: Số lượng hàng bán (ký hiệu là a), đơn giá của mỗi mặt hàng (ký hiệu là b đồng).

Dữ liệu vào:

Gồm một dòng duy nhất chứa hai số nguyên dương a và b ($1 \leq a \leq 10^4, 2 \times 10 \leq b \leq 10^9$) cách nhau bởi dấu cách.

Dữ liệu ra:

Gồm một dòng duy nhất số tiền thực tế khách hàng cần thanh toán. Kết quả đảm bảo là số nguyên.

Ràng buộc dữ liệu:

- Subtask 1 (80%): $1 \leq a \leq 100$, $2 \times 10 \leq b \leq 10^6$.
- Subtask 2 (20%): $10^2 < a \leq 10^4$, $10^6 < b \leq 10^9$.

Ví dụ:**Input**

2 1000000

Output

1700000

Hướng dẫn giải: Solution C++:

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     long long a, b;
5     cin >> a >> b;
6     long long total = a * b;
7     if (total >= 2000000) {
8         total -= total * 15 / 100;
9     }
10    cout << total << endl;
11    return 0;
12 }
```

2.2 Bài 2: Số cùng nhau

Đề bài: Trong một ngôi làng cổ, các số tự nhiên thường kết đôi để cùng nhau thực hiện những nhiệm vụ đặc biệt. Nhưng không phải cặp số nào cũng có thể đồng hành, chỉ những cặp "tương thích hoàn hảo" mới được chọn. Một cặp số i và j được xem là tương thích hoàn hảo nếu chúng không có bất kỳ ước chung nào khác ngoài số 1, nghĩa là $\text{UCLN}(i, j) = 1$.

Yêu cầu:

Nhiệm vụ của bạn là giúp trưởng làng đếm xem có bao nhiêu cặp số cùng nhau trong đoạn $[a, b]$ được chọn làm bạn đồng hành lý tưởng.

Dữ liệu vào:

Một dòng duy nhất chứa số nguyên dương a và b , cách nhau bởi dấu cách ($1 \leq a < b \leq 10^3$).

Dữ liệu ra:

In ra một số nguyên là số lượng cặp số (i, j) thỏa mãn điều kiện trên.

Ràng buộc dữ liệu:

- Subtask 1 (40%): $1 \leq a, b \leq 10$.
- Subtask 2 (30%): $10 < a, b \leq 100$.

- Subtask 3 (30%): $100 < a, b \leq 1000$.

Ví dụ:

Input

1 5

Output

9

Giải thích: Trong đoạn $[1, 5]$, các cặp số cùng nhau là: $(1, 2)$, $(1, 3)$, $(1, 4)$, $(1, 5)$, $(2, 3)$, $(2, 5)$, $(3, 4)$, $(3, 5)$ và $(4, 5)$. Tổng cộng có 9 cặp.

Hướng dẫn giải:

Sử dụng hàm Euler $\varphi(k)$ để đếm số lượng số nguyên nhỏ hơn k và nguyên tố cùng nhau với k .

Solution C++:

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 const int N = 1e6 + 5;
4 int phi[N];
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 1; i <= n; i++) phi[i] = i;
9     for (int i = 2; i <= n; i++) {
10         if (phi[i] == i) {
11             for (int j = i; j <= n; j += i)
12                 phi[j] -= phi[j] / i;
13         }
14     }
15     long long res = 0;
16     for (int i = 2; i <= n; i++) res += phi[i];
17     cout << res << endl;
18     return 0;
19 }
```

2.3 Bài 3: Thu nhập

Đề bài: Để biết được mức thu nhập trung bình hàng tháng của các hộ dân trong thành phố. Thành phố đã tiến hành khảo sát N hộ dân, hộ dân thứ i có thu nhập a_i triệu đồng. Nhìn vào số liệu thống kê nhận thấy rằng mỗi hộ dân có thu nhập khác nhau, lãnh đạo thành phố muốn biết mức thu nhập thấp nhất và cao nhất cũng như mức thu nhập nào phổ biến trong các hộ dân nhất để thành phố có kế hoạch phát triển kinh tế giàu mạnh.

Yêu cầu: Hãy cho biết thu nhập thấp nhất, cao nhất và đếm các hộ dân có thu nhập phổ biến nhất trong thành phố.

