

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

Dinh Xuan Minh

Ngày 23 tháng 7 năm 2025

## Mục lục

<b>1</b>	<b>ĐỀ HSG 9 THCS TỈNH HẢI DƯƠNG NĂM HỌC 2024-2025</b>	<b>2</b>
1.1	Bài 1: Số tam giác . . . . .	2
1.2	Bài 2: Chia phần . . . . .	3
1.3	Bài 3: Kiểm tra số chẵn lẻ . . . . .	5
1.4	Bài 4: Tích lớn nhất . . . . .	6
1.5	Bài 5: Đếm số cặp . . . . .	7
<b>2</b>	<b>ĐỀ HSG 9 THCS TỈNH ĐIỆN BIÊN NĂM HỌC 2024-2025</b>	<b>9</b>
2.1	Bài 1: Tính tiền . . . . .	9
2.2	Bài 2: Số cùng nhau . . . . .	10
2.3	Bài 3: Thu nhập . . . . .	11
2.4	Bài 4: Đếm dây con liên tiếp . . . . .	12
<b>3</b>	<b>ĐỀ HSG 9 THCS TỈNH NGHỆ AN NĂM HỌC 2024-2025</b>	<b>14</b>
3.1	Bài 1: Số chính phương . . . . .	14
3.2	Bài 2: Cây thông . . . . .	15
3.3	Bài 3: Trồng cây . . . . .	17
3.4	Bài 4: Giáng sinh . . . . .	19
<b>4</b>	<b>ĐỀ HSG 9 THCS TỈNH THANH HÓA NĂM HỌC 2024-2025</b>	<b>22</b>
4.1	Bài 1: Diện tích . . . . .	22
4.2	Bài 2: Số lớn . . . . .	23
4.3	Bài 3: Hộp số . . . . .	24
4.4	Bài 4: Bội số chung nhỏ nhất . . . . .	26
<b>5</b>	<b>ĐỀ HSG 10 THCS TỈNH THÁI BÌNH NĂM HỌC 2024-2025</b>	<b>27</b>
5.1	Bài 1: Số đặc biệt . . . . .	27
5.2	Bài 2: Vườn cây . . . . .	29
5.3	Bài 3: Dây con thình vượng . . . . .	31
5.4	Bài 4: Mã mặt hàng . . . . .	32

<b>6 ĐỀ HSG 9 THCS TỈNH HÀ TĨNH 2024 - 2025</b>	<b>34</b>
6.1 Bài 1: Số nguyên dương $k$ . . . . .	34
6.2 Bài 2: Nuôi cá cảnh . . . . .	35
6.3 Bài 3: Số nguyên tố . . . . .	36
6.4 Bài 4: Dây con . . . . .	37
<b>7 ĐỀ HSG 9 THCS TỈNH VĨNH PHÚC 2024 - 2025</b>	<b>38</b>
7.1 Bài 1: Quân Hậu . . . . .	38
7.2 Bài 2: Trung vị lớn nhất . . . . .	40
7.3 Bài 3: Xâu rút gọn . . . . .	41

## 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.

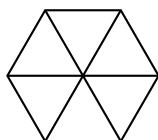
# 1 ĐỀ HSG 9 THCS TỈNH HẢI DƯƠNG NĂM HỌC 2024-2025

Thời gian làm bài: 150 phút

Độ khó:

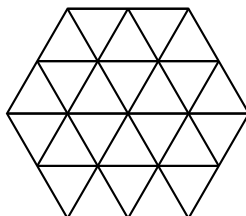
## 1.1 Bài 1: Số tam giác

**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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu vào:**

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

**Dữ liệu đầu 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 đầu 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 đầu 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 đầu 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 đầu 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;

```

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

**Ràng buộc dữ liệu:**

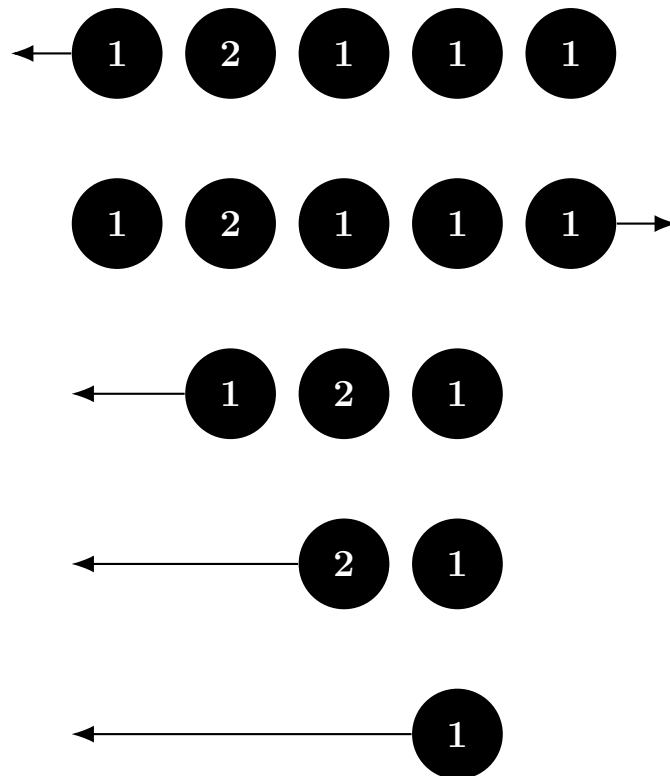
- Subtask 1 (25%):  $1 \leq A_i \leq 2$  với mọi  $\forall i = 1, 2, \dots, n$ .
- Subtask 2 (20%):  $1 \leq n \leq 20$ .
- Subtask 3 (10%):  $1 \leq A_i \leq 20$  với mọi  $\forall i = 1, 2, \dots, n$ .
- Subtask 4 (20%):  $1 \leq n \leq 300$ .
- Subtask 5 (25%):  $300 < n \leq 3000$ .

**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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 đầu 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 }
```

## 5 ĐỀ HSG 10 THCS TỈNH THÁI BÌNH NĂM HỌC 2024-2025

Thời gian làm bài: 150 phút Độ khó:

### 5.1 Bài 1: Số đặc biệt

**Đề bài:** Nam rất yêu thích các con số, đặc biệt là số nguyên tố. Một lần, trong giờ học Nam nhận được câu hỏi của thầy như sau: Số đặc biệt là một số nguyên dương mà có tổng các chữ số là một số nguyên tố. Cho số nguyên dương  $N$ . Hãy kiểm tra xem  $N$  có phải là số đặc biệt hay không?

**Yêu cầu:** Em hãy giúp Nam kiểm tra xem  $N$  có phải là số đặc biệt hay không.

**Dữ liệu đầu vào:** Gồm một dòng duy nhất chứa một số nguyên dương  $N$  ( $1 \leq N \leq 10^{255}$ ).

**Dữ liệu đầu ra:** Gồm một dòng gồm:

- Thông báo: YES nếu  $N$  là số đặc biệt.
- Thông báo: NO nếu  $N$  không phải là số đặc biệt.

**Ràng buộc dữ liệu:**

- Subtask 1 (50%):  $0 < N \leq 10^9$ .
- Subtask 2 (40%):  $10^9 < N \leq 10^{18}$ .
- Subtask 3 (10%):  $10^{18} < N \leq 10^{255}$ .

**Ví dụ 1:****Input**

23

**Output**

YES

**Giải thích:** Số 23 có tổng các chữ số là  $2 + 3 = 5$ , mà 5 là một số nguyên tố.

**Ví dụ 2:****Input**

17

**Output**

NO

**Giải thích:** Số 17 có tổng các chữ số là  $1 + 7 = 8$ , mà 8 không phải là một số nguyên tố.

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

- Tính tổng các chữ số của số  $N$ .
- Kiểm tra xem tổng đó có phải là một số nguyên tố hay không.

**Solution C++:**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 bool is_prime(int n) {
4     if (n <= 1) return false;
5     for (int i = 2; i * i <= n; i++) {
6         if (n % i == 0) return false;
7     }
8     return true;
9 }
10 int main() {
11     string N;
12     cin >> N;
```

