

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

BỘ ĐỀ NÂNG CAO NĂNG LỰC LẬP TRÌNH TIN HỌC CẤP THCS

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Số Fibonacci lớn nhất	FIBO*	Bàn phím	Màn hình	4
Bài 2	Phép nhân lớn nhất	MUL*	Bàn phím	Màn hình	5
Bài 3	Tìm ký tự lặp	CHAR*	Bàn phím	Màn hình	5
Bài 4	Tổng chữ số lẻ và chiều dài	ODD*	Bàn phím	Màn hình	6

Bài 1: Số Fibonacci lớn nhất

Cho một dãy số nguyên dương. Hãy tìm số lớn nhất trong dãy đó là số Fibonacci.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output

- In ra số lớn nhất trong dãy là số Fibonacci hoặc -1 nếu không có số nào là số Fibonacci.

Ví dụ:

Input	Output
5 10 20 34 50 89	34

Bài 2: Phép nhân lớn nhất

Cho dãy số nguyên dương. Tìm phép nhân lớn nhất của hai số khác nhau trong dãy.

Input

- Dòng đầu tiên chứa số nguyên dương n ($2 \leq n \leq 10^6$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output

- In ra phép nhân lớn nhất của hai số khác nhau hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
4 10 20 5 5	100

Bài 3: Tìm ký tự lặp

Cho một chuỗi ký tự S . Hãy tìm ký tự đầu tiên lặp lại nhiều hơn một lần trong chuỗi.

Input

- Dòng đầu tiên là chuỗi ký tự S có độ dài không quá 10^5 ký tự, chỉ gồm các chữ cái latin thường.

Output

- In ra ký tự đầu tiên lặp lại nhiều hơn một lần hoặc "No character repeats" nếu không tìm thấy.

Ví dụ:

Input	Output
programming	r

Bài 4. Tổng chữ số lẻ và chiều dài

Cho một chuỗi ký tự chứa các số và chữ cái. Tính tổng các chữ số lẻ và ghép với chiều dài chuỗi.

Input:

- Một dòng duy nhất chứa chuỗi ký tự có độ dài không quá 10^5 ký tự.

Output:

- In ra tổng các chữ số lẻ và chiều dài chuỗi.

Ví dụ:

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

Input	Output
abc1234	4 7

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Số Fibonacci lớn nhất

1. Phân tích giải thuật:

1. Xây dựng một tập hợp các số Fibonacci có giá trị đến giới hạn tối đa (10^9).
2. Duyệt qua dãy số và kiểm tra xem số nào thuộc tập hợp Fibonacci.
3. Giữ lại số Fibonacci lớn nhất tìm được.

2. Code tham khảo:

```
def is_fibo(n):  
    a, b = 0, 1  
    while b <= n:  
        if b == n:  
            return True  
        a, b = b, a + b  
    return False  
  
n = int(input())  
arr = list(map(int, input().split()))  
max_fibo = -1  
  
for num in arr:  
    if is_fibo(num):  
        max_fibo = max(max_fibo, num)  
  
print(max_fibo)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm số Fibonacci nhỏ nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số nhỏ nhất trong dãy là số Fibonacci hoặc -1 nếu không có số nào là số Fibonacci.

Ví dụ:

Input	Output
6 10 8 34 21 50 89	8

Bài 1.2: Kiểm tra số là Fibonacci hay không

Input:

- Một số nguyên dương n ($1 \leq n \leq 10^9$).

Output:

- "YES" nếu số đó là Fibonacci, "NO" nếu không phải.

Ví dụ:

Input	Output
21	YES
15	NO

Bài 1.3: Đếm số Fibonacci trong dãy

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số lượng số Fibonacci trong dãy.

Ví dụ:

Input	Output
6 1 2 3 5 7 13	4

Bài 2: Phép nhân lớn nhất

1. Phân tích giải thuật:

1. Sắp xếp dãy số theo thứ tự giảm dần.

2. Phép nhân lớn nhất sẽ là phép nhân của hai số đầu tiên trong dãy.

2. Code tham khảo

```
n = int(input())
arr = sorted(map(int, input().split()),
reverse=True)
max_product = arr[0] * arr[1] if n > 1 else -1
print(max_product)
```

3. Bài tập mở rộng:

Bài 2.1 : Tìm phép cộng lớn nhất của hai số

Input:

- Dòng đầu tiên chứa số nguyên dương n ($2 \leq n \leq 10^6$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra phép cộng lớn nhất của hai số khác nhau trong dãy.

Ví dụ:

Input	Output
4 10 20 5 5	30

Bài 2.2 : Tìm hiệu lớn nhất của hai số

Input:

- Dòng đầu tiên chứa số nguyên dương n ($2 \leq n \leq 10^6$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra hiệu lớn nhất của hai số khác.

Ví dụ:

Input	Output
5 20 10 5 7 3	15

Bài 2.3 : Tìm số lớn nhất bằng tích của ba số

Input:

- Dòng đầu tiên chứa số nguyên dương n ($3 \leq n \leq 10^6$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra phép nhân lớn nhất của ba số khác nhau trong dãy hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
6 10 20 30 5 4 2	1200

Bài 3: Tìm ký tự lặp

1. Phân tích giải thuật:

1. Duyệt qua từng ký tự trong xâu.
2. Sử dụng một tập hợp để theo dõi các ký tự đã gặp.
3. Nếu một ký tự xuất hiện lần thứ hai, in ký tự đó và dừng.

2. Code tham khảo

```
python
Sao chép mã
s = input()
seen = set()
for char in s:
    if char in seen:
        print(char)
        break
    seen.add(char)
else:
    print("No character repeats")
```

3. Bài tập mở rộng:

Bài 3.1: Tìm ký tự không lặp đầu tiên

Input:

- Dòng đầu tiên là chuỗi ký tự S có độ dài không quá 10^5 ký tự.

Output:

- In ra ký tự đầu tiên chỉ xuất hiện một lần hoặc "No unique character" nếu không tìm thấy.

Ví dụ:

Input	Output
abcbabcde	d

Bài 3.2: Đếm số lần lặp lại của một ký tự cho trước

Input:

- Dòng đầu tiên là chuỗi ký tự S có độ dài không quá 10^5 ký tự.
- Dòng thứ hai là một ký tự cần đếm.

Output:

- In ra số lần ký tự xuất hiện trong chuỗi.

Ví dụ:

Input	Output
Aabbcbde a	2

Bài 3.3: Kiểm tra xem chuỗi có chứa tất cả các chữ cái từ 'a' đến 'z'

Input:

- Dòng đầu tiên là chuỗi ký tự S có độ dài không quá 10^5 ký tự.

Output:

- "YES" nếu chuỗi chứa tất cả các chữ cái từ 'a' đến 'z', "NO" nếu không.

Ví dụ:

Input	Output
thequickbrownfoxjumpsoverthelazydog	YES

Bài 4: Tổng chữ số lẻ và chiều dài

1. Phân tích giải thuật

1. Tách các chữ số ra từ xâu.
2. Tính tổng các chữ số lẻ.
3. In ra tổng kèm với chiều dài xâu.

2. Code tham khảo

```
s = input()
odd_sum = sum(int(char) for char in s if
char.isdigit() and int(char) % 2 != 0)
print(odd_sum, len(s))
```

3. Bài tập mở rộng:

Bài 4.1: Tính tổng các chữ số chẵn và số lượng ký tự đặc biệt

Input:

- Một dòng duy nhất chứa xâu ký tự có độ dài không quá 10^5 ký tự.

Output:

- In ra tổng các chữ số chẵn và số lượng ký tự đặc biệt trong xâu.

Ví dụ:

Input	Output
a!b@c#1234	6 3

Bài 4.2: Tìm tổng các số nguyên tố và số ký tự thường

Input:

- Một dòng duy nhất chứa xâu ký tự có độ dài không quá 10^5 ký tự.

Output:

- In ra tổng các số nguyên tố và số lượng ký tự thường trong xâu.

Ví dụ:

Input	Output
Hello1237	10 5

Bài 4.3: Tính tổng các số chia hết cho 5 và số khoảng trắng

Input:

- Một dòng duy nhất chứa xâu ký tự có độ dài không quá 10^5 ký tự.

Output:

- In ra tổng các số chia hết cho 5 và số lượng khoảng trắng trong xâu.

Ví dụ:

Input	Output
15 6 14 8 5 25	45 2

ĐỀ SỐ 02

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Đếm số nguyên tố trong dãy	PRIME*	Bàn phím	Màn hình	4
Bài 2	Tìm dãy con có tổng lớn nhất	MAXSUB*	Bàn phím	Màn hình	5
Bài 3	Đếm từ trong xâu	WORDS*	Bàn phím	Màn hình	5
Bài 4	Số lớn nhất có thể tạo được	MAXNUM*	Bàn phím	Màn hình	6

Bài 1: Đếm số nguyên tố trong dãy

Cho một dãy số nguyên dương. Hãy đếm số lượng số nguyên tố có trong dãy.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output

- In ra số lượng số nguyên tố trong dãy.

Ví dụ:

Input	Output
5 10 3 5 8 14	2

Bài 2: Tìm dãy con có tổng lớn nhất

Cho một dãy số nguyên (có thể có số âm). Hãy tìm dãy con liên tiếp có tổng lớn nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output

- In ra tổng lớn nhất của dãy con liên tiếp có tổng lớn nhất.

Ví dụ:

Input	Output
8 -2 1 -3 4 -1 2 1 -5 4	10

Bài 3: Đếm từ trong chuỗi

Cho một chuỗi ký tự. Hãy đếm số lượng từ trong chuỗi, từ được định nghĩa là các ký tự không phải khoảng trắng liên tiếp.

Input

- Một dòng duy nhất chứa chuỗi ký tự (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng từ trong chuỗi.

Ví dụ:

Input	Output
Hello world!	2
Python is fun	3

Bài 4. Số lớn nhất có thể tạo được

Cho một dãy số nguyên dương. Hãy sắp xếp các số này thành một số lớn nhất có thể tạo được.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số lớn nhất có thể tạo được bằng cách sắp xếp các số trong dãy.

Ví dụ:

Input	Output
3	95321
3 5 9 21	

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Số Fibonacci lớn nhất

1. Phân tích giải thuật:

- Sử dụng thuật toán kiểm tra số nguyên tố hiệu quả (như Sàng Eratosthenes hoặc kiểm tra chia từ 2 đến căn bậc hai của số).
- Duyệt qua dãy số và đếm số lượng số nguyên tố.

2. Code tham khảo:

```
import math

def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
    return True

n = int(input())
arr = list(map(int, input().split()))
count_prime = sum(1 for num in arr if is_prime(num))
print(count_prime)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm số nguyên tố lớn nhất trong dãy

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số nguyên tố lớn nhất trong dãy hoặc -1 nếu không có số nguyên tố.

Ví dụ:

Input	Output
6 4 6 8 10 13 7	13

Bài 1.2: Kiểm tra dãy số có phải là dãy số nguyên tố hay không

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- "YES" nếu tất cả các số trong dãy đều là số nguyên tố, "NO" nếu có ít nhất một số không phải.

Ví dụ:

Input	Output
4 7 13 19 20	NO

Bài 1.3: Đếm số lượng số nguyên tố trong dãy con liên tiếp có độ dài lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số lượng số nguyên tố trong dãy con liên tiếp có độ dài lớn nhất mà tất cả các số đều là số nguyên tố.

Ví dụ:

Input	Output
7 2 3 5 7 4 6 11 13	3

Bài 2: Tìm dãy con có tổng lớn nhất

1. Phân tích giải thuật:

1. Sử dụng thuật toán Kadane's để tìm tổng lớn nhất của dãy con liên tiếp.
2. Khởi tạo biến lưu trữ tổng lớn nhất và tổng hiện tại, duyệt qua dãy số để cập nhật.