Dữ liệu vào: Gồm hai dòng:

- Dòng đầu tiên chứa số nguyên dương N ($1 \leq N \leq 10^5$) là số lượng hộ dân.

- Dòng thứ hai chứa N số nguyên dương a_i ($1 \leq a_i \leq 10^9, i = 1 \dots N$) là thu nhập của các hộ dân, cách nhau bởi dấu cách.

Dữ liệu ra: Gồm hai dòng:

- Dòng đầu tiên in ra thu nhập thấp nhất và cao nhất, cách nhau bởi dấu cách.
- Dòng thứ hai in ra số nguyên dương là số hộ dân nhiều nhất có mức thu nhập bằng nhau.

Ràng buộc dữ liệu:

- Subtask 1 (40%): $1 \leq N \leq 10^2$.
- Subtask 2 (30%): $10^2 < N \leq 10^3$.
- Subtask 3 (30%): $10^3 < N \leq 10^5$.

Ví dụ:

Input

9
5 1 5 8 6 2 3 6 3

Output

1 8
2

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     int min_income = *min_element(a.begin(), a.end());
12     int max_income = *max_element(a.begin(), a.end());
13
14     unordered_map<int, int> freq;
15     for (int i = 0; i < n; i++) {
16         freq[a[i]]++;
17     }
18
19     int max_count = 0;
20     for (auto& pair : freq) {
21         max_count = max(max_count, pair.second);
22     }
```

```

23
24     cout << min_income << " " << max_income << endl;
25     cout << max_count << endl;
26
27     return 0;
28 }
```

2.4 Bài 4: Đếm dãy con liên tiếp

Đề bài:

Cho dãy số A có n số nguyên a_1, a_2, \dots, a_n . Một dãy con liên tiếp các số hạng của dãy A là dãy các số hạng từ số hạng a_i đến số hạng a_j ($1 \leq i \leq j \leq n$).

Yêu cầu:

Hãy cho biết dãy A có bao nhiêu dãy con liên tiếp mà giá trị tuyệt đối của tổng các số hạng trong dãy con đó lớn hơn một số nguyên dương S cho trước.

Dữ liệu vào:

Gồm hai dòng:

- Dòng thứ nhất chứa hai số nguyên dương n và S ($n \leq 10^5, S \leq 10^{14}$);
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$). Hai số liên tiếp trên cùng dòng được ghi cách nhau bởi dấu cách.

Dữ liệu ra:

Gồm một số nguyên duy nhất là số dãy con liên tiếp thỏa mãn yêu cầu của bài toán.

Ràng buộc dữ liệu:

- Subtask 1 (50%): $n \leq 100$.
- Subtask 2 (30%): $n \leq 10^3$.
- Subtask 3 (20%): $n \leq 10^5$.

Ví dụ 1:

Input
4 4
5 - 1 8 -5
Output
6

Giải thích: Các dãy con liên tiếp có tổng tuyệt đối lớn hơn 4 là:

$$\{5\}, \{8\}, \{-1, 8\}, \{5, -1, 8\}, \{5, -1, 8, -5\}$$

Ví dụ 2:

Input

10 7
-4 9 2 -11 -3 8 -6 5 -3 1

Output

12

Hướng dẫn giải:

Sử dụng prefix sum và hai con trỏ để đếm số lượng dãy con liên tiếp có tổng tuyệt đối lớn hơn S .

Solution C++:

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 typedef long long ll;
4 int main() {
5     int n;
6     ll S;
7     cin >> n >> S;
8     vector<ll> a(n+1, 0), prefix(n+1, 0);
9     for (int i = 1; i <= n; ++i) {
10         cin >> a[i];
11         prefix[i] = prefix[i-1] + a[i];
12     }
13     ll res = 0;
14     for (int l = 1; l <= n; ++l) {
15         for (int r = l; r <= n; ++r) {
16             ll sum = prefix[r] - prefix[l-1];
17             if (abs(sum) > S) res++;
18         }
19     }
20     cout << res << endl;
21     return 0;
22 }
```

3 ĐỀ HSG 9 THCS TỈNH NGHỆ AN NĂM HỌC 2024-2025

Thời gian làm bài: 150 phút Độ khó:

3.1 Bài 1: Số chính phương

Đề bài:

Trong buổi ôn tập hôm nay, thầy giáo đã chuẩn bị một số món quà để trao tặng cho các bạn trong đội tuyển học sinh giỏi trả lời đúng bài toán về số học của thầy như sau:

Cho hai số nguyên dương L, R ($1 \leq L \leq R \leq 10^{18}$). Hãy đếm số lượng số chính phương trong đoạn $[L, R]$.