```

13
14     int sum = 0;
15     for (char c : N) {
16         sum += c - '0';
17     }
18
19     if (is_prime(sum)) {
20         cout << "YES" << endl;
21     } else {
22         cout << "NO" << endl;
23     }
24
25     return 0;
26 }

```

## 5.2 Bài 2: Vườn cây

**Đề bài:** Một mảnh vườn hình chữ nhật được chia thành các ô đất nhỏ gồm  $M$  hàng,  $N$  cột. Trên các ô đất đó, bác Ba trồng các loại cây ăn quả, cây ở hàng  $i$  và cột  $j$  có sản lượng quả là  $a_{ij}$ . Mỗi đợt cuối năm, bác Ba muốn xem tổng sản lượng quả của các cây trên mỗi hàng dọc (cột) của khu vườn để bác có biện pháp chăm sóc hàng cây đó cho phù hợp.

**Yêu cầu:** Tính tổng sản lượng trái cây của các cây trên các hàng dọc (cột) trong khu vườn giúp bác Ba.

**Dữ liệu đầu vào:** Gồm  $M + 1$  dòng:

- Dòng đầu tiên chứa hai số nguyên dương  $M, N$  ( $1 \leq M, N \leq 10^4$ ).
- $M$  dòng tiếp theo, mỗi dòng chứa  $N$  số nguyên không âm. Giá trị ở dòng thứ  $i$  và cột thứ  $j$  là  $a_{ij}$  với ( $1 \leq i \leq M, 1 \leq j \leq N$ ) để mô tả sản lượng tại thời điểm thống kê của cây được trồng tại ô ở hàng  $i$  cột  $j$  của mảnh vườn.

**Dữ liệu đầu ra:** Gồm một dòng duy nhất chứa  $N$  số nguyên dương, mỗi số ghi cách nhau một dấu cách là tổng sản lượng trái cây của các cây trên các hàng dọc (cột) theo thứ tự.

**Ràng buộc dữ liệu:**

- Có 50% số test (tương ứng 50% số điểm) với  $1 \leq M, N \leq 10^2, 0 \leq a_{ij} \leq 10^3$ .
- Có 40% số test (tương ứng 40% số điểm) với  $10^2 < M, N \leq 10^3, 0 \leq a_{ij} \leq 10^8$ .
- Có 10% số test (tương ứng 10% số điểm) với  $10^3 < M, N \leq 10^4, 0 \leq a_{ij} \leq 10^{12}$ .

**Ví dụ 1:**

Input

```

3 4
1 3 5 7
2 4 6 9
5 6 9 0

```

**Output**

8 13 20 16

**Giải thích:** Trong ví dụ trên, ta có:

- Cột 1:  $1 + 2 + 5 = 8$ .
- Cột 2:  $3 + 4 + 6 = 13$ .
- Cột 3:  $5 + 6 + 9 = 20$ .
- Cột 4:  $7 + 9 + 0 = 16$ .

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

- Đọc dữ liệu đầu vào để lấy số hàng  $M$  và số cột  $N$ .
- Khởi tạo một mảng để lưu trữ tổng sản lượng trái cây của từng cột.
- Duyệt qua từng ô trong mảng vườn, cộng giá trị của ô đó vào tổng tương ứng của cột.
- In ra kết quả.

**Solution C++:**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int M, N;
6     cin >> M >> N;
7
8     vector<vector<int>> garden(M, vector<int>(N));
9     for (int i = 0; i < M; i++) {
10         for (int j = 0; j < N; j++) {
11             cin >> garden[i][j];
12         }
13     }
14
15     vector<int> col_sum(N, 0);
16     for (int j = 0; j < N; j++) {
17         for (int i = 0; i < M; i++) {
18             col_sum[j] += garden[i][j];
19         }
20     }
21
22     for (int j = 0; j < N; j++) {
23         cout << col_sum[j] << " ";
24     }
25     cout << endl;
26
27     return 0;
28 }
```

### 5.3 Bài 3: Dãy con thịnh vượng

Xét dãy số gồm  $n$  phần tử  $a_1, a_2, \dots, a_n$ . Một dãy con liên tiếp của dãy  $a_1, a_2, \dots, a_n$  là dãy số nguyên có dạng  $a_i, a_{i+1}, \dots, a_j$  với  $1 \leq i \leq j \leq n$ .

Một dãy con liên tiếp được gọi là thịnh vượng nếu tổng các phần tử của nó là một số nguyên dương. Một dãy con liên tiếp được gọi là dãy con thịnh vượng nếu tổng của các phần tử trong dãy con liên tiếp đó là lớn nhất trong tất cả các dãy con liên tiếp.

**Yêu cầu:** Cho trước một dãy số nguyên  $a_1, a_2, \dots, a_n$ . Hãy tìm tổng của một dãy con thịnh vượng của dãy đã cho.

**Ví dụ:** Cho dãy 5, 3, 7, -9. Một dãy con thịnh vượng có các phần tử là 5, 3, 7. Khi đó, tổng của dãy con thịnh vượng là  $S = 5 - 3 + 7 = 9$  là tổng các phần tử liên tiếp lớn nhất.

**Dữ liệu đầu vào:** Gồm hai dòng:

- Dòng đầu tiên chứa một số nguyên dương  $n$  ( $1 \leq n \leq 10^6$ ) là số lượng phần tử của dãy.
- Dòng thứ hai chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ ), các số trên cùng dòng viết cách nhau một dấu cách.