2. Code tham khảo

```
n = int(input())
arr = list(map(int, input().split()))

max_sum = current_sum = arr[0]

for i in range(1, n):
    current_sum = max(arr[i], current_sum + arr[i])
    max_sum = max(max_sum, current_sum)

print(max_sum)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm dãy con liên tiếp có tổng nhỏ nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng nhỏ nhất của dãy con liên tiếp có tổng nhỏ nhất.

Ví dụ:

Input	Output
5 3 -1 -4 1 -5	-5

Bài 2.2: Tìm dãy con có tổng lớn nhất không chứa số âm

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp không chứa số âm.

Ví dụ:

Input	Output
6 -2 3 4 -1 2 1	10

Bài 2.3: Tìm dãy con có tổng lớn nhất với độ dài tối thiểu k

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp có độ dài ít nhất là k.

Ví dụ:

Input	Output
7 3 -1 3 2 -2 4 -1 5	6

Bài 3: Đếm từ trong xâu

1. Phân tích giải thuật:

1. Sử dụng phương pháp tách xâu dựa trên khoảng trắng để đếm số từ.
2. Bỏ qua các khoảng trắng thừa ở đầu và cuối xâu.

2. Code tham khảo

```
s = input().strip()
words = s.split()
print(len(words))
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng ký tự không phải khoảng trắng trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng ký tự không phải khoảng trắng trong xâu.

Ví dụ:

Input	Output
Hello world!	10
Python is fun	12

Bài 3.2: Đếm số lần xuất hiện của một từ cho trước

Input:

- Dòng đầu tiên chứa xâu ký tự (độ dài không quá 10^5 ký tự).
- Dòng thứ hai chứa từ cần đếm.

Output:

- In ra số lần từ xuất hiện trong xâu.

Ví dụ:

Input	Output
Python is fun, fun fun fun	3

Bài 3.3: Tìm từ dài nhất trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất trong xâu, nếu có nhiều từ cùng độ dài thì in ra từ xuất hiện trước.

Ví dụ:

Input	Output
Programming is fun	Programming

Bài 4: Số lớn nhất có thể tạo được

1. Phân tích giải thuật

1. Sử dụng phương pháp sắp xếp tùy chỉnh, dựa trên việc so sánh từng cặp số nếu nối chúng theo thứ tự nào sẽ tạo ra giá trị lớn hơn

2. Sau khi sắp xếp, nối các số lại với nhau thành một xâu số lớn nhất.

2. Code tham khảo

```
from functools import cmp_to_key

def compare(x, y):
    if x + y > y + x:
        return -1
    else:
        return 1

n = int(input())
arr = input().split()
arr.sort(key=cmp_to_key(compare))
print("".join(arr))
```

3. Bài tập mở rộng:

Bài 4.1: Số nhỏ nhất có thể tạo được

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số nhỏ nhất có thể tạo được bằng cách sắp xếp các số trong dãy.

Ví dụ:

Input	Output
3	12359
9 21 3 5	

Bài 4.2: Tạo số lớn nhất có thể bằng cách thêm một số

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).
- Dòng thứ ba chứa một số nguyên dương m (mỗi số không vượt quá 10^9).

Output:

- In ra số lớn nhất có thể tạo được bằng cách thêm số m vào dãy rồi sắp xếp.

Ví dụ:

Input	Output
3	95321
3 5 9	
21	

Bài 4.3: Tạo số lớn nhất có thể từ dãy số có một số số bị trùng lặp

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số lớn nhất có thể tạo được từ dãy số với các số trùng lặp.

Ví dụ:

Input	Output
4	5553221
3 5 5 2 1	

ĐỀ SỐ 03

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tìm số xuất hiện nhiều nhất	MOSTFREQ*	Bàn phím	Màn hình	4
Bài 2	Sắp xếp số theo quy tắc đặc biệt	SPECIALSORT*	Bàn phím	Màn hình	5
Bài 3	Đếm số chuỗi con đối xứng	PALINDROME*	Bàn phím	Màn hình	5
Bài 4	Tổng các chữ số trong xâu	SUMDIGITS*	Bàn phím	Màn hình	6

Bài 1: Tìm số xuất hiện nhiều nhất

Cho một dãy số nguyên dương. Hãy tìm số xuất hiện nhiều nhất trong dãy. Nếu có nhiều số cùng xuất hiện nhiều nhất, in ra số lớn nhất trong số đó.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output

- In ra số xuất hiện nhiều nhất, nếu có nhiều số cùng xuất hiện, in ra số lớn nhất.

Ví dụ:

Input	Output
6 2 3 5 2 5 5	5

Bài 2: Sắp xếp số theo quy tắc đặc biệt

Cho một dãy số nguyên dương. Hãy sắp xếp các số theo thứ tự tăng dần nhưng các số chia hết cho 5 phải được đưa lên đầu tiên và vẫn giữ nguyên thứ tự.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output

- In ra dãy số sau khi đã sắp xếp theo quy tắc đặc biệt.

Ví dụ:

Input	Output
5 3 8 10 2 5	10 5 2 3 8

Bài 3: Đếm số chuỗi con đối xứng

Cho một xâu ký tự S . Hãy đếm số lượng chuỗi con đối xứng trong xâu S .

Input

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.

Output

- In ra số lượng chuỗi con đối xứng trong xâu S .

Ví dụ:

Input	Output
6 abbaac	9

Bài 4. Tổng các chữ số trong xâu

Cho một xâu ký tự chứa các chữ cái và chữ số. Hãy tính tổng tất cả các chữ số trong xâu.

Input:

- Một dòng duy nhất chứa xâu ký tự có độ dài không quá 10^5 ký tự.

Output:

- In ra tổng các chữ số trong xâu.

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

Ví dụ:

Input	Output
abc123xy45z	15

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tìm số xuất hiện nhiều nhất

1. Phân tích giải thuật:

1. Sử dụng một từ điển để đếm số lần xuất hiện của từng số trong dãy.
2. Duyệt qua từ điển để tìm số có số lần xuất hiện nhiều nhất và có giá trị lớn nhất nếu có nhiều số cùng số lần xuất hiện.

2. Code tham khảo:

```
from collections import defaultdict

n = int(input())
arr = list(map(int, input().split()))

freq = defaultdict(int)

for num in arr:
    freq[num] += 1

max_freq = max(freq.values())
max_num = max(num for num in freq if freq[num] == max_freq)

print(max_num)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm số xuất hiện ít nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số xuất hiện ít nhất, nếu có nhiều số cùng xuất hiện, in ra số nhỏ nhất.

Ví dụ:

Input	Output
5 1 2 3 1 2	3

Bài 1.2: Đếm số lượng số xuất hiện đúng k lần

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số lượng số xuất hiện đúng k lần.

Ví dụ:

Input	Output
6 2 4 4 5 5 6 6	2

Bài 1.3: Tìm số xuất hiện nhiều nhất trong dãy số sau khi nhân đôi các số

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số xuất hiện nhiều nhất sau khi nhân đôi tất cả các số, nếu có nhiều số cùng xuất hiện, in ra số lớn nhất.

Ví dụ:

Input	Output
4 2 3 5 5	10

Bài 2: Sắp xếp số theo quy tắc đặc biệt

1. Phân tích giải thuật:

1. Tách dãy số thành hai phần: phần chứa các số chia hết cho 5 và phần còn lại.
2. Sắp xếp cả hai phần, sau đó ghép lại với phần chia hết cho 5 ở đầu.

2. Code tham khảo

```
n = int(input())
arr = list(map(int, input().split()))

div_by_5 = [x for x in arr if x % 5 == 0]
not_div_by_5 = [x for x in arr if x % 5 != 0]

div_by_5.sort()
not_div_by_5.sort()

result = div_by_5 + not_div_by_5
print(*result)
```

3. Bài tập mở rộng:

Bài 2.1: Sắp xếp số lẻ trước, số chẵn sau

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra dãy số sau khi đã sắp xếp với số lẻ trước, số chẵn sau.

Ví dụ:

Input	Output
6	1 3 7 4 6 8
4 1 6 3 8 7	

Bài 2.2: Sắp xếp số theo thứ tự tăng dần nhưng các số là bội của 3 phải ở cuối

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra dãy số sau khi đã sắp xếp theo quy tắc đặc biệt với các số là bội của 3 phải ở cuối.

Ví dụ:

Input	Output
5 3 9 2 7 8	2 7 8 3 9

Bài 2.3: Sắp xếp số giảm dần nhưng số nguyên tố phải ở đầu

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra dãy số sau khi đã sắp xếp theo quy tắc đặc biệt với số nguyên tố phải ở đầu.

Ví dụ:

Input	Output
6 8 15 7 5 12 11	11 7 5 15 12 8

Bài 3: Đếm số chuỗi con đối xứng

1. Phân tích giải thuật:

1. Duyệt qua tất cả các chuỗi con của xâu S.
2. Kiểm tra xem chuỗi con đó có đối xứng không và đếm số lượng.

2. Code tham khảo

```
def is_palindrome(s):  
    return s == s[::-1]  
  
n = int(input())  
s = input()  
count = 0
```

```
for i in range(n):  
    for j in range(i+1, n+1):  
        if is_palindrome(s[i:j]):  
            count += 1  
  
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng chuỗi con có độ dài lẻ và đối xứng

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.

Output:

- In ra số lượng chuỗi con có độ dài lẻ và đối xứng.

Ví dụ:

Input	Output
5 ababa	7

Bài 3.2: Tìm chuỗi con đối xứng dài nhất

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.

Output:

- In ra chuỗi con đối xứng dài nhất, nếu có nhiều chuỗi cùng độ dài, in ra chuỗi xuất hiện đầu tiên.

Ví dụ:

Input	Output
7 abbacca	abba

Bài 3.3: Kiểm tra xem xâu có chứa chuỗi con đối xứng với độ dài tối thiểu k

Input:

- Dòng đầu tiên chứa số nguyên dương N và k ($1 \leq k \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.

Output:

- "YES" nếu có ít nhất một chuỗi con đối xứng có độ dài tối thiểu k, "NO" nếu không có.

Ví dụ:

Input	Output
6 4 abbaac	YES

Bài 4: Tổng các chữ số trong xâu

1. Phân tích giải thuật

1. Duyệt qua từng ký tự trong xâu.
2. Nếu ký tự là chữ số, chuyển nó thành số và cộng vào tổng.

2. Code tham khảo

```
s = input()
total = sum(int(char) for char in s if
char.isdigit())
print(total)
```

3. Bài tập mở rộng:

Bài 4.1: Tính tổng các chữ số lẻ trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự có độ dài không quá 10^5 ký tự.

Output:

- In ra tổng các chữ số lẻ trong xâu.

Ví dụ:

Input	Output
123abc456	9

Bài 4.2: Tìm chữ số lớn nhất trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự có độ dài không quá 10^5 ký tự.

Output:

- In ra chữ số lớn nhất trong xâu, nếu không có chữ số thì in ra -1.

Ví dụ:

Input	Output
ab12cd34	4

Bài 4.3: Kiểm tra xem tổng các chữ số trong xâu có phải là số nguyên tố hay không

Input:

- Một dòng duy nhất chứa xâu ký tự có độ dài không quá 10^5 ký tự.

Output:

- "YES" nếu tổng các chữ số là số nguyên tố, "NO" nếu không phải.