Biết rằng số chính phương là số bằng bình phương của một số nguyên dương. Ví dụ: 1, 4, 9, 16, 25, ... là các số chính phương vì chúng là bình phương của các số nguyên dương 1, 2, 3, 4, 5,

Rất nhanh chóng An đã đưa ra kết quả của bài toán. Em hãy lập trình giải quyết bài toán trên để biết xem An có được nhận phần thưởng của thầy giáo hay không.

Yêu cầu:

Hãy viết chương trình đếm số lượng số chính phương trong đoạn $[L, R]$.

Dữ liệu vào:

Gồm một dòng duy nhất chứa hai số nguyên dương L, R . ($1 \leq L \leq R \leq 10^{18}$).

Dữ liệu ra:

Gồm một số nguyên duy nhất là số lượng số chính phương trong đoạn $[L, R]$.

Ràng buộc dữ liệu:

- Subtask 1 (50%): $1 \leq L, R \leq 10^6$.
- Subtask 2 (50%): $10^6 < L, R \leq 10^{18}$.

Ví dụ:

Input
4 30
Output
4

Giải thích: Trong đoạn $[4, 30]$ có các số chính phương là: 4, 9, 16, 25. Tổng cộng có 4 số chính phương.

Hướng dẫn giải: Sử dụng hàm căn bậc hai để tìm số lượng số chính phương trong đoạn $[L, R]$. **Solution C++:**

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     long long L, R;
5     cin >> L >> R;
6
7     long long left = ceil(sqrt(L));
8     long long right = floor(sqrt(R));
9
10    if (left > right) {
11        cout << 0 << endl;
12    } else {
13        cout << right - left + 1 << endl;
14    }
15
16    return 0;
17 }
```

3.2 Bài 2: Cây thông

Đề bài: Chào đón Giáng sinh an lành, một cửa hàng có chương trình quà tặng đặc biệt. Lối vào của cửa hàng được trang trí bởi hai cây thông, cây thứ nhất treo n tấm thẻ ghi các giá trị lần lượt là A_1, A_2, \dots, A_n và cây thứ hai cũng có n tấm thẻ ghi các giá trị lần lượt là B_1, B_2, \dots, B_n . Người khách nào chọn được cặp thẻ A_i và B_j ($1 \leq i, j \leq n$) sao cho $|A_i + B_j|$ là giá trị nhỏ nhất thì sẽ được tặng một cây thông mình thích nhất trong cửa hàng.

Yêu cầu:

Em hãy giúp cửa hàng xác định giá trị nhỏ nhất của $|A_i + B_j|$ để tặng quà cho người khách lựa chọn được cặp thẻ may mắn.

Dữ liệu vào: Gồm ba dòng:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^6$) là số lượng tấm thẻ trên mỗi cây thông.
- Dòng thứ hai chứa n số nguyên A_1, A_2, \dots, A_n ($-10^9 \leq A_i \leq 10^9$) là giá trị ghi trên các tấm thẻ của cây thông thứ nhất.
- Dòng thứ ba chứa n số nguyên B_1, B_2, \dots, B_n ($-10^9 \leq B_i \leq 10^9$) là giá trị ghi trên các tấm thẻ của cây thông thứ hai.

Dữ liệu ra: Gồm một số nguyên duy nhất là giá trị nhỏ nhất của $|A_i + B_j|$.

Ràng buộc dữ liệu:

- Subtask 1 (60%): $1 \leq n \leq 10^3$.
- Subtask 2 (40%): $10^3 < n \leq 10^6$.

Ví dụ:

Input
5
-9 3 -17 -5 3
-1 7 2 3 20
Output
2

Giải thích: Trong ví dụ trên, ta có các cặp thẻ A_i và B_j như sau:

- $A_1 + B_1 = -9 - 1 = -10$.
- $A_1 + B_2 = -9 + 7 = -2$.
- $A_1 + B_3 = -9 + 2 = -7$.
- $A_1 + B_4 = -9 + 3 = -6$.
- $A_1 + B_5 = -9 + 20 = 11$.
- $A_2 + B_1 = 3 - 1 = 2$.

- $A_2 + B_2 = 3 + 7 = 10$.
- $A_2 + B_3 = 3 + 2 = 5$.
- $A_2 + B_4 = 3 + 3 = 6$.
- $A_2 + B_5 = 3 + 20 = 23$.
- $A_3 + B_1 = -17 - 1 = -18$.
- $A_3 + B_2 = -17 + 7 = -10$.
- $A_3 + B_3 = -17 + 2 = -15$.
- $A_3 + B_4 = -17 + 3 = -14$.
- $A_3 + B_5 = -17 + 20 = 3$.
- $A_4 + B_1 = -5 - 1 = -6$.
- $A_4 + B_2 = -5 + 7 = 2$.
- $A_4 + B_3 = -5 + 2 = -3$.
- $A_4 + B_4 = -5 + 3 = -2$.
- $A_4 + B_5 = -5 + 20 = 15$.
- $A_5 + B_1 = 3 - 1 = 2$.
- $A_5 + B_2 = 3 + 7 = 10$.
- $A_5 + B_3 = 3 + 2 = 5$.
- $A_5 + B_4 = 3 + 3 = 6$.
- $A_5 + B_5 = 3 + 20 = 23$.

Từ các cặp trên, ta thấy rằng giá trị nhỏ nhất của $|A_i + B_j|$ là 2, với các cặp (A_2, B_1) và (A_4, B_2) .

Hướng dẫn giải: Sắp xếp hai mảng A và B , sau đó sử dụng hai con trỏ để tìm cặp thẻ có tổng gần bằng 0 nhất. **Solution C++:**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int n;
5     cin >> n;
6     vector<long long> A(n), B(n);
7
8     for (int i = 0; i < n; i++) {
9         cin >> A[i];
10    }
11
12    for (int i = 0; i < n; i++) {
13        cin >> B[i];
```



```

14     }
15
16     sort(A.begin(), A.end());
17     sort(B.begin(), B.end());
18
19     long long min_sum = LLONG_MAX;
20     int i = 0, j = n - 1;
21
22     while (i < n && j >= 0) {
23         long long sum = A[i] + B[j];
24         min_sum = min(min_sum, abs(sum));
25
26         if (sum < 0) {
27             i++;
28         } else {
29             j--;
30         }
31     }
32
33     cout << min_sum << endl;
34
35     return 0;
36 }