**Dữ liệu đầu ra:** Gồm một số nguyên duy nhất là tổng của dãy con thịnh vượng lớn nhất trong dãy đã cho.

- Subtask 1 (50%):  $n \leq 100$ .
- Subtask 2 (30%):  $100 < n \leq 5000$ .
- Subtask 3 (20%):  $5000 < n \leq 10^6$ .

**Ví dụ 1:**

Input
4
8 -2 7 -17
Output
13

**Ví dụ 2:**

Input
3
2 1 -9
Output
3

**Ví dụ 3**

## Input

3  
-5 4 -9

## Output

4

**Hướng dẫn giải:** Để giải bài toán này, ta có thể sử dụng thuật toán Kadane để tìm dãy con thịnh vượng lớn nhất. Thuật toán này sẽ duyệt qua từng phần tử của dãy và tính tổng của dãy con thịnh vượng hiện tại, đồng thời cập nhật tổng lớn nhất nếu cần.

**Solution C++:**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7
8     vector<int> a(n);
9     for (int i = 0; i < n; i++) {
10         cin >> a[i];
11     }
12
13     int max_sum = 0, current_sum = 0;
14     for (int i = 0; i < n; i++) {
15         current_sum += a[i];
16         if (current_sum < 0) {
17             current_sum = 0;
18         }
19         max_sum = max(max_sum, current_sum);
20     }
21
22     cout << max_sum << endl;
23
24     return 0;
25 }
```

## 5.4 Bài 4: Mã mặt hàng

**Đề bài:** Trong một hệ thống quản lý mặt hàng của một siêu thị, mã mặt hàng được lưu trữ dưới dạng một chuỗi ký tự hỗn hợp chỉ gồm các chữ cái (in hoa hoặc in thường) và chữ số (các số có mặt trong mã mặt hàng không vượt quá  $10^{255}$ ).

Ví dụ, một mã mặt hàng có thể là 789Abc123xyZ456deF789acb1235656.

Hệ thống quản lý mặt hàng của siêu thị cần tìm ra số lớn nhất xuất hiện trong mã mặt hàng này để phục vụ công tác phân tích và quản lý của siêu thị.

**Yêu cầu:** Bằng khả năng lập trình của mình em hãy giúp siêu thị thực hiện yêu cầu trên.



**Dữ liệu đầu vào:** Gồm một dòng duy nhất chứa một xâu ký tự  $S$  ( $1 \leq |S| \leq 10^6$ ) là mã mật hàng cần phân tích. Dữ liệu đảm bảo rằng trong xâu  $S$  đảm bảo luôn có chữ số.

**Dữ liệu đầu ra:** Gồm một dòng duy nhất là số lớn nhất xuất hiện trong mã mật hàng  $S$ .

**Ràng buộc dữ liệu:**

- Có 50% số test tương ứng với 50% số điểm của bài có độ dài xâu  $S$  không quá  $255$  ký tự và số xuất hiện trong xâu không quá  $10^9$ .
- Có 40% số test tương ứng với 40% số điểm của bài có độ dài xâu  $S$  không quá  $10^4$  ký tự và số xuất hiện trong xâu không quá  $10^{18}$ .
- Có 10% số test tương ứng với 10% số điểm của bài có độ dài xâu  $S$  không quá  $10^6$  ký tự và số xuất hiện trong xâu không quá  $10^{255}$ .

**Ví dụ 1:**

Input

789Abc123xyZ456deF789acb1235656

Output

1235656

**Ví dụ 2:**

Input

789aBc0004578978Xyz456Def789aCb1235

Output

4578978

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

- Duyệt qua từng ký tự trong xâu  $S$  và tách các chuỗi số liên tiếp.
- So sánh các chuỗi số đã tách để tìm ra chuỗi số lớn nhất.
- In ra chuỗi số lớn nhất tìm được.

**Solution C++:**

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     string S;
5     cin >> S;
6
7     string max_num = "";
8     string current_num = "";
9 }
```

```

10     for (char c : S) {
11         if (isdigit(c)) {
12             current_num += c;
13         } else {
14             if (current_num.length() > max_num.length() ||
15                 (current_num.length() == max_num.length() &&
16                    current_num > max_num)) {
17                 max_num = current_num; //
18             }
19             current_num = "";
20         }
21     }
22     if (!current_num.empty()) {
23         if (current_num.length() > max_num.length() ||
24             (current_num.length() == max_num.length() &&
25                current_num > max_num)) {
26             max_num = current_num;
27         }
28     }
29     cout << max_num << endl;
30
31     return 0;
32 }