Ví dụ:

Input	Output
abc23def4	NO

ĐỀ SỐ 04

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tìm chuỗi con có độ dài lớn nhất với các ký tự khác nhau	UNIQUESTR*	Bàn phím	Màn hình	4
Bài 2	Tìm số nhỏ nhất chia hết cho tất cả các số trong dãy	MINMULT*	Bàn phím	Màn hình	5
Bài 3	Xóa các phần tử trùng lặp trong danh sách	REMDUP*	Bàn phím	Màn hình	5
Bài 4	Đếm số lần xuất hiện của chuỗi con trong chuỗi	SUBCNT*	Bàn phím	Màn hình	6

Bài 1: Tìm chuỗi con có độ dài lớn nhất với các ký tự khác nhau

Cho một xâu ký tự S. Hãy tìm chuỗi con có độ dài lớn nhất mà trong đó tất cả các ký tự đều khác nhau.

Input

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.

Output

- In ra độ dài của chuỗi con dài nhất mà trong đó tất cả các ký tự đều khác nhau.

Ví dụ:

Input	Output
8	4
abcadeaf	

Bài 2: Tìm số nhỏ nhất chia hết cho tất cả các số trong dãy

Cho một dãy số nguyên dương. Hãy tìm số nhỏ nhất chia hết cho tất cả các số trong dãy.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output

- In ra số nhỏ nhất chia hết cho tất cả các số trong dãy.

Ví dụ:

Input	Output
3 4 5 6	60

Bài 3: Xóa các phần tử trùng lặp trong danh sách

Cho một danh sách các số nguyên dương. Hãy xóa tất cả các phần tử trùng lặp trong danh sách và in ra danh sách mới, giữ nguyên thứ tự xuất hiện của các phần tử.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output

- In ra danh sách mới sau khi đã xóa các phần tử trùng lặp.

Ví dụ:

Input	Output
6 1 2 2 3 4 4	1 2 3 4

Bài 4. Đếm số lần xuất hiện của chuỗi con trong chuỗi

Cho một xâu ký tự S và một chuỗi con T . Hãy đếm số lần xuất hiện của T trong S .

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.
- Dòng thứ ba là chuỗi con T cần tìm.

Output:

- In ra số lần xuất hiện của T trong S .

Ví dụ:

Input	Output
11 abcdefabcgh abc	2

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tìm số xuất hiện nhiều nhất

1. Phân tích giải thuật:

1. Sử dụng kỹ thuật "Sliding Window" để duyệt qua các ký tự trong xâu.
2. Dùng một tập hợp để theo dõi các ký tự khác nhau, và cập nhật độ dài của chuỗi con dài nhất.

2. Code tham khảo:

```
n = int(input())
s = input()

max_len = 0
start = 0
seen = {}

for end in range(n):
    if s[end] in seen and seen[s[end]] >= start:
        start = seen[s[end]] + 1
    seen[s[end]] = end
    max_len = max(max_len, end - start + 1)

print(max_len)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm chuỗi con có độ dài lớn nhất mà không có ký tự nào lặp lại quá k lần

Input:

- Dòng đầu tiên chứa số nguyên dương N và k ($1 \leq N \leq 10^5$, $1 \leq k \leq 26$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.

Output:

- In ra độ dài của chuỗi con dài nhất mà không có ký tự nào lặp lại quá k lần.

Ví dụ:

Input	Output
10 2 abbaaccde	8

Bài 1.2: Tìm chuỗi con có độ dài nhỏ nhất với tất cả các ký tự khác nhau trong S

Input:

- Dòng đầu tiên là số nguyên dương N ($1 \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.

Output:

- In ra độ dài của chuỗi con ngắn nhất mà trong đó tất cả các ký tự đều khác nhau.

Ví dụ:

Input	Output
6 abcaef	4

Bài 1.3: Đếm số lượng chuỗi con có tất cả các ký tự khác nhau với độ dài lớn hơn hoặc bằng k

Input:

- Dòng đầu tiên chứa số nguyên dương N và k ($1 \leq N \leq 10^5$, $1 \leq k \leq N$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.

Output:

- In ra số lượng chuỗi con có tất cả các ký tự khác nhau với độ dài lớn hơn hoặc bằng k .

Ví dụ:

Input	Output
6 3 abcaef	7

Bài 2: Tìm số nhỏ nhất chia hết cho tất cả các số trong dãy

1. Phân tích giải thuật:

1. Sử dụng lý thuyết số học về bội chung nhỏ nhất (LCM).
2. Áp dụng LCM giữa các số trong dãy để tìm số nhỏ nhất chia hết cho tất cả các số.

2. Code tham khảo

```
import math

def lcm(a, b):
    return a * b // math.gcd(a, b)

n = int(input())
arr = list(map(int, input().split()))

result = arr[0]
for i in range(1, n):
    result = lcm(result, arr[i])

print(result)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm số lớn nhất chia hết cho tất cả các số trong dãy

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số lớn nhất chia hết cho tất cả các số trong dãy, hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
3 4 5 6	-1

Bài 2.2: Tìm số nhỏ nhất chia hết cho ít nhất một nửa số trong dãy

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).

- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số nhỏ nhất chia hết cho ít nhất một nửa số trong dãy.

Ví dụ:

Input	Output
4 3 4 6 12	12

Bài 2.3: Tìm số nhỏ nhất chia hết cho tất cả các số trong dãy sau khi tăng mỗi số lên k

Input:

- Dòng đầu tiên chứa số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số nhỏ nhất chia hết cho tất cả các số trong dãy sau khi tăng mỗi số lên k .

Ví dụ:

Input	Output
3 2 4 5 6	72

Bài 3: Xóa các phần tử trùng lặp trong danh sách

1. Phân tích giải thuật:

1. Sử dụng một tập hợp để lưu các phần tử đã gặp, duyệt qua danh sách và loại bỏ các phần tử trùng lặp.

2. Giữ nguyên thứ tự xuất hiện của các phần tử.

2. Code tham khảo

```
n = int(input())
arr = list(map(int, input().split()))

seen = set()
```

```
result = []

for num in arr:
    if num not in seen:
        seen.add(num)
        result.append(num)

print(*result)
```

3. Bài tập mở rộng:

Bài 3.1: Xóa các phần tử xuất hiện đúng k lần trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq n$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra danh sách mới sau khi đã xóa các phần tử xuất hiện đúng k lần.

Ví dụ:

Input	Output
6 2 1 2 2 3 4 4	1 3 4

Bài 3.2: Xóa tất cả các phần tử xuất hiện nhiều hơn k lần

Input:

- Dòng đầu tiên chứa số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq n$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra danh sách mới sau khi đã xóa tất cả các phần tử xuất hiện nhiều hơn k lần.

Ví dụ:

Input	Output
6 1 6 1	2 3

1 1 2 2 3 3	
-------------	--

Bài 3.3: Xóa các phần tử trùng lặp nhưng giữ lại phần tử cuối cùng trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra danh sách mới sau khi đã xóa các phần tử trùng lặp, giữ lại phần tử cuối cùng.

Ví dụ:

Input	Output
6 1 2 3 1 4 2	2 3 4

Bài 4: Đếm số lần xuất hiện của chuỗi con trong chuỗi

1. Phân tích giải thuật

1. Sử dụng phương pháp đếm xuất hiện của chuỗi con bằng cách duyệt qua xâu S.
2. Mỗi lần tìm thấy chuỗi con T trong S, tăng biến đếm lên.

2. Code tham khảo

```
s = input()
t = input()

count = s.count(t)

print(count)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm vị trí đầu tiên của chuỗi con trong chuỗi

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^5$).

- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.
- Dòng thứ ba là chuỗi con T cần tìm.

Output:

- In ra vị trí đầu tiên của T trong S, nếu không tìm thấy thì in ra -1.

Ví dụ:

Input	Output
11 abcdefabcgh abc	0

Bài 4.2: Kiểm tra xem chuỗi con T có phải là chuỗi con của S hay không

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.
- Dòng thứ ba là chuỗi con T cần tìm.

Output:

- "YES" nếu T là chuỗi con của S, "NO" nếu không phải.

Ví dụ:

Input	Output
11 abcdefabcgh abc	YES

Bài 4.3: Đếm số lần xuất hiện của chuỗi con trong chuỗi với phép xoay vòng

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^5$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái latin thường.
- Dòng thứ ba là chuỗi con T cần tìm.

Output:

- In ra số lần xuất hiện của T trong S kể cả khi T có thể xoay vòng.

Ví dụ:

Input	Output
10 abcabcabca cab	3

ĐỀ SỐ 05

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Đếm số lượng số Fibonacci	FIBCNT*	Bàn phím	Màn hình	4
Bài 2	Tìm dãy con liên tiếp có tổng nhỏ nhất	MINCUMSUM*	Bàn phím	Màn hình	5
Bài 3	Đếm số lượng từ có độ dài lẻ	ODDWORD*	Bàn phím	Màn hình	5
Bài 4	Kiểm tra danh sách tăng dần	CHECKASC*	Bàn phím	Màn hình	6

Bài 1: Đếm số lượng số Fibonacci

Cho một dãy số nguyên dương. Hãy đếm xem trong dãy có bao nhiêu số Fibonacci.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output

- In ra số lượng số Fibonacci trong dãy.

Ví dụ:

Input	Output
5 1 4 8 13 21	2

Bài 2: Tìm dãy con liên tiếp có tổng nhỏ nhất

Cho một dãy số nguyên (có thể có số âm). Hãy tìm dãy con liên tiếp có tổng nhỏ nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).

- Dòng thứ hai chứa n số nguyên.

Output

- In ra tổng nhỏ nhất của dãy con liên tiếp có tổng nhỏ nhất.

Ví dụ:

Input	Output
5 3 -4 2 -5 1	-9

Bài 3: Đếm số lượng từ có độ dài lẻ

Cho một chuỗi ký tự chứa các từ được phân cách bởi khoảng trắng. Hãy đếm số lượng từ có độ dài lẻ trong chuỗi.

Input

- Một dòng duy nhất chứa chuỗi ký tự (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng từ có độ dài lẻ trong chuỗi.

Ví dụ:

Input	Output
this is an example	2

Bài 4. Kiểm tra danh sách tăng dần

Cho một danh sách các số nguyên dương. Hãy kiểm tra xem danh sách có phải là danh sách tăng dần hay không.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra "YES" nếu danh sách là tăng dần, "NO" nếu không phải.

Ví dụ:

Input	Output
5	YES
1 2 3 4 5	

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Đếm số lượng số Fibonacci

1. Phân tích giải thuật:

1. Tạo danh sách hoặc tập hợp các số Fibonacci nhỏ hơn hoặc bằng giá trị tối đa trong dãy.
2. Kiểm tra từng phần tử trong dãy có phải là số Fibonacci không, nếu có thì tăng biến đếm.

2. Code tham khảo:

```
def generate_fibonacci(limit):  
    fib_set = set()  
    a, b = 0, 1  
    while a <= limit:  
        fib_set.add(a)  
        a, b = b, a + b  
    return fib_set  
  
n = int(input())  
arr = list(map(int, input().split()))  
fib_set = generate_fibonacci(max(arr))  
  
count = sum(1 for num in arr if num in fib_set)  
print(count)
```

3. Bài tập mở rộng:

Bài 1.1: Đếm số Fibonacci trong dãy với số lượng xuất hiện nhiều hơn một lần

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số lượng số Fibonacci xuất hiện nhiều hơn một lần trong dãy.

Ví dụ:

Input	Output
6 1 4 8 13 21 21	1

Bài 1.2: Tìm số Fibonacci lớn nhất trong dãy

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số Fibonacci lớn nhất trong dãy, nếu không có số Fibonacci thì in ra -1.

Ví dụ:

Input	Output
5 1 3 8 13 21	21

Bài 1.3: Tìm số Fibonacci nhỏ nhất trong dãy và in ra vị trí của nó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra số Fibonacci nhỏ nhất trong dãy và vị trí của nó, nếu không có số Fibonacci thì in ra -1.

Ví dụ:

Input	Output
5 1 3 8 13 21	1 1

Bài 2: Tìm dãy con liên tiếp có tổng nhỏ nhất

1. Phân tích giải thuật:

1. Sử dụng thuật toán Kadane's nhưng thay đổi điều kiện để tìm tổng nhỏ nhất thay vì tổng lớn nhất.
2. Duyệt qua dãy số, cập nhật tổng nhỏ nhất cho dãy con liên tiếp.

2. Code tham khảo

```
n = int(input())
arr = list(map(int, input().split()))

min_sum = current_sum = arr[0]

for i in range(1, n):
    current_sum = min(arr[i], current_sum + arr[i])
    min_sum = min(min_sum, current_sum)

print(min_sum)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm dãy con liên tiếp có tổng lớn nhất và độ dài tối thiểu k

Input:

- Dòng đầu tiên chứa số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp có độ dài ít nhất là k.