```

3.3 Bài 3: Trồng cây

Đề bài: An là chủ nhiệm của CLB Sống Xanh nơi mình sinh sống. Nhân dịp lễ Giáng sinh và chuẩn bị đón tết Nguyên Đán, CLB phát động chiến dịch "Xanh quê hương" với nhiều hoạt động có ý nghĩa nhằm tạo môi trường Xanh - Sạch - Đẹp. Hoạt động đầu tiên trong chiến dịch là thực hiện trồng một hàng cây chạy dọc theo một tuyến đường.

Trên tuyến đường đã được đánh dấu n vị trí cách đều nhau để trồng cây, trong đó có một số vị trí đã được trồng cây từ trước. CLB gồm An và k thành viên sẽ trồng $k + 1$ cây vào $k + 1$ vị trí trống (mỗi thành viên sẽ trồng một cây). Để thuận tiện quản lí, An muốn tìm một vị trí trồng cây của mình và vị trí của k thành viên, sao cho vị trí thành viên xa nhất đến vị trí của An là ngắn nhất.

Yêu cầu: Hãy lập trình giúp An xác định giá trị nhỏ nhất của khoảng cách từ vị trí từ vị trí thành viên xa nhất đến vị trí của An.

Dữ liệu vào: Gồm hai dòng:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$) là số vị trí trồng cây và số thành viên trong CLB.
- Dòng thứ hai chứa một xâu nhị phân s gồm n phần tử biểu diễn trạng thái của n vị trí. Giá trị 0 biểu diễn vị trí trống, giá trị 1 biểu diễn vị trí đã trồng cây. (Dữ liệu đảm bảo số phần tử có giá trị 0 trong xâu s luôn lớn hơn k).

Dữ liệu ra: Gồm một số nguyên duy nhất là giá trị nhỏ nhất của khoảng cách từ vị trí thành viên xa nhất đến vị trí của An.

Ràng buộc dữ liệu:

- Subtask 1 (60%): $1 \leq n \leq 10^3$.

- Subtask 2 (40%): $10^3 < n \leq 10^5$.