```

## 6 ĐỀ HSG 9 THCS TỈNH HÀ TĨNH 2024 - 2025

### 6.1 Bài 1: Số nguyên dương $k$

**Đề bài:** Cho một số nguyên dương  $n (n \leq 10^{18})$ .

**Yêu cầu:** Hãy tìm số nguyên dương  $k$  lớn nhất thỏa mãn điều kiện  $1 + 2 + 3 + \dots + k \leq n$ .

**Dữ liệu đầu vào:** Gồm một dòng duy nhất chứa một số nguyên dương  $n$ .

**Dữ liệu đầu ra:** Gồm một số nguyên dương  $k$  thỏa mãn yêu cầu bài toán.

**Ràng buộc dữ liệu:**

- Subtask 1 (80%):  $n \leq 10^6$ .
- Subtask 2 (20%): Không có ràng buộc gì thêm.

**Ví dụ 1:**

Input

5

Output

2

**Giải thích:** Với  $n = 5$  thì giá trị  $k = 2$  là lớn nhất thỏa mãn  $1 + 2 \leq 5$ .

**Ví dụ 2:**

Input
6
Output
3

**Giải thích:** Với  $n = 6$  thì giá trị  $k = 3$  là lớn nhất thỏa mãn  $1 + 2 + 3 \leq 6$ .

**Hướng dẫn giải:**

**Solution C++:**

**6.2 Bài 2: Nuôi cá cảnh**

**Đề bài:** BigZero có một bể cá với đàn cá nhiều màu sắc. Hằng ngày sau những giờ học bài, cậu thường ngồi ngắm đàn cá và cho chúng ăn. Thức ăn của cá được đựng trong các gói đóng sẵn. Mỗi ngày đàn cá ăn hết đúng 3 gói, giá bán thức ăn thường xuyên biến động. Cửa hàng cho biết trước giá bán trong  $n$  ngày lần lượt là  $a_1, a_2, \dots, a_n$  mỗi ngày được mua nhiều gói với giá bán của ngày đó, thức ăn thừa có thể được dùng cho các ngày tiếp theo. BigZero đang lên kế hoạch để mua thức ăn cho đàn cá trong  $n$  ngày sao cho tiết kiệm nhất

**Yêu cầu:** Cho một số nguyên dương  $n$  và các số nguyên dương  $a_1, a_2, \dots, a_n$  trong đó  $a_i$  là giá bán một gói thức ăn trong ngày thứ  $i$  ( $1 \leq i \leq n \leq 10^6; a_i \leq 10^9$ ). Hãy xác định số tiền tối thiểu để mua thức ăn cho đàn cá trong  $n$  ngày.

**Dữ liệu đầu vào:** Gồm hai dòng:

- Dòng thứ nhất chứa một số nguyên dương  $n$  ( $1 \leq n \leq 10^6$ ).
- Dòng thứ hai chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $1 \leq i \leq n; a_i \leq 10^9$ ).

**Dữ liệu đầu ra:** Gồm một số nguyên duy nhất là số tiền tối thiểu để mua thức ăn cho đàn cá trong  $n$  ngày.

**Ràng buộc dữ liệu:**

- Subtask 1 (30%):  $a_1 \leq a_2 \leq \dots \leq a_n$ .
- Subtask 2 (30%):  $a_1 \geq a_2 \geq \dots \geq a_n$ .
- Subtask 3 (40%): Không có ràng buộc gì thêm.

**Ví dụ 1:**

Input
3 2 3 5
Output
18

**Giải thích:** Kế hoạch mua thức ăn là:

- Ngày 1 mua 9 gói với giá là 2.
- Ngày 2, 3 không mua gói nào.

Số tiền tối thiểu để mua thức ăn là  $9 \times 2 + 0 \times 3 + 0 \times 5 = 18$ .

**Ví dụ 2:**

Input
3
5 3 2

Output
30

**Giải thích:** Kế hoạch mua thức ăn là:

- Ngày 1 mua 3 gói với giá là 5.
- Ngày 2 mua 3 gói với giá là 3.
- Ngày 3 mua 3 gói với giá là 2.

Số tiền tối thiểu để mua thức ăn là  $3 \times 5 + 3 \times 3 + 3 \times 2 = 30$ .

**Ví dụ 3:**

Input
3
5 2 3

Output
27

**Giải thích:** Kế hoạch mua thức ăn là:

- Ngày 1 mua 3 gói với giá là 5.
- Ngày 2 mua 6 gói với giá là 2.
- Ngày 3 không mua gói nào.

Số tiền tối thiểu để mua thức ăn là  $3 \times 5 + 6 \times 2 + 0 \times 3 = 27$ .