Ví dụ:

Input	Output
6 3	0
3 -4 2 -5 1 4	

Bài 2.2: Tìm dãy con liên tiếp có tổng lớn nhất với các phần tử không liên tiếp nhau

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng lớn nhất của dãy con với các phần tử không liên tiếp nhau.

Ví dụ:

Input	Output
5 3 2 7 10 -5	5

Bài 2.3: Tìm dãy con liên tiếp có tổng nhỏ nhất với các phần tử liên tiếp nhau không âm

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng nhỏ nhất của dãy con liên tiếp có tất cả các phần tử không âm.

Ví dụ:

Input	Output
6 1 2 -3 4 5 -6	1

Bài 3: Đếm số lượng từ có độ dài lẻ

1. Phân tích giải thuật:

1. Sử dụng phương pháp tách từ bằng cách sử dụng hàm `split()`.
2. Kiểm tra độ dài của từng từ và đếm số lượng từ có độ dài lẻ.

2. Code tham khảo

```
s = input().split()
count = sum(1 for word in s if len(word) % 2 != 0)
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng từ có độ dài chẵn trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ có độ dài chẵn trong xâu.

Ví dụ:

Input	Output
this is an example	2

Bài 3.2: Đếm số lượng từ có chứa ít nhất một chữ số

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ có chứa ít nhất một chữ số trong xâu.

Ví dụ:

Input	Output
my number is 123abc and 456xyz	2

Bài 3.3: Đếm số lượng từ bắt đầu bằng nguyên âm trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ bắt đầu bằng nguyên âm (a, e, i, o, u) trong xâu.

Ví dụ:

Input	Output
apple orange grape	2

Bài 4: Kiểm tra danh sách tăng dần

1. Phân tích giải thuật

1. Duyệt qua danh sách và kiểm tra xem mỗi phần tử có lớn hơn phần tử trước đó không.

2. Nếu tất cả các phần tử đều thỏa mãn điều kiện, in "YES", ngược lại in "NO".

2. Code tham khảo

```
n = int(input())
arr = list(map(int, input().split()))

if all(arr[i] < arr[i+1] for i in range(n-1)):
    print("YES")
else:
    print("NO")
```

3. Bài tập mở rộng:

Bài 4.1: Kiểm tra danh sách giảm dần

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra "YES" nếu danh sách là giảm dần, "NO" nếu không phải.

Ví dụ:

Input	Output
5 5 4 3 2 1	NO

Bài 4.2: Kiểm tra danh sách có chứa phần tử trùng lặp hay không

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra "YES" nếu danh sách có chứa phần tử trùng lặp, "NO" nếu không có.

Ví dụ:

Input	Output
6	YES

1 2 3 4 5 5	
-------------	--

Bài 4.3: Kiểm tra danh sách có phải là danh sách tăng dần chẵn hay không

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 10^9).

Output:

- In ra "YES" nếu danh sách là tăng dần và tất cả các số đều là số chẵn, "NO" nếu không thỏa mãn điều kiện.

Ví dụ:

Input	Output
5 2 4 6 8 10	NO

ĐỀ SỐ 06

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Đếm số lượng chữ số xuất hiện	DIGITCNT*	Bàn phím	Màn hình	4
Bài 2	Tìm tổng lớn nhất của dãy con với k phần tử	MAXSUMK*	Bàn phím	Màn hình	5
Bài 3	Đếm số phần tử nhỏ hơn giá trị trung bình	LESSTHANAVG*	Bàn phím	Màn hình	5
Bài 4	Tìm chuỗi con là số nhị phân đối xứng	BINPALINDROME*	Bàn phím	Màn hình	6

Bài 1: Đếm số lượng chữ số xuất hiện

Cho một xâu ký tự, hãy đếm số lượng từng chữ số xuất hiện trong xâu.

Input

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 105 ký tự), bao gồm cả chữ cái và chữ số.

Output

- In ra 10 dòng, mỗi dòng chứa số lượng của một chữ số từ 0 đến 9 xuất hiện trong xâu.

Ví dụ:

Input	Output
a1b2c3d4e5f6g7h8i9j0	1 1 1 1 1 1 1 1 1 1

Bài 2: Tìm tổng lớn nhất của dãy con với k phần tử

Cho một dãy số nguyên dương và một số k. Hãy tìm tổng lớn nhất của dãy con liên tiếp có k phần tử.

Input

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra tổng lớn nhất của dãy con liên tiếp có k phần tử.

Ví dụ:

Input	Output
5 2 2 3 5 1 7	9

Bài 3: Đếm số phần tử nhỏ hơn giá trị trung bình

Cho một dãy số nguyên dương. Hãy đếm số lượng phần tử trong dãy có giá trị nhỏ hơn giá trị trung bình của cả dãy.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra số lượng phần tử có giá trị nhỏ hơn giá trị trung bình của dãy.

Ví dụ:

Input	Output
5 1 2 3 4 5	3

Bài 4. Tìm chuỗi con là số nhị phân đối xứng

Cho một chuỗi ký tự nhị phân (chỉ chứa '0' và '1'). Hãy tìm chuỗi con là số nhị phân đối xứng dài nhất trong chuỗi.

Input:

- Một dòng duy nhất chứa chuỗi ký tự nhị phân (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con là số nhị phân đối xứng dài nhất trong chuỗi. Nếu có nhiều chuỗi con có cùng độ dài, in ra chuỗi con xuất hiện đầu tiên.

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

Ví dụ:

Input	Output
110011001	110011

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Đếm số lượng chữ số xuất hiện

1. Phân tích giải thuật:

1. Sử dụng một mảng hoặc từ điển để đếm số lần xuất hiện của các chữ số từ 0 đến 9.
2. Duyệt qua chuỗi ký tự và cập nhật mảng/từ điển.

2. Code tham khảo:

```
s = input()
count = [0] * 10

for char in s:
    if char.isdigit():
        count[int(char)] += 1

for i in range(10):
    print(count[i])
```

3. Bài tập mở rộng:

Bài 1.1: Đếm số lượng chữ số chẵn xuất hiện trong chuỗi

Input:

- Một dòng duy nhất chứa chuỗi ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng các chữ số chẵn xuất hiện trong chuỗi.

Ví dụ:

Input	Output
abc1234567890	5

Bài 1.2: Đếm tổng các chữ số xuất hiện trong chuỗi

Input:

- Một dòng duy nhất chứa chuỗi ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng tất cả các chữ số xuất hiện trong chuỗi.

Ví dụ:

Input	Output
abc1234567890	45

Bài 1.3: Đếm số lượng chữ số lẻ xuất hiện trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng các chữ số lẻ xuất hiện trong xâu.

Ví dụ:

Input	Output
abc1234567890	5

Bài 2: Tìm tổng lớn nhất của dãy con với k phần tử

1. Phân tích giải thuật:

1. Sử dụng kỹ thuật "Sliding Window" để tính tổng của dãy con liên tiếp và cập nhật tổng lớn nhất.

2. Bắt đầu với tổng của k phần tử đầu tiên, sau đó dịch cửa sổ sang phải và trừ phần tử đầu, cộng phần tử tiếp theo để tìm tổng mới.

2. Code tham khảo

```
n, k = map(int, input().split())
arr = list(map(int, input().split()))

max_sum = sum(arr[:k])
current_sum = max_sum

for i in range(k, n):
    current_sum += arr[i] - arr[i - k]
    max_sum = max(max_sum, current_sum)

print(max_sum)
```


3. Bài tập mở rộng:

Bài 2.1: Tìm tổng nhỏ nhất của dãy con liên tiếp với k phần tử

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng nhỏ nhất của dãy con liên tiếp có k phần tử.

Ví dụ:

Input	Output
5 2 2 3 5 1 7	4

Bài 2.2: Tìm tổng lớn nhất của dãy con liên tiếp với độ dài tối thiểu k

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp có độ dài ít nhất là k .

Ví dụ:

Input	Output
7 3 2 3 5 1 7 8 9	15

Bài 2.3: Tìm tổng lớn nhất của dãy con liên tiếp có k phần tử và không có số nguyên tố

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp có k phần tử mà không có số nguyên tố.

Ví dụ:

Input	Output
6 2 4 6 8 10 14 16	14

Bài 3: Đếm số phần tử nhỏ hơn giá trị trung bình

1. Phân tích giải thuật:

1. Tính tổng và giá trị trung bình của dãy số.
2. Duyệt qua dãy số và đếm số phần tử có giá trị nhỏ hơn giá trị trung bình.

2. Code tham khảo

```
n = int(input())  
arr = list(map(int, input().split()))  
  
average = sum(arr) / n  
count = sum(1 for x in arr if x < average)  
  
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số phần tử lớn hơn giá trị trung bình

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lượng phần tử có giá trị lớn hơn giá trị trung bình của dãy.

Ví dụ:

Input	Output
5 1 2 3 4 5	2

Bài 3.2: Đếm số phần tử bằng giá trị trung bình

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lượng phần tử có giá trị bằng với giá trị trung bình của dãy.

Ví dụ:

Input	Output
5 1 2 3 4 5	1

Bài 3.3: Tìm phần tử nhỏ nhất lớn hơn giá trị trung bình

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra giá trị của phần tử nhỏ nhất trong dãy mà lớn hơn giá trị trung bình, nếu không có in ra -1.

Ví dụ:

Input	Output
5 1 2 3 4 5	4

Bài 4: Tìm chuỗi con là số nhị phân đối xứng

1. Phân tích giải thuật

1. Duyệt qua tất cả các chuỗi con của xâu và kiểm tra xem chuỗi con đó có đối xứng không.
2. Tìm chuỗi con đối xứng dài nhất.