Ví dụ:

Input

7 2
1010100

Output

2

Giải thích:

- Cách 1: Chọn các vị trí 2, 4, 6. An ở vị trí số 4 và khoảng cách đến thành viên xa nhất là $|6 - 4| = |2 - 4| = 2$.
- Cách 2: Chọn các vị trí 4, 6, 7. An ở vị trí số 6 và khoảng cách đến thành viên xa nhất là $|4 - 6| = 2$.

Hướng dẫn giải: Solution C++:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int n, k;
6     cin >> n >> k;
7     string s;
8     cin >> s;
9
10    vector<int> empty_positions;
11    for (int i = 0; i < n; i++) {
12        if (s[i] == '0') {
13            empty_positions.push_back(i);
14        }
15    }
16
17    int left = 0, right = empty_positions.size() - 1;
18    int min_distance = INT_MAX;
19
20    while (right - left + 1 > k) {
21        min_distance = min(min_distance, empty_positions[right] -
22                           empty_positions[left]);
23        left++;
24    }
25
26    cout << min_distance << endl;
27
28    return 0;
29 }
```

3.4 Bài 4: Giáng sinh

Đề bài: Giáng sinh là khoảng thời gian đẹp nhất trong năm. Hai anh em William và Jacica là hai diễn viên múa chính của đoàn nghệ thuật đỉnh cao số một thế giới. Các vở diễn của họ góp phần hồi sinh các giá trị đạo đức, văn hóa truyền thống, đem lại cho người xem năng lượng tích cực, cảm nhận sâu sắc về các giá trị tốt đẹp và sự bình yên trong tâm hồn. William và Jacica vừa trở về nhà sau chuyến lưu diễn vòng quanh thế giới và bắt đầu trang trí cây thông Noel của họ bằng những món đồ xinh xắn đã mua trong quá trình lưu diễn.

Họ đã mua n món đồ trang trí cây thông được xếp cạnh nhau trong một hộp dài, món đồ trang trí thứ i có màu A_i . Hộp được mở ở cả hai đầu, vì vậy các món đồ có thể được lấy ra từ bên trái hoặc bên phải của hộp. Hộp này trong suốt, nên William và Jacica có thể nhìn thấy màu sắc của từng món đồ trang trí.

Jacica nghĩ ra một trò chơi để việc trang trí cây thông trở nên thú vị hơn. Trò chơi diễn ra như sau: William và Jacica thay phiên nhau chơi, William là người bắt đầu. Người chơi trong lượt của mình sẽ lấy một món đồ trang trí từ hộp (có thể từ bên trái hoặc bên phải) và đặt nó lên cây thông. Nếu món đồ được lấy có màu chưa từng được người nào lấy trước đó, người chơi sẽ ghi được một điểm. Trò chơi kết thúc khi món đồ trang trí cuối cùng được lấy ra khỏi hộp. Người chiến thắng là người ghi được nhiều điểm hơn. Vì cả William và Jacica đều là những người chơi xuất sắc, họ sẽ chơi một cách tối ưu.

Yêu cầu: Em hãy đưa ra kết quả cuối cùng của trò chơi.

Dữ liệu vào: Gồm hai dòng:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 3000$) là số lượng món đồ trang trí trong hộp.
- Dòng thứ hai chứa n số nguyên dương A_1, A_2, \dots, A_n với A_i ($1 \leq A_i \leq n$) là màu sắc của các món đồ trang trí, cách nhau bởi dấu cách.

Dữ liệu ra: Gồm một dòng duy nhất gồm hai số, được nối bằng một ký tự ":" (Dấu hai chấm), lần lượt là điểm số của William và Jacica.

Ràng buộc dữ liệu:

- Subtask 1 (25%): $1 \leq A_i \leq 2$ với mọi $\forall i = 1, 2, \dots, n$.
- Subtask 2 (20%): $1 \leq n \leq 20$.
- Subtask 3 (10%): $1 \leq A_i \leq 20$ với mọi $\forall i = 1, 2, \dots, n$.
- Subtask 4 (20%): $1 \leq n \leq 300$.
- Subtask 5 (25%): $300 < n \leq 3000$.

Ví dụ 1:

Input

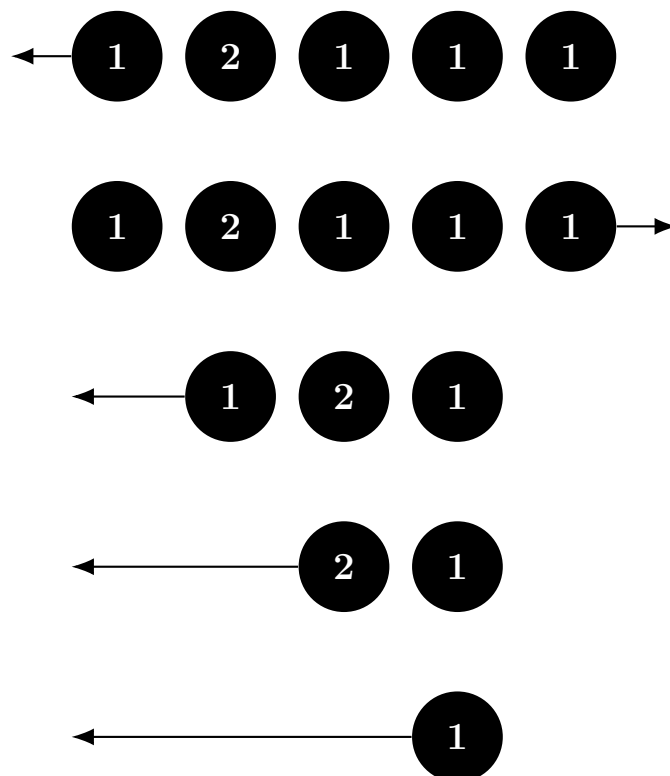
5
1 1 2 1 1

Output

1:1

Giải thích: Trong ví dụ này, Wiliam và Jacica sẽ chơi như sau:

- Đầu tiên Wiliam chọn món đồ trang trí có màu 1 từ bên trái, ghi được 1 điểm.
- Tiếp theo Jacica chọn món đồ trang trí có màu 1 từ bên phải và không có điểm vì màu này đã được Wiliam lấy.
- Tiếp theo Wiliam chọn màu 1 ở bên trái và không có điểm vì màu 1 đã được lấy.
- Tiếp theo Jacica chọn màu 2 ở về bên trái và được 1 điểm vì màu 2 chưa được lấy.
- Cuối cùng Wiliam chọn màu 1 và không được điểm vì màu 1 đã được lấy.



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     vector<vector<int>> dp(n, vector<int>(n, 0));
12
13     for (int len = 1; len <= n; len++) {
14         for (int l = 0; l + len - 1 < n; l++) {
15             int r = l + len - 1;
16             if (l == r) {