**Hướng dẫn giải:**

**Solution C++:**

### 6.3 Bài 3: Số nguyên tố

**Đề bài:** Số nguyên tố là số tự nhiên lớn hơn 1 và chỉ có đúng hai ước là 1 và chính nó. Ví dụ các số tự nhiên 2, 3, 5, 7, 11, 13, 17, 19, 23, ... là các số nguyên tố.

**Yêu cầu:** Cho số tự nhiên  $n$ , hãy tìm số tự nhiên  $p$  thỏa mãn điều kiện  $p$  là số nguyên tố nhỏ nhất và  $p \geq n$ .

**Dữ liệu đầu vào:** Gồm hai dòng:

- Dòng thứ nhất chứa số nguyên dương  $Q (Q \leq 10^6)$  là số bộ test.
- $Q$  dòng tiếp theo, mỗi dòng chứa một số tự nhiên  $n (n \leq 10^9)$ .

**Dữ liệu đầu ra:** Gồm  $Q$  dòng, mỗi dòng ghi một số nguyên tố tìm được tương ứng với Dữ liệu đầu vào.

**Ràng buộc dữ liệu:**

- Có 30% số test ứng với 30% số điểm của bài thỏa mãn:  $Q = 1, n \leq 10^3$ ;
- Có 40% số test khác ứng với 40% số điểm của bài thỏa mãn:  $Q \leq 10^2, n \leq 10^9$ ;
- 30% số test còn lại ứng với 30% số điểm của bài thỏa mãn:  $Q \leq 10^6, n \leq 10^6$ .

**Ví dụ 1:**

Input
2
5
8
Output
5
11

**Giải thích:**

- Với  $n = 5$ , số nguyên tố nhỏ nhất  $p \geq n$  là 5.
- Với  $n = 8$ , số nguyên tố nhỏ nhất  $p \geq n$  là 11.

**Hướng dẫn giải:**

**Solution C++:**

## 6.4 Bài 4: Dãy con

**Đề bài:** Cho một dãy  $a$  gồm  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  và một số nguyên dương  $m$ .

**Yêu cầu:** Hãy tìm số nguyên dương  $L$  nhỏ nhất sao cho tất cả các dãy con gồm  $L$  phần tử liên tiếp của dãy  $a$  đều có tổng lớn hơn hoặc bằng  $m$ .

**Dữ liệu đầu vào:** Gồm hai dòng:

- Dòng thứ nhất chứa hai số nguyên dương  $n$  và  $m (1 \leq n \leq 10^6; m \leq 10^{18})$ .
- Dòng tiếp theo chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n (1 \leq i \leq n; a_i \leq 10^9)$ .

**Dữ liệu đầu ra:** Gồm một số nguyên dương  $L$  nhỏ nhất tìm được thỏa mãn yêu cầu bài toán. Nếu không tìm được giá trị thỏa mãn thì ghi  $-1$ .

**Ràng buộc dữ liệu:**

- Subtask 1 (30%):  $a_1 \leq a_2 \leq \dots \leq a_n$ .
- Subtask 2 (40%):  $n \leq 10^3$ .

- Subtask 3 (30%): Không có ràng buộc gì thêm.

**Ví dụ 1:**

Input
5 6 3 2 1 4 5
Output
3

**Ví dụ 2:**

Input
4 16 7 1 2 5
Output
-1

**Hướng dẫn giải:**

**Solution C++:**

## 7 ĐỀ HSG 9 THCS TỈNH VĨNH PHÚC 2024 - 2025

### 7.1 Bài 1: Quân Hậu

**Đề bài:** Huy là một học sinh yêu thích cờ vua, toán học và lập trình. Huy biết rằng quân cờ mạnh nhất trên bàn cờ vua là quân Hậu, vì nó có thể di chuyển như quân Xe (trên cùng một cột hoặc một hàng) và như quân Tượng (theo đường chéo).

**Yêu cầu:** Huy có một bàn cờ hình chữ nhật kích thước  $N \times M$ . Huy muốn biết nếu đặt một quân Hậu lên bàn cờ này thì số lượng ô tối đa mà nó có thể kiểm soát là bao nhiêu. Chẳng hạn, nếu  $N = M = 8$  thì một quân Hậu có thể kiểm soát tối đa 27 ô (không tính ô đặt quân Hậu, xem giải thích test ví dụ 1).

**Dữ liệu đầu vào:** Gồm hai dòng:

- Dòng 1: số nguyên  $N (1 \leq N \leq 10^9)$  là kích thước bàn cờ theo chiều dọc.
- Dòng 2: số nguyên  $M (1 \leq M \leq 10^9)$  là kích thước bàn cờ theo chiều ngang.