2. Code tham khảo

```
s = input()
```

```
max_palindrome = ""

for i in range(len(s)):
    for j in range(i + 1, len(s) + 1):
        sub = s[i:j]
        if sub == sub[::-1] and len(sub) > len(max_palindrome):
            max_palindrome = sub

print(max_palindrome)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm chuỗi con là số nhị phân đối xứng có độ dài lẻ lớn nhất

Input:

- Một dòng duy nhất chứa xâu ký tự nhị phân (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con là số nhị phân đối xứng có độ dài lẻ lớn nhất trong xâu.

Ví dụ:

Input	Output
110101101	10101

Bài 4.2: Đếm số lượng chuỗi con là số nhị phân đối xứng có độ dài ít nhất k

Input:

- Dòng đầu tiên chứa số nguyên dương k ($1 \leq k \leq 10^5$).
- Dòng thứ hai chứa xâu ký tự nhị phân (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng chuỗi con là số nhị phân đối xứng có độ dài ít nhất là k.

Ví dụ:

Input	Output
4	2
110011001	

Bài 4.3: Tìm chuỗi con là số nhị phân đối xứng dài nhất có số '0' và '1' bằng nhau

Input:

- Một dòng duy nhất chứa xâu ký tự nhị phân (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con là số nhị phân đối xứng dài nhất có số '0' và '1' bằng nhau.

Ví dụ:

Input	Output
1010010101	10100101

ĐỀ SỐ 07

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Đếm số phần tử lớn hơn phần tử kế trước	COUNTGREATER*	Bàn phím	Màn hình	4
Bài 2	Tìm dãy con liên tiếp có tích lớn nhất	MAXPRODUCT*	Bàn phím	Màn hình	5
Bài 3	Đếm số lượng từ có chứa chữ hoa	COUNTCAPITAL*	Bàn phím	Màn hình	5
Bài 4	Xây dựng chuỗi từ danh sách con lặp lại	BUILDSUBSTRING*	Bàn phím	Màn hình	6

Bài 1: Đếm số phần tử lớn hơn phần tử kế trước

Cho một danh sách các số nguyên dương. Hãy đếm xem có bao nhiêu phần tử trong danh sách lớn hơn phần tử ngay trước nó.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra số lượng phần tử lớn hơn phần tử ngay trước nó.

Ví dụ:

Input	Output
5 1 2 2 4 3	3

Bài 2: Tìm dãy con liên tiếp có tích lớn nhất

Cho một dãy số nguyên (có thể có số âm). Hãy tìm dãy con liên tiếp có tích lớn nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).

- Dòng thứ hai chứa n số nguyên.

Output

- In ra tích lớn nhất của dãy con liên tiếp có tích lớn nhất.

Ví dụ:

Input	Output
5 2 3 -2 4 5	120

Bài 3: Đếm số lượng từ có chứa chữ hoa

Cho một xâu ký tự chứa các từ được phân cách bởi khoảng trắng. Hãy đếm số lượng từ có chứa ít nhất một chữ cái viết hoa.

Input

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng từ có chứa ít nhất một chữ cái viết hoa.

Ví dụ:

Input	Output
This is An Example	2

Bài 4. Xây dựng chuỗi từ danh sách con lặp lại

Cho một xâu ký tự S . Hãy xây dựng một chuỗi mới bằng cách nối các chuỗi con lặp lại của S có độ dài lớn nhất.

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi mới được xây dựng từ các chuỗi con lặp lại có độ dài lớn nhất.

Ví dụ:

Input	Output
abababab	ababab

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Đếm số phần tử lớn hơn phần tử kế trước

1. Phân tích giải thuật:

1. Duyệt qua danh sách và so sánh từng phần tử với phần tử trước nó.
2. Đếm số lượng phần tử thỏa mãn điều kiện lớn hơn phần tử trước đó.

2. Code tham khảo:

```
n = int(input())
arr = list(map(int, input().split()))

count = sum(1 for i in range(1, n) if arr[i] >
arr[i-1])
print(count)
```

3. Bài tập mở rộng:

Bài 1.1: Đếm số phần tử nhỏ hơn phần tử kế tiếp

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lượng phần tử nhỏ hơn phần tử ngay sau nó.

Ví dụ:

Input	Output
5	2
5 4 3 2 1	

Bài 1.2: Tìm phần tử lớn nhất mà lớn hơn phần tử kế trước

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra phần tử lớn nhất trong dãy mà lớn hơn phần tử ngay trước nó, nếu không có thì in ra -1.

Ví dụ:

Input	Output
6 1 2 3 1 4 5	4

Bài 1.3: Đếm số phần tử lớn hơn cả phần tử trước và sau nó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lượng phần tử lớn hơn cả phần tử trước và sau nó.

Ví dụ:

Input	Output
5 1 3 2 4 1	1

Bài 2: Tìm dãy con liên tiếp có tích lớn nhất

1. Phân tích giải thuật:

1. Sử dụng kỹ thuật Sliding Window để tính toán tích của dãy con liên tiếp.
2. Giữ một biến lưu tích lớn nhất và cập nhật khi tìm thấy tích lớn hơn.

2. Code tham khảo

```
n = int(input())
arr = list(map(int, input().split()))

max_prod = arr[0]
current_max = current_min = arr[0]

for i in range(1, n):
    if arr[i] < 0:
        current_max, current_min = current_min,
```

```
current_max  
    current_max = max(arr[i], current_max * arr[i])  
    current_min = min(arr[i], current_min * arr[i])  
    max_prod = max(max_prod, current_max)  
  
print(max_prod)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm tích nhỏ nhất của dãy con liên tiếp

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tích nhỏ nhất của dãy con liên tiếp có tích nhỏ nhất.

Ví dụ:

Input	Output
4 2 3 -2 4	-24

Bài 2.2: Tìm dãy con có tích lớn nhất và độ dài tối thiểu k

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tích lớn nhất của dãy con liên tiếp có độ dài ít nhất là k .

Ví dụ:

Input	Output
5 3 2 3 -2 4 5	60

Bài 2.3: Tìm dãy con liên tiếp có tích lớn nhất mà không có số âm

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tích lớn nhất của dãy con liên tiếp mà không có số âm.

Ví dụ:

Input	Output
6 1 -2 0 3 2 4	6

Bài 3: Đếm số lượng từ có chứa chữ hoa

1. Phân tích giải thuật:

1. Sử dụng phương pháp tách từ bằng cách sử dụng hàm `split()`.
2. Duyệt qua từng từ và kiểm tra xem từ đó có chứa chữ hoa không.

2. Code tham khảo

```
s = input().split()
count = sum(1 for word in s if any(char.isupper() for char in word))
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng từ bắt đầu bằng chữ hoa

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ bắt đầu bằng chữ cái viết hoa.

Ví dụ:

Input	Output
This is An Example	3

Bài 3.2: Đếm số lượng từ chỉ chứa chữ hoa

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ chỉ chứa chữ cái viết hoa.

Ví dụ:

Input	Output
THIS IS an EXAMPLE	3

Bài 3.3: Tìm từ chứa nhiều chữ hoa nhất trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra từ có chứa nhiều chữ hoa nhất trong xâu, nếu có nhiều từ cùng số lượng chữ hoa thì in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
This Is An Example	Example

Bài 4: Xây dựng chuỗi từ danh sách con lặp lại

1. Phân tích giải thuật

1. Duyệt qua các chuỗi con của S và tìm các chuỗi con lặp lại có độ dài lớn nhất.
2. Nối các chuỗi con đó lại với nhau để tạo thành chuỗi mới.

2. Code tham khảo

```
s = input()
n = len(s)

max_substring = ""
for i in range(1, n//2 + 1):
    if n % i == 0:
```

```
sub = s[:i]
if sub * (n // i) == s:
    max_substring = sub

print(max_substring * (n // len(max_substring)))
```

3. Bài tập mở rộng:

Bài 4.1: Tìm chuỗi con lặp lại dài nhất xuất hiện ít nhất hai lần

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con lặp lại dài nhất xuất hiện ít nhất hai lần trong S.

Ví dụ:

Input	Output
abcbabcabc	abc

Bài 4.2: Tìm chuỗi con lặp lại có độ dài lớn nhất mà không có ký tự nào lặp lại trong nó

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con lặp lại có độ dài lớn nhất mà không có ký tự nào lặp lại trong chuỗi đó.

Ví dụ:

Input	Output
abcbabcabc	abc

Bài 4.3: Tìm chuỗi con lặp lại có độ dài lớn nhất và tất cả các ký tự trong đó đều là chữ số

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con lặp lại có độ dài lớn nhất mà tất cả các ký tự trong chuỗi đó đều là chữ số.

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

Ví dụ:

Input	Output
123123456456	123123

ĐỀ SỐ 08

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tìm số lớn nhất chia hết cho k	MAXDIVK*	Bàn phím	Màn hình	4
Bài 2	Tìm dãy con liên tiếp có tổng lớn nhất với k phần tử lẻ	MAXODDKSUM*	Bàn phím	Màn hình	5
Bài 3	Đếm số lượng từ có độ dài nhỏ nhất	MINLENGTHWORDS*	Bàn phím	Màn hình	5
Bài 4	Tạo chuỗi từ các ký tự không lặp lại	UNIQUECHARSTR*	Bàn phím	Màn hình	6

Bài 1: Tìm số lớn nhất chia hết cho k

Cho một danh sách các số nguyên dương và một số k. Hãy tìm số lớn nhất trong danh sách chia hết cho k.

Input

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra số lớn nhất trong danh sách chia hết cho k. Nếu không có số nào chia hết cho k, in ra -1.

Ví dụ:

Input	Output
5 3 3 6 9 10 15	9

Bài 2: Tìm dãy con liên tiếp có tổng lớn nhất với k phần tử lẻ

Cho một dãy số nguyên và một số k. Hãy tìm dãy con liên tiếp có tổng lớn nhất với đúng k phần tử lẻ.

Input

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra tổng lớn nhất của dãy con liên tiếp có đúng k phần tử lẻ. Nếu không tìm thấy, in ra -1.

Ví dụ:

Input	Output
7 3 1 2 3 4 5 6 7	19

Bài 3: Đếm số lượng từ có độ dài nhỏ nhất

Cho một chuỗi ký tự chứa các từ được phân cách bởi khoảng trắng. Hãy đếm số lượng từ có độ dài nhỏ nhất trong chuỗi.

Input

- Một dòng duy nhất chứa chuỗi ký tự (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng từ có độ dài nhỏ nhất.

Ví dụ:

Input	Output
this is an example	1

Bài 4. Tạo chuỗi từ các ký tự không lặp lại

Cho một chuỗi ký tự S . Hãy tạo một chuỗi mới từ các ký tự không lặp lại trong S và giữ nguyên thứ tự xuất hiện của chúng.

Input:

- Một dòng duy nhất chứa chuỗi ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi mới được tạo từ các ký tự không lặp lại trong S .

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

Ví dụ:

Input	Output
programming	progamin

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tìm số lớn nhất chia hết cho k

1. Phân tích giải thuật:

1. Duyệt qua danh sách và kiểm tra điều kiện chia hết cho k.
2. Giữ lại số lớn nhất thỏa mãn điều kiện này.

2. Code tham khảo:

```
n, k = map(int, input().split())
arr = list(map(int, input().split()))

max_num = -1
for num in arr:
    if num % k == 0:
        max_num = max(max_num, num)

print(max_num)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm số nhỏ nhất chia hết cho k trong danh sách

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số nhỏ nhất trong danh sách chia hết cho k. Nếu không có số nào chia hết cho k, in ra -1.

Ví dụ:

Input	Output
6 4	8
8 12 16 20 5 9	

Bài 1.2: Đếm số lượng số chia hết cho k trong danh sách

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lượng số chia hết cho k trong danh sách.

Ví dụ:

Input	Output
7 5 5 10 15 20 25 1 2	3

Bài 1.3: Tìm số lớn nhất chia hết cho k và là số nguyên tố trong danh sách

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lớn nhất trong danh sách chia hết cho k và là số nguyên tố.
- Nếu không có số nào thỏa mãn điều kiện, in ra -1.