```

```

17         dp[l][r] = 1; // only one item
18     } else {
19         int left = A[l], right = A[r];
20         if (left != right) {
21             dp[l][r] = max(dp[l + 1][r] + (A[l] != A[l + 1]), dp[l][r - 1] + (A[r] != A[r - 1]));
22         } else {
23             dp[l][r] = max(dp[l + 1][r], dp[l][r - 1]);
24         }
25     }
26 }
27
28
29 int william_score = (n + dp[0][n - 1]) / 2;
30 int jacica_score = n - william_score;
31
32 cout << william_score << ":" << jacica_score << endl;
33
34 return 0;
35 }

```

4 ĐỀ HSG 9 THCS TỈNH THANH HÓA NĂM HỌC 2024-2025

Thời gian làm bài: 150 phút Độ khó:

4.1 Bài 1: Diện tích

Đề bài: Biết Nam đang ôn tập để tham gia kỳ thi học sinh giỏi cấp tỉnh môn Tin học. Bố đã nhờ bạn ấy lập trình để giải bài toán có nội dung như sau:

Cho 4 thanh sắt có độ dài lần lượt là a, b, c, d . Bố muốn Nam tạo một khung hình chữ nhật từ 4 thanh sắt đó (phải sử dụng cả 4 thanh), những đoạn dư của các thanh sắt (nếu có) sẽ được cắt bỏ. Hãy tìm diện tích lớn nhất của khung sắt được tạo thành.

Yêu cầu: Em hãy giúp bạn Nam giải bài toán trên.

Dữ liệu vào: Gồm một dòng duy nhất chứa 4 số nguyên dương a, b, c, d ($0 < a, b, c, d \leq 10^5$).

Dữ liệu ra: Gồm một số nguyên duy nhất là diện tích lớn nhất của khung hình chữ nhật được tạo thành từ 4 thanh sắt.