**Dữ liệu đầu ra:** Gồm một số nguyên là số lượng ô tối đa mà quân Hậu có thể kiểm soát trên bàn cờ kích thước  $N \times M$ .

**Ràng buộc dữ liệu:**

- Subtask 1 (42%):  $N, M \leq 10$ .
- Subtask 2 (38%):  $N, M \leq 500$ .
- Subtask 3 (20%): Không có ràng buộc gì thêm.

**Ví dụ 1:**

Input
8
8

Output
27

**Giải thích:**

x				x			
	x			x			x
		x		x		x	
			x	x	x		
x	x	x	x	Q	x	x	x
			x	x	x		
		x		x		x	
	x			x			x

**Ví dụ 2:**

Input
3
4

Output
9

**Giải thích:**

x	x	x	
x	Q	x	x
x	x	x	

Hướng dẫn giải:  
Solution C++:

## 7.2 Bài 2: Trung vị lớn nhất

**Đề bài:** Trung vị của một dãy số  $X = (x_1, x_2, \dots, x_N)$  được xác định như sau:

- Xét dãy  $Y = (y_1, y_2, \dots, y_N)$  là kết quả của việc sắp xếp dãy  $X$  theo thứ tự không giảm;
- Nếu  $N = 2k$  trung vị dãy  $X$  là  $y_k$ , nếu  $N = 2k + 1$ , trung vị của dãy  $X$  là  $y_{k+1}$ .

Chẳng hạn, trung vị của dãy  $X = (3, 1, 2, 4)$  là 2, trung vị của dãy  $X = (1, 3, 2, 3, 5)$  là 3.

Huy có một dãy số  $A = (a_1, a_2, \dots, a_N)$ . Huy muốn biến đổi dãy số về dạng dãy hằng (dãy có tất cả các phần tử bằng nhau) bằng cách sử dụng số lần phép biến đổi:

- Chọn hai chỉ số  $l$  và  $r$  ( $1 \leq l < r \leq N$ ), gọi  $x$  là trung vị của đoạn con  $(a_l, a_{l+1}, \dots, a_r)$ ;
- Gán tất cả các phần tử  $a_l, a_{l+1}, \dots, a_r$  thành  $x$ .

Chẳng hạn, nếu  $A = (1, 3, 5, 2, 4)$ , thực hiện biến đổi trên với  $l = 3$  và  $r = 4$  thì dãy trở thành  $A = (1, 3, 2, 2, 4)$ .

**Yêu cầu:** Hãy giúp Huy xác định giá trị lớn nhất của phần tử dãy hằng có thể nhận được từ dãy  $A$ .

**Dữ liệu đầu vào:** Gồm hai dòng:

- Dòng 1: số nguyên  $N$  ( $2 \leq N \leq 10^5$ );
- Dòng 2:  $N$  số nguyên  $a_1, a_2, \dots, a_N$  ( $1 \leq a_i \leq 10^9, \forall i = 1 \dots N$ ).

**Dữ liệu đầu ra:** Gồm một số nguyên ra kết quả

**Ràng buộc dữ liệu:**

- Subtask 1 (30%):  $2 \leq N \leq 10^2; 1 \leq a_i \leq 10^5, \forall i$ ;
- Subtask 2 (30%):  $10^2 \leq N \leq 10^3; 10^5 \leq a_i \leq 10^6, \forall i$ ;
- Subtask 3 (40%): Không có ràng buộc gì thêm.

**Ví dụ:**

Input
5
1 2 3 4 5
Output
4

**Giải thích:** Có thể thực hiện 3 phép biến đổi sau:

- $(l, r) = (4, 5)$  thì dãy mới  $A = [1, 2, 3, 4, 4]$
- $(l, r) = (3, 5)$  thì dãy mới  $A = [1, 2, 4, 4, 4]$
- $(l, r) = (1, 5)$  thì dãy mới  $A = [4, 4, 4, 4, 4]$

**Hướng dẫn giải:**

**Solution C++:**



### 7.3 Bài 3: Xâu rút gọn

**Đề bài:** Một xâu  $A$  được gọi là rút gọn của xâu  $B$  nếu ta có thể tạo ra  $A$  bằng cách xóa đi 0 hoặc nhiều ký tự trong  $B$  mà không thay đổi thứ tự các ký tự còn lại. Theo định nghĩa này, một xâu luôn là xâu rút gọn của chính nó.

Chẳng hạn:

- $ac$ ,  $ab$ ,  $aa$  là các xâu rút gọn của  $aabc$ ;
- $d$ ,  $aaa$ ,  $ba$  không phải là xâu rút gọn của  $aabc$ .