Ví dụ:

Input	Output
8 7 14 7 21 28 49 8 3 11	7

Bài 2: Tìm dãy con liên tiếp có tổng lớn nhất với k phần tử lẻ

1. Phân tích giải thuật:

1. Sử dụng kỹ thuật Sliding Window để kiểm tra các dãy con liên tiếp với k phần tử lẻ.

2. Tính tổng của các dãy con này và giữ lại tổng lớn nhất.

2. Code tham khảo

```
n, k = map(int, input().split())
arr = list(map(int, input().split()))

max_sum = -1
current_sum = 0
odd_count = 0
start = 0
```

```
for end in range(n):
    current_sum += arr[end]
    if arr[end] % 2 != 0:
        odd_count += 1

    while odd_count > k:
        if arr[start] % 2 != 0:
            odd_count -= 1
        current_sum -= arr[start]
        start += 1

    if odd_count == k:
        max_sum = max(max_sum, current_sum)

print(max_sum)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm dãy con liên tiếp có tổng nhỏ nhất với k phần tử lẻ

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng nhỏ nhất của dãy con liên tiếp có đúng k phần tử lẻ.

Ví dụ:

Input	Output
6 2 1 2 3 4 5 6	5

Bài 2.2: Tìm dãy con liên tiếp có tổng lớn nhất với k phần tử chẵn

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).

- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp có đúng k phần tử chẵn.

Ví dụ:

Input	Output
7 3 2 4 6 1 3 5 7	16

Bài 2.3: Tìm dãy con liên tiếp có tổng lớn nhất với k phần tử không chia hết cho 3

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp có đúng k phần tử không chia hết cho 3.

Ví dụ:

Input	Output
6 2 3 1 4 1 5 9	10

Bài 3: Đếm số lượng từ có độ dài nhỏ nhất

1. Phân tích giải thuật:

1. Sử dụng phương pháp tách từ bằng cách sử dụng hàm `split()`.
2. Tìm độ dài nhỏ nhất của các từ, sau đó đếm số lượng từ có độ dài này.

2. Code tham khảo

```
s = input().split()
min_length = min(len(word) for word in s)
count = sum(1 for word in s if len(word) == min_length)
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng từ có độ dài lớn nhất trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ có độ dài lớn nhất trong xâu.

Ví dụ:

Input	Output
this is an example	1

Bài 3.2: Tìm từ có độ dài nhỏ nhất và in ra

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra từ có độ dài nhỏ nhất trong xâu, nếu có nhiều từ cùng độ dài thì in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
this is an example	an

Bài 3.3: Đếm số lượng từ có độ dài chẵn trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ có độ dài chẵn trong xâu.

Ví dụ:

Input	Output
this is an example	2

Bài 4: Tạo chuỗi từ các ký tự không lặp lại

1. Phân tích giải thuật

1. Duyệt qua xâu và sử dụng một tập hợp để lưu các ký tự đã xuất hiện.
2. Thêm vào chuỗi mới các ký tự chưa xuất hiện trước đó.

2. Code tham khảo

```
s = input()
seen = set()
result = []

for char in s:
    if char not in seen:
        seen.add(char)
        result.append(char)

print("".join(result))
```

3. Bài tập mở rộng:

Bài 4.1: Tạo chuỗi từ các ký tự lặp lại đúng 2 lần trong S

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi mới được tạo từ các ký tự lặp lại đúng 2 lần trong S.

Ví dụ:

Input	Output
programming	rg

Bài 4.2: Tạo chuỗi từ các ký tự không lặp lại trong S với thứ tự ngược lại

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi mới được tạo từ các ký tự không lặp lại trong S nhưng theo thứ tự ngược lại.

Ví dụ:

Input	Output
programming	nigamorp

Bài 4.3: Tạo chuỗi từ các ký tự xuất hiện ít nhất 2 lần trong S

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi mới được tạo từ các ký tự xuất hiện ít nhất 2 lần trong S.

Ví dụ:

Input	Output
programming	rg

ĐỀ SỐ 09

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Đếm số lần xuất hiện của mỗi phần tử	ELEMENTFR EQ*	Bàn phím	Màn hình	4
Bài 2	Tìm tổng tích lũy lớn nhất	MAXCUMSU M*	Bàn phím	Màn hình	5
Bài 3	Đếm số từ không có nguyên âm	COUNTNOV OWEL*	Bàn phím	Màn hình	5
Bài 4	Tìm chuỗi con dài nhất có các ký tự theo thứ tự tăng dần	LONGESTOR DEREDSUBS TR*	Bàn phím	Màn hình	6

Bài 1: Đếm số lần xuất hiện của mỗi phần tử

Cho một danh sách các số nguyên. Hãy đếm số lần xuất hiện của mỗi phần tử trong danh sách và in ra kết quả.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra các cặp số nguyên theo thứ tự xuất hiện ban đầu, mỗi cặp gồm phần tử và số lần xuất hiện của nó.

Ví dụ:

Input	Output
7 1 2 1 3 1 2 3	1 2 2 2 3 2

Bài 2: Tìm tổng tích lũy lớn nhất

Cho một dãy số nguyên dương. Hãy tìm tổng tích lũy lớn nhất của một

đoạn con liên tiếp trong dãy số.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output

- In ra tổng tích lũy lớn nhất của một đoạn con liên tiếp.

Ví dụ:

Input	Output
5 1 2 3 4 5	15

Bài 3: Đếm số từ không có nguyên âm

Cho một xâu ký tự chứa các từ được phân cách bởi khoảng trắng. Hãy đếm số từ không chứa bất kỳ nguyên âm nào (a, e, i, o, u).

Input

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng từ không chứa bất kỳ nguyên âm nào.

Ví dụ:

Input	Output
my gym bcd xyz	3

Bài 4. Tìm chuỗi con dài nhất có các ký tự theo thứ tự tăng dần

Cho một xâu ký tự S . Hãy tìm chuỗi con dài nhất trong đó các ký tự xuất hiện theo thứ tự tăng dần theo bảng chữ cái.

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con dài nhất có các ký tự theo thứ tự tăng dần.

Ví dụ:

Input	Output
abcbda b cefg	abcefg

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Đếm số lần xuất hiện của mỗi phần tử

1. Phân tích giải thuật:

1. Dùng từ điển để đếm số lần xuất hiện của mỗi phần tử trong danh sách.
2. Duyệt qua danh sách và in ra các cặp số và số lần xuất hiện tương ứng.

2. Code tham khảo:

```
n = int(input())
arr = list(map(int, input().split()))

freq = {}
for num in arr:
    if num in freq:
        freq[num] += 1
    else:
        freq[num] = 1

for num in arr:
    if freq[num] > 0:
        print(num, freq[num])
        freq[num] = 0
```

3. Bài tập mở rộng:

Bài 1.1: Tìm phần tử xuất hiện nhiều nhất trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra phần tử xuất hiện nhiều nhất và số lần xuất hiện của nó.

Ví dụ:

Input	Output
6 1 2 1 3 1 2	1 3

Bài 1.2: Tìm phần tử xuất hiện ít nhất trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).

- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra phần tử xuất hiện ít nhất và số lần xuất hiện của nó.

Ví dụ:

Input	Output
6 1 2 1 3 1 2	3 1

Bài 1.3: Đếm số lượng phần tử xuất hiện đúng k lần trong danh sách

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq n$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra số lượng phần tử xuất hiện đúng k lần.

Ví dụ:

Input	Output
7 2 1 2 1 3 1 2 3	2

Bài 2: Tìm tổng tích lũy lớn nhất

1. Phân tích giải thuật:

1. Sử dụng thuật toán Kadane để tìm tổng tích lũy lớn nhất.
2. Duyệt qua dãy số và cập nhật tổng tích lũy lớn nhất khi gặp đoạn con mới có tổng lớn hơn.

2. Code tham khảo

```
n = int(input())
arr = list(map(int, input().split()))

max_sum = arr[0]
current_sum = arr[0]
```

```
for i in range(1, n):  
    current_sum = max(arr[i], current_sum + arr[i])  
    max_sum = max(max_sum, current_sum)  
  
print(max_sum)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm tổng tích lũy nhỏ nhất của một đoạn con liên tiếp

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng tích lũy nhỏ nhất của một đoạn con liên tiếp.

Ví dụ:

Input	Output
5 2 3 1 5 4	1

Bài 2.2: Tìm tổng tích lũy lớn nhất của một đoạn con liên tiếp với độ dài lớn hơn hoặc bằng k

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng tích lũy lớn nhất của một đoạn con liên tiếp có độ dài ít nhất là k .

Ví dụ:

Input	Output
7 3 2 1 5 6 3 2 4	12

Bài 2.3: Tìm tổng tích lũy lớn nhất của một đoạn con liên tiếp mà không chứa số chẵn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng tích lũy lớn nhất của một đoạn con liên tiếp mà không chứa số chẵn.

Ví dụ:

Input	Output
6 1 3 5 2 4 6	9

Bài 3: Đếm số từ không có nguyên âm

1. Phân tích giải thuật:

1. Sử dụng phương pháp tách từ bằng cách sử dụng hàm `split()`.
2. Kiểm tra từng từ và đếm số từ không chứa nguyên âm.

2. Code tham khảo

```
s = input().split()
vowels = set("aeiou")
count = sum(1 for word in s if not any(char in vowels
for char in word))
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số từ chứa tất cả các nguyên âm

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ chứa tất cả các nguyên âm (a, e, i, o, u).

Ví dụ:

Input	Output
education bcd euouae	2

Bài 3.2: Tìm từ có số lượng nguyên âm nhiều nhất

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra từ có số lượng nguyên âm nhiều nhất, nếu có nhiều từ cùng số lượng nguyên âm thì in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
education university	education

Bài 3.3: Tìm từ dài nhất không chứa nguyên âm

Input:

- Một dòng duy nhất chứa xâu ký tự (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất không chứa nguyên âm, nếu có nhiều từ cùng độ dài thì in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
bcd fgh jklm gym	jklm