Ví dụ 1:

Input

7 3 4 6

Output

18

Ví dụ 2:

Input

5 5 5 5

Output

25

Hướng dẫn giải: Solution C++:

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int a, b, c, d;
6     cin >> a >> b >> c >> d;
7
8     int max_area = 0;
9     for (int w = 1; w <= 100000; w++) {
10         int h = min({a / w, b / w, c / w, d / w});
11         max_area = max(max_area, w * h);
12     }
13
14     cout << max_area << endl;
15     return 0;
16 }

```

4.2 Bài 2: Số lớn

Đề bài: Việt và Nam cùng nhau ôn luyện chủ đề xâu kí tự. Để buổi học trở nên thú vị hơn, mỗi bạn sẽ lần lượt đưa ra một bài toán dành cho bạn của mình. Bài toán của Việt dành cho Nam như sau: Cho một xâu ST bao gồm các kí tự chữ cái tiếng Anh (thường và hoa) và các kí tự số. Hãy thực hiện xóa đi các kí tự trong xâu ST để được một xâu mới $ST1$ chỉ còn K kí tự đều là kí tự số và khi giữ nguyên trật tự ban đầu thì xâu $ST1$ là xâu số lớn nhất có thể.

Yêu cầu: Em hãy giúp Nam tìm ra xâu $ST1$ theo đúng yêu cầu của bài toán.

Dữ liệu vào: Gồm hai dòng:

- Dòng đầu tiên chứa một xâu ST ($1 \leq |ST| \leq 10^5$) là xâu kí tự cần xử lý.
- Dòng thứ hai chứa một số nguyên dương K ($1 \leq K \leq |ST|$). Biết rằng trong xâu ST luôn đảm bảo có ít nhất K kí tự là số.

Dữ liệu ra: Gồm một dòng duy nhất là xâu $ST1$ theo đúng yêu cầu của bài toán.

Ràng buộc dữ liệu:

- Subtask 1 (40%): $|ST| \leq 18$, xâu ST chỉ gồm các kí tự số.
- Subtask 2 (60%): Không có ràng buộc gì thêm.

Ví dụ 1:

Input

AmN69pQ3e6
2

Output

96

Ví dụ 2:**Input**

Fish36colo99
3

Output

699

Hướng dẫn giải: Để giải bài toán này, ta cần thực hiện các bước sau:

- Đầu tiên, ta sẽ duyệt qua chuỗi ST và lưu trữ tất cả các kí tự số vào một danh sách.
- Sau đó, ta sẽ sắp xếp danh sách các kí tự số theo thứ tự giảm dần.
- Cuối cùng, ta sẽ lấy K kí tự đầu tiên trong danh sách đã sắp xếp để tạo thành chuỗi $ST1$.

Solution C++:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     string ST;
5     int K;
6     cin >> ST >> K;
7
8     vector<char> digits;
9     for (char c : ST) {
10         if (isdigit(c)) {
11             digits.push_back(c);
12         }
13     }
14
15     sort(digits.rbegin(), digits.rend());
16
17     string ST1 = "";
18     for (int i = 0; i < K; i++) {
19         ST1 += digits[i];
20     }
21
22     cout << ST1 << endl;
23     return 0;
```

24 }

4.3 Bài 3: Hộp số

Đề bài: Có n chiếc hộp được đánh số theo thứ tự từ 1 đến n và xếp chúng theo một hàng ngang. Mỗi chiếc hộp có một trong hai giá trị 0 hoặc 1. Thực hiện một lần thay đổi giá trị của tất cả các hộp từ vị trí i, j ($1 \leq i \leq j \leq n$) theo quy tắc "Những hộp có giá trị bằng 1 sẽ được thay đổi bằng 0 và ngược lại", để sau khi theo đổi thì trong n chiếc hộp nhận được số chiếc hộp có giá trị bằng 1 là nhiều nhất.

Yêu cầu: Đếm số hộp có giá trị bằng 1 nhiều nhất sau khi thay đổi như trên.

Dữ liệu vào: Gồm hai dòng:

- Dòng đầu tiên chứa một số nguyên dương n ($0 < n \leq 10^6$) là số lượng hộp.
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($a_i \in \{0, 1\}$) là giá trị của các hộp.

Dữ liệu ra: Gồm một số nguyên duy nhất là số lượng hộp có giá trị bằng 1 nhiều nhất sau khi thay đổi.