**Yêu cầu:** Cho hai xâu  $S$  và  $T$  chỉ gồm các ký tự chữ cái thường trong bảng chữ cái tiếng Anh. Gọi  $T^n$  là xâu được tạo ra bằng cách nối  $n$  xâu  $T$  lại với nhau. Hãy tìm giá trị nhỏ nhất của  $n$  sao cho  $S$  là một xâu rút gọn của xâu  $T^n$ .

**Dữ liệu đầu vào:** Gồm hai dòng:

- Dòng 1: xâu  $S$  với độ dài  $|S|$  ( $1 \leq |S| \leq 10^6$ ).
- Dòng 2: xâu  $T$  với độ dài  $|T|$  ( $1 \leq |T| \leq 10^5$ ).

**Dữ liệu đầu ra:** Gồm một số nguyên là giá trị  $n$  nhỏ nhất sao cho  $S$  là xâu rút gọn của  $T$ . Nếu không tồn tại giá trị  $n$  như vậy thì in ra  $-1$ .

**Ràng buộc dữ liệu:**

- Subtask 1 (8%):  $S$  và  $T$  chỉ chứa ký tự  $a$ .
- Subtask 2 (13%):  $|S|, |T| \leq 100$ .
- Subtask 3 (21%):  $|S| \leq 10^4, |T| \leq 100$ .
- Subtask 4 (34%):  $|T| \leq 1000$ .
- Subtask 5 (24%): Không có ràng buộc bổ sung.

**Ví dụ 1:**

Input
caa
ac
Output
3

**Giải thích:** Ta có:  $T^1 = T = ac$ ,  $T^2 = T = acac$ ,  $T^3 = T = acacac$ ;  $n = 3$  là giá trị nhỏ nhất để xâu  $S$  trở thành xâu rút gọn của  $T^n$ .

**Ví dụ 2:**

Input
cab
acca

## Output

-1

**Giải thích:** Không tìm được  $n$  thỏa mãn điều kiện.

**Hướng dẫn giải:**

**Solution C++:**

## 7.4 Bài 4: Dãy đẹp

**Đề bài:** Cho dãy  $A = (a_1, a_2, \dots, a_n)$ . Độ đẹp của dãy  $A$  được định nghĩa là tổng lớn nhất của một đoạn con liên tiếp (có thể rỗng) của dãy. Chẳng hạn, dãy  $A = (-3, 8, 4, -2, 12)$  có độ đẹp bằng 22 (đoạn con  $(8, 4, -2, 12)$ ), dãy  $B = (-1, -2, -3, -4, -5)$  có độ đẹp bằng 0 (đoạn con rỗng).

**Yêu cầu:** Để gia tăng độ đẹp của dãy  $A$ , bạn được phép chọn tối đa một đoạn con liên tiếp của dãy và nhân từng phần tử trong đoạn con đó lên  $X$  lần. Xác định độ đẹp lớn nhất có thể đạt được của dãy.

**Dữ liệu đầu vào:** Gồm hai dòng:

- Dòng 1: Hai số nguyên  $N, X$  ( $1 \leq N \leq 4 \times 10^5; -100 \leq X \leq 100$ );
- Dòng 2:  $N$  số nguyên  $a_1, a_2, \dots, a_N$  ( $|a_i| \leq 10^9$ ).

**Dữ liệu đầu ra:** Gồm một số nguyên là độ đẹp tối đa của dãy  $A$  sau khi thực hiện không quá một thao tác nói trên.

**Ràng buộc dữ liệu:**

- 20% số điểm dành cho các test có  $1 \leq N \leq 50$ ;
- 30% số điểm dành cho các test có  $1 \leq N \leq 300$ ;
- 20% số điểm dành cho các test có  $a_i \geq 0$  với mọi  $i$ ;
- 30% số điểm còn lại không có ràng buộc bổ sung.

**Ví dụ 1:**

## Input

```
5 -2
-3 8 -2 1 -6
```

## Output

22

**Giải thích:** Thực hiện thao tác với đoạn  $[-2, 1, -6]$  thu được dãy  $[-3, 8, 4, -2, 12]$ . Dãy này có độ đẹp là 22. Đây là độ đẹp lớn nhất có thể đạt được.

**Ví dụ 2:**

## Input

```
8 -4
1 2 1 1 2 0 0 7
```

**Output**

14

**Giải thích:** Không cần thực hiện thao tác nào.**Ví dụ 3:****Input**5 10  
-1 -2 -3 -4 -5**Output**

0

**Hướng dẫn giải:****Solution C++:**