Bài 4: Tìm chuỗi con dài nhất có các ký tự theo thứ tự tăng dần

1. Phân tích giải thuật

1. Duyệt qua xâu ký tự và tìm các chuỗi con có ký tự theo thứ tự tăng dần.
2. Lưu lại chuỗi con dài nhất.

2. Code tham khảo

```
s = input()

longest = current = s[0]
```



```
for i in range(1, len(s)):\n    if s[i] >= s[i - 1]:\n        current += s[i]\n    else:\n        if len(current) > len(longest):\n            longest = current\n        current = s[i]\n\nif len(current) > len(longest):\n    longest = current\n\nprint(longest)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm chuỗi con dài nhất có các ký tự theo thứ tự giảm dần

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con dài nhất có các ký tự theo thứ tự giảm dần.

Ví dụ:

Input	Output
fedcbaabc	fedcba

Bài 4.2: Tìm chuỗi con dài nhất có các ký tự không trùng lặp

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con dài nhất mà không có ký tự nào trùng lặp.

Ví dụ:

Input	Output
abcdabcde	abcde

Bài 4.3: Tìm chuỗi con dài nhất có các ký tự theo thứ tự tăng dần mà chỉ chứa các nguyên âm

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con dài nhất có các ký tự theo thứ tự tăng dần mà chỉ chứa các nguyên âm.

Ví dụ:

Input	Output
aeiouaeiou	aeiou

ĐỀ SỐ 10

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tìm phần tử nhỏ nhất thỏa mãn điều kiện	SMALLESTCOND*	Bàn phím	Màn hình	4
Bài 2	Tìm số nhỏ nhất có tổng các chữ số là S	MINNUMSUMDIGIT*	Bàn phím	Màn hình	5
Bài 3	Đếm số lượng xâu nhị phân đối xứng	COUNTPALINDROM*	Bàn phím	Màn hình	5
Bài 4	Tìm số con đường ngắn nhất trên bảng	SHORTESTPATHGRID*	Bàn phím	Màn hình	6

Bài 1: Tìm phần tử nhỏ nhất thỏa mãn điều kiện

Cho một danh sách các số nguyên dương. Hãy tìm phần tử nhỏ nhất trong danh sách sao cho phần tử đó lớn hơn x.

Input

- Dòng đầu tiên chứa hai số nguyên dương n và x ($1 \leq n \leq 10^5$, $1 \leq x \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra phần tử nhỏ nhất trong danh sách lớn hơn x. Nếu không có phần tử nào thỏa mãn, in ra -1.

Ví dụ:

Input	Output
6 7 5 9 8 7 10 6	8

Bài 2: Tìm số nhỏ nhất có tổng các chữ số là S

Cho một số nguyên dương S. Hãy tìm số nguyên dương nhỏ nhất có tổng các chữ số bằng S.

Input

- Một dòng duy nhất chứa số nguyên dương S ($1 \leq S \leq 1000$).

Output

- In ra số nguyên dương nhỏ nhất có tổng các chữ số bằng S .

Ví dụ:

Input	Output
15	69

Bài 3: Đếm số lượng xâu nhị phân đối xứng

Cho một số nguyên dương n . Hãy đếm số lượng xâu nhị phân đối xứng độ dài n .

Input

- Một dòng duy nhất chứa số nguyên dương n ($1 \leq n \leq 100$).

Output

- In ra số lượng xâu nhị phân đối xứng độ dài n .

Ví dụ:

Input	Output
4	4

Bài 4. Tìm số con đường ngắn nhất trên bảng

Cho một bảng kích thước $m \times n$ với các ô có thể là chướng ngại vật hoặc không. Hãy tìm số lượng con đường ngắn nhất từ góc trên trái đến góc dưới phải của bảng.

Input:

- Dòng đầu tiên chứa hai số nguyên dương m và n ($1 \leq m, n \leq 100$).
- m dòng tiếp theo, mỗi dòng chứa n số, với số 1 đại diện cho ô có thể đi qua và số 0 đại diện cho chướng ngại vật.

Output:

- In ra số lượng con đường ngắn nhất từ góc trên trái đến góc dưới phải.

Ví dụ:

Input	Output
3 3 1 1 1 1 0 1 1 1 1	2

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tìm phần tử nhỏ nhất thỏa mãn điều kiện

1. Phân tích giải thuật:

1. Duyệt qua danh sách và tìm phần tử nhỏ nhất lớn hơn x.
2. Giữ lại phần tử đó và cập nhật nếu tìm thấy phần tử nhỏ hơn.

2. Code tham khảo:

```
n, x = map(int, input().split())
arr = list(map(int, input().split()))

smallest = float('inf')

for num in arr:
    if num > x:
        smallest = min(smallest, num)

print(smallest if smallest != float('inf') else -1)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm phần tử lớn nhất nhỏ hơn x

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và x ($1 \leq n \leq 10^5$, $1 \leq x \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra phần tử lớn nhất trong danh sách nhỏ hơn x. Nếu không có phần tử nào thỏa mãn, in ra -1.

Ví dụ:

Input	Output
6 7	6
5 9 8 7 10 6	

Bài 1.2: Đếm số phần tử trong danh sách nằm trong khoảng [a, b]

Input:

- Dòng đầu tiên chứa ba số nguyên dương n, a và b ($1 \leq n \leq 10^5$, $1 \leq a < b \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lượng phần tử trong danh sách nằm trong khoảng $[a, b]$.

Ví dụ:

Input	Output
6 5 10 5 9 8 7 10 6	3

Bài 1.3: Tìm phần tử nhỏ nhất trong danh sách không chia hết cho k

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra phần tử nhỏ nhất trong danh sách không chia hết cho k . Nếu không có phần tử nào thỏa mãn, in ra -1.

Ví dụ:

Input	Output
6 2 4 6 8 10 12 7	7

Bài 2: Tìm số nhỏ nhất có tổng các chữ số là S

1. Phân tích giải thuật:

1. Tạo số từ các chữ số lớn nhất có thể để giảm độ lớn của số.
2. Đảm bảo tổng các chữ số bằng S.

2. Code tham khảo

```
s = int(input())
result = ""

while s > 9:
    result += "9"
    s -= 9
```

```
if s > 0:
    result += str(s)

print(result[::-1])
```

3. Bài tập mở rộng:

Bài 2.1: Tìm số lớn nhất có tổng các chữ số bằng S

Input:

- Một dòng duy nhất chứa số nguyên dương S ($1 \leq S \leq 1000$).

Output:

- In ra số nguyên dương lớn nhất có tổng các chữ số bằng S.

Ví dụ:

Input	Output
15	96

Bài 2.2: Tìm số có ít chữ số nhất với tổng các chữ số bằng S

Input:

- Một dòng duy nhất chứa số nguyên dương S ($1 \leq S \leq 1000$).

Output:

- In ra số có ít chữ số nhất với tổng các chữ số bằng S.

Ví dụ:

Input	Output
10	19

Bài 2.3: Tìm số có tổng các chữ số bằng S và là số nguyên tố nhỏ nhất

Input:

- Một dòng duy nhất chứa số nguyên dương S ($1 \leq S \leq 1000$).

Output:

- In ra số nguyên tố nhỏ nhất có tổng các chữ số bằng S.

Ví dụ:

Input	Output
7	7

Bài 3: Đếm số lượng chuỗi nhị phân đối xứng

1. Phân tích giải thuật:

1. Một chuỗi nhị phân đối xứng có thể được tạo ra bằng cách xây dựng từ đầu và cuối về giữa.
2. Tính toán số lượng các chuỗi thỏa mãn điều kiện này.

2. Code tham khảo

```
n = int(input())

if n % 2 == 0:
    print(2 ** (n // 2))
else:
    print(2 ** (n // 2 + 1))
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng chuỗi nhị phân đối xứng có độ dài lẻ

Input:

- Một dòng duy nhất chứa số nguyên dương n ($1 \leq n \leq 100$).

Output:

- In ra số lượng chuỗi nhị phân đối xứng có độ dài lẻ n .

Ví dụ:

Input	Output
5	4

Bài 3.2: Đếm số lượng chuỗi nhị phân đối xứng có độ dài chẵn

Input:

- Một dòng duy nhất chứa số nguyên dương n ($1 \leq n \leq 100$).

Output:

- In ra số lượng xâu nhị phân đối xứng có độ dài chẵn n.

Ví dụ:

Input	Output
6	8

Bài 3.3: Tìm xâu nhị phân đối xứng lớn nhất có độ dài n

Input:

- Một dòng duy nhất chứa số nguyên dương n ($1 \leq n \leq 100$).

Output:

- In ra xâu nhị phân đối xứng lớn nhất có độ dài n.

Ví dụ:

Input	Output
5	11111

Bài 4: Tìm số con đường ngắn nhất trên bảng

1. Phân tích giải thuật

1. Sử dụng BFS để tìm tất cả các con đường ngắn nhất từ góc trên trái đến góc dưới phải.
2. Đếm số lượng con đường ngắn nhất tìm thấy.

2. Code tham khảo

```
from collections import deque

m, n = map(int, input().split())
grid = [list(map(int, input().split())) for _ in range(m)]

directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]
queue = deque([(0, 0)])
dist = [[float('inf')] * n for _ in range(m)]
dist[0][0] = 1
```

```
path_count = [[0] * n for _ in range(m)]
path_count[0][0] = 1

while queue:
    x, y = queue.popleft()

    for dx, dy in directions:
        nx, ny = x + dx, y + dy

        if 0 <= nx < m and 0 <= ny < n and
grid[nx][ny] == 1:
            if dist[nx][ny] > dist[x][y] + 1:
                dist[nx][ny] = dist[x][y] + 1
                path_count[nx][ny] = path_count[x][y]
                queue.append((nx, ny))
            elif dist[nx][ny] == dist[x][y] + 1:
                path_count[nx][ny] +=
path_count[x][y]

print(path_count[m-1][n-1])
```

3. Bài tập mở rộng:

Bài 4.1: Tìm số con đường ngắn nhất từ góc trên trái đến góc dưới phải với k chướng ngại vật bị xóa

Input:

- Dòng đầu tiên chứa ba số nguyên dương m, n và k ($1 \leq m, n \leq 100$, $0 \leq k \leq 10$).
- m dòng tiếp theo, mỗi dòng chứa n số, với số 1 đại diện cho ô có thể đi qua và số 0 đại diện cho chướng ngại vật.

Output:

- In ra số lượng con đường ngắn nhất từ góc trên trái đến góc dưới phải sau khi xóa k chướng ngại vật.

Ví dụ:

Input	Output
-------	--------

3 3 1	3
1 0 1	
1 0 1	
1 1 1	

Bài 4.2: Tìm số con đường ngắn nhất từ góc trên trái đến góc dưới phải mà chỉ được di chuyển theo đường chéo

Input:

- Dòng đầu tiên chứa hai số nguyên dương m và n ($1 \leq m, n \leq 100$).
- m dòng tiếp theo, mỗi dòng chứa n số, với số 1 đại diện cho ô có thể đi qua và số 0 đại diện cho chướng ngại vật.

Output:

- In ra số lượng con đường ngắn nhất từ góc trên trái đến góc dưới phải mà chỉ được di chuyển theo đường chéo.

Ví dụ:

Input	Output
3 3 1 0 1 0 1 0 1 0 1	1

Bài 4.3: Tìm số con đường ngắn nhất từ góc trên trái đến góc dưới phải với điều kiện phải đi qua một ô xác định

Input:

- Dòng đầu tiên chứa ba số nguyên dương m , n và k ($1 \leq m, n \leq 100$, $1 \leq k \leq m*n$).
- m dòng tiếp theo, mỗi dòng chứa n số, với số 1 đại diện cho ô có thể đi qua và số 0 đại diện cho chướng ngại vật.
- Dòng cuối cùng chứa hai số nguyên dương r và c ($1 \leq r \leq m$, $1 \leq c \leq n$), chỉ định vị trí của ô phải đi qua.

Output:

● In ra số lượng con đường ngắn nhất từ góc trên trái đến góc dưới phải và đi qua ô xác định.

Ví dụ:

Input	Output
3 3 2 2 1 1 1 1 0 1 1 1 1	1

ĐỀ SỐ 11

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Số nguyên tố lớn nhất	MAX*	Bàn phím	Màn hình	4
Bài 2	Sắp xếp số tăng dần trong chuỗi	SORT*	Bàn phím	Màn hình	5
Bài 3	Đếm cặp có tổng bằng 0	COUNT*	Bàn phím	Màn hình	5
Bài 4	Thống kê số lượng từ trong xâu	STAT*	Bàn phím	Màn hình	6

Bài 1: Số nguyên tố lớn nhất

Cho dãy số nguyên (a_1, a_2, \dots, a_n) , $1 \leq n \leq 10000$; với mọi i sao cho $a_i \leq 10^8$. Hãy tìm số nguyên tố lớn nhất trong dãy trên.

Input

- Dòng thứ nhất chứa số nguyên dương n .
- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n .

Output

- Dòng thứ nhất ghi số nguyên tố lớn nhất.
- Dòng thứ hai ghi các chỉ số trong dãy mà giá trị của nó là số nguyên tố lớn nhất.

Ví dụ:

Input	Output
9 19 7 81 33 17 4 19 21 13	19

Bài 2: Sắp xếp số tăng dần trong chuỗi

Viết chương trình nhập vào 1 chuỗi ký tự có số. Sắp xếp các số trong chuỗi tăng dần và in chuỗi ra màn hình (Đề thi Tin học trẻ Thành phố Quy Nhơn năm 2018)

Ví dụ:

Input	Output
thi10tin21hoc9tre5nam2018	thi5tin9hoc10tre21nam2018

Bài 3: Đếm cặp có tổng bằng 0

Cho dãy số A có N số nguyên. Hãy đếm số cặp (i, j) sao cho $A_i + A_j = 0$, với $i < j$.

Input

- Dòng đầu tiên chứa một số nguyên dương N ($1 \leq N \leq 2 \cdot 10^5$)
- Dòng thứ hai chứa dãy số A gồm N số nguyên cách nhau bởi một ký tự khoảng trống. ($|A_i| \leq 10^9$)

Output

- In ra một số nguyên duy nhất, là số cặp phần tử trong dãy A mà có tổng là 0.

Ví dụ:

Input	Output
3 -2 0 2	1
6 -2 -1 0 0 1 2	3

Bài 4. Thống kê số lượng từ trong câu

Cho một câu ký tự gồm nhiều từ, các từ được phân cách bởi dấu cách hoặc dấu câu. Hãy thống kê số lượng từ theo độ dài và in ra các từ có tần suất xuất hiện lớn nhất.

Input

- Dòng thứ nhất chứa câu ký tự.
 - Chiều dài của câu không quá 1000 ký tự.
 - Mỗi từ không dài quá 20 ký tự.
 - Số lượng từ tối đa là 200 từ.

Output

- Dòng thứ nhất ghi số lượng từ tương ứng với từng độ dài (1, 2, 3, ...).

- Dòng thứ hai ghi các từ có tần suất xuất hiện nhiều nhất trong câu.

Ví dụ:

Input	Output
"Tôi đang học tin học. Tôi rất thích lập trình!"	0 0 7 1 2 "Tôi"

0 từ có độ dài 1, 0 từ có độ dài 2, 7 từ có độ dài 3, 1 từ có độ dài 4, 2 từ có độ dài 5

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Số nguyên tố lớn nhất

1. Phân tích giải thuật:

1. Tạo hàm kiểm tra số nguyên tố
2. Duyệt danh sách và sử dụng hàm kiểm tra số nguyên tố, để tìm số nguyên tố lớn nhất và vị trí của nó
3. In ra số nguyên tố lớn nhất, và vị trí số lớn nhất (vị trí + 1)

2. Code tham khảo:

```
n = int(input())
ds = list(map(int, input().split()))
def check(n):
    if n<2:
        return 0
    dem=0
    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            dem+=1
    return 0 if dem>0 else 1
dsMax=[]
indexMax=[]
for i in range(len(ds)):
    if check(ds[i])==1:
        dsMax.append(ds[i])
        indexMax.append(i)
max = max(dsMax)
print(max)
for i in range(len(ds)):
    if max==ds[i]:
        print(i+1, end=" ")
```

3. Bài tập mở rộng:

Bài 1.1: Tìm số nguyên tố không trùng nhau

Viết một chương trình nhận đầu vào là một dãy số nguyên và tìm hai số nguyên tố không trùng nhau có tổng lớn nhất trong dãy đó. Trả về tổng của hai số nguyên tố đó cùng với các chỉ số tương ứng.

Ví dụ:

Input	Output
6	2 4
10 5 7 13 8 11	4 6

Bài 1.2: Tìm số nguyên tố đặc biệt

Viết chương trình nhận đầu vào một số nguyên dương n và tìm số nguyên tố đặc biệt lớn nhất không vượt quá n . Một số nguyên tố đặc biệt là số nguyên tố mà tổng của các chữ số bình phương của nó cũng là một số nguyên tố.

Ví dụ:

Input	Output
30	23

Giải thích: Số nguyên tố đặc biệt lớn nhất nhỏ hơn hoặc bằng 30 là 23, vì $2^2 + 3^3 = 4 + 9 = 13$ là một số nguyên tố.

Bài 1.3: Tìm số nguyên tố Fibonacci

Viết một chương trình nhận vào một số nguyên dương n . Tìm số nguyên tố lớn nhất trong dãy Fibonacci đến n .

Ví dụ:

Input	Output
20	13

Bài 2: Sắp xếp số tăng dần trong chuỗi

1. Phân tích giải thuật:

1. Tách chữ số khỏi chuỗi đầu vào (Kiến thức liên quan Regular Expression)
2. Chuyển đổi số sang kiểu int và sắp xếp theo thứ tự tăng dần.
3. Tách chuỗi ban đầu thành các phần không chứa số.
4. Khởi tạo lại chuỗi và xây dựng lại theo yêu cầu.

2. Code tham khảo

```
import re
s = input()
nums = re.findall(r'\d+', s)
temp = [int(i) for i in nums]
temp.sort()
subs = re.split(r'\d+', s)
s = ""
for i in range(max(len(nums), len(subs))):
    if (i < len(subs)):
        s += subs[i]
    if (i < len(nums)):
        s += str(temp[i])
print(s)
```

3. Bài tập mở rộng:

Bài 2.1: Đảo ngược các số trong chuỗi

Viết chương trình nhập vào một chuỗi ký tự có chứa số. Đảo ngược các số trong chuỗi mà không thay đổi vị trí của các ký tự không phải số.

Ví dụ:

Input	Output
"abc123def456ghi"	"abc321def654ghi"

Bài 2.2: Thêm dấu phẩy vào các số

Viết chương trình nhập vào một chuỗi có chứa các số. Thêm dấu phẩy vào các số theo định dạng hàng nghìn (ví dụ, 1000 thành 1,000) mà không thay đổi các phần khác của chuỗi.

Ví dụ:

Input	Output
"Cost estimated 2000 dollars per 100 units."	"Cost estimated 2,000 dollars per 100 units."

Bài 2.3: Tách các từ và số, sắp xếp riêng biệt

Viết chương trình nhập vào một chuỗi ký tự có chứa từ và số. Tách riêng từ và số ra, sắp xếp các từ theo thứ tự bảng chữ cái và các số theo thứ tự tăng dần, sau đó ghép chúng lại thành một chuỗi mới.

Ví dụ:

Input	Output
" banana2apple1orange3"	"apple1banana2orange3"

Bài 3: Đếm cặp có tổng bằng 0

1. Phân tích giải thuật:

1. Tạo một từ điển để lưu trữ số lần xuất hiện của từng số trong dãy
2. Duyệt danh sách, kiểm tra xem số đối của số có trong từ điển không:

+ Nếu có, tức là đã tìm thấy một cặp số mà số âm là số đối của một số dương, tăng biến đếm lên bằng số lần xuất hiện của số đối đó trong từ điển.

+ Tiếp theo, kiểm tra xem số hiện tại đã có trong từ điển chưa. Nếu có, tăng số lần xuất hiện của số đó lên 1. Nếu chưa có, thêm số vào từ điển với số lần xuất hiện là 1.

3. In ra giá trị của biến đếm, chính là số lần mà một số âm trong dãy là số đối của một số dương trong dãy đó.

2. Code tham khảo

```
n = int(input())
a = list(map(int, input().split())) dem = 0
cacso = {} for so in a:
    if -so in cacso:
        dem += cacso[-so] if so in cacso:
cacso[so] += 1 else:
cacso[so] = 1 print(dem)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm cặp số có tổng nguyên tố

Đếm cặp số có tổng là một số nguyên tố trong dãy số nguyên

Input:

- Dòng đầu tiên chứa một số nguyên dương N ($1 \leq N \leq 2 \cdot 10^5$).
- Dòng thứ hai chứa dãy số A gồm N số nguyên cách nhau bởi một ký tự khoảng trống. ($|A_i| \leq 10^9$)

Output:

- In ra một số nguyên duy nhất, là số cặp phân tử trong dãy A mà có tổng là một số nguyên tố.

Ví dụ:

Input	Output
5 2 3 5 7 11	2

Bài 3.2: Đếm cặp số có tổng Fibonacci

Đếm cặp số có tổng là một số Fibonacci trong dãy số nguyên

Input:

- Dòng đầu tiên chứa một số nguyên dương N ($1 \leq N \leq 2 \cdot 10^5$).

- Dòng thứ hai chứa dãy số A gồm N số nguyên cách nhau bởi một ký tự khoảng trống. ($|A_i| \leq 10^9$)

Output:

- In ra một số nguyên duy nhất, là số cặp phần tử trong dãy A mà có tổng là một số Fibonacci.

Ví dụ:

Input	Output
6 1 2 3 5 8 13	4

Bài 3.3: Đếm cặp số có tổng là một số hoàn hảo trong dãy số nguyên

Input:

- Dòng đầu tiên chứa một số nguyên dương N ($1 \leq N \leq 2 \cdot 10^5$).
- Dòng thứ hai chứa dãy số A gồm N số nguyên cách nhau bởi một ký tự khoảng trống. ($|A_i| \leq 10^9$)

Output:

- In ra một số nguyên duy nhất, là số cặp phần tử trong dãy A mà có tổng là một số hoàn hảo.

Ví dụ:

Input	Output
5 6 27 15 9 1	1

Bài 4: Thống kê số lượng từ trong xâu

1. Phân tích giải thuật

1. Tách từ: Sử dụng biểu thức chính quy để tách các từ ra khỏi chuỗi ký tự, bỏ qua các dấu câu.

2. Đếm độ dài từ: Duyệt qua danh sách từ đã tách và đếm số lượng từ theo độ dài (1, 2, 3,...).

3. Đếm tần suất từ: Dùng Counter để đếm số lần xuất hiện của mỗi từ và tìm từ xuất hiện nhiều nhất.

4. Xuất kết quả: In ra số lượng từ tương ứng với các độ dài từ 1 đến 5 và in từ có tần suất xuất hiện cao nhất.

2. Code tham khảo

```
import re
from collections import defaultdict, Counter

chuoi = input()

tu = re.findall(r'\b\w+\b', chuoi)
dem_do_dai = defaultdict(int)
for t in tu:
    dem_do_dai[len(t)] += 1

tan_suat_tu = Counter(tu)
tan_suat_cao_nhat = max(tan_suat_tu.values())
tu_pho_bien = [t for t, ts in tan_suat_tu.items() if
                ts == tan_suat_cao_nhat]

ket_qua_do_dai = [str(dem_do_dai[i]) if i in
                  dem_do_dai else '0' for i in range(1, 6)]
ket_qua_do_dai_str = " ".join(ket_qua_do_dai)
tu_pho_bien_str = " ".join(tu_pho_bien[:1])

print(ket_qua_do_dai_str)
print(tu_pho_bien_str)
```

3. Bài tập mở rộng:

Bài 4.1: Thống kê từ viết tắt trong email

Cho một đoạn email chứa nhiều từ viết tắt (các từ được phân cách bởi dấu cách hoặc dấu câu). Hãy xác định các từ viết tắt có tần suất xuất hiện nhiều nhất và liệt kê ra danh sách.

Input:

- Một đoạn email, độ dài không quá 2000 ký tự.
- Mỗi từ không dài quá 30 ký tự.
- Số lượng từ tối đa là 500 từ.

Output:

- Dòng thứ nhất ghi tổng số từ viết tắt có trong email.
- Dòng thứ hai ghi danh sách các từ viết tắt có tần suất xuất hiện nhiều nhất, mỗi từ viết tắt cách nhau bởi dấu phẩy.

Ví dụ:

Input	Output
"Chúng tôi sẽ sắp xếp meeting ASAP. Dự án cần được hoàn thành trước EOD. Hãy review proposal ASAP."	2 ASAP, EOD

Bài 4.2: Thống kê số lượng từ đơn, từ ghép trong bài báo

Cho nội dung một bài báo khoa học, các từ trong bài báo được phân cách bởi dấu cách hoặc dấu câu. Hãy thống kê số lượng từ đơn và từ ghép trong bài báo và liệt kê các từ ghép xuất hiện nhiều nhất.

Input:

- Nội dung bài báo không quá 5000 ký tự.
- Mỗi từ không dài quá 50 ký tự.
- Số lượng từ tối đa là 1000 từ.

Output:

- Dòng thứ nhất ghi