Ràng buộc dữ liệu:

- Subtask 1 (30%): $1 \leq n \leq 500$.
- Subtask 2 (30%): $500 < n \leq 8000$.
- Subtask 3 (40%): $8000 < n \leq 10^6$.

Ví dụ 1:

Input

8
1 0 0 1 1 0 0 0

Output

6

Hướng dẫn giải: Để giải bài toán này, ta có thể sử dụng thuật toán quét hai đầu (two-pointer) hoặc thuật toán tiền tố (prefix sum) để tìm đoạn con có số lượng hộp có giá trị bằng 1 nhiều nhất sau khi thay đổi. **Solution C++:**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7     vector<int> a(n);
8     for (int i = 0; i < n; i++) {
9         cin >> a[i];
10    }
11
12    int max_ones = 0;
```



```

13     for (int i = 0; i < n; i++) {
14         for (int j = i; j < n; j++) {
15             int count = 0;
16             for (int k = i; k <= j; k++) {
17                 count += (a[k] == 1) ? 0 : 1;
18             }
19             max_ones = max(max_ones, count);
20         }
21     }
22
23     cout << max_ones << endl;
24     return 0;
25 }

```

4.4 Bài 4: Bội số chung nhỏ nhất

Đề bài: Bội số chung nhỏ nhất của hai số nguyên x và y là số nguyên dương nhỏ nhất chia hết cho cả x và y , ký hiệu là $LCM(x, y)$.

Cho hai số nguyên dương a và b ($a \leq b$).

Yêu cầu: Hãy đếm số cặp số nguyên dương x, y sao cho $LCM(x, y)$ bằng tích các số nguyên liên tiếp từ a đến b (trường hợp a bằng b thì tích này bằng a).

Dữ liệu vào: Gồm hai dòng:

- Dòng đầu ghi số nguyên dương T ($T \leq 10$) là số lượng bộ test.
- T dòng tiếp theo, mỗi dòng chứa hai số nguyên dương a, b ($1 \leq a \leq b \leq 10^6$) là đầu và cuối của đoạn cần xét.

Dữ liệu ra: Gồm T dòng, mỗi dòng chứa một số nguyên duy nhất là số lượng cặp số tìm được khi chia hết cho $10^9 + 7$.

Ràng buộc dữ liệu:

- Subtask 1 (40%): $1 \leq a, b \leq 10$.
- Subtask 2 (30%): $1 \leq a, b \leq 100$.
- Subtask 3 (30%): Không có ràng buộc gì thêm.

Ví dụ 1:

Input

2
3 4
7 7

Output

15
3

Giải thích: Trong ví dụ trên, ta có hai bộ test:

- Bộ test đầu tiên: Tích các số nguyên liên tiếp từ 3 đến 4 là $3 \times 4 = 12$. Các cặp (x, y) sao cho $LCM(x, y) = 12$ là: $(1, 12)$, $(2, 6)$, $(3, 4)$, $(4, 3)$, $(6, 2)$, $(12, 1)$, $(2, 12)$, $(3, 12)$, $(4, 12)$, $(6, 12)$ và các hoán vị của chúng. Tổng cộng có 15 cặp.
- Bộ test thứ hai: Tích các số nguyên liên tiếp từ 7 đến 7 là 7. Các cặp (x, y) sao cho $LCM(x, y) = 7$ là: $(1, 7)$; $(7, 1)$; $(7, 7)$. Tổng cộng có 3 cặp.

Hướng dẫn giải: Để giải bài toán này, ta cần tính tích các số nguyên liên tiếp từ a đến b và sau đó đếm số cặp (x, y) sao cho $LCM(x, y)$ bằng tích đó. Ta có thể sử dụng công thức:

$$LCM(x, y) = \frac{x \times y}{GCD(x, y)}$$

Trong đó $GCD(x, y)$ là ước số chung lớn nhất của x và y . **Solution C++:**

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int T;
6     cin >> T;
7     while (T--) {
8         int a, b;
9         cin >> a >> b;
10        long long product = 1;
11        for (int i = a; i <= b; i++) {
12            product *= i;
13        }
14        int count = 0;
15        for (int x = 1; x <= product; x++) {
16            for (int y = 1; y <= product; y++) {
17                if (lcm(x, y) == product) {
18                    count++;
19                }
20            }
21        }
22        cout << count % (1000000007) << endl;
23    }
24    return 0;
25 }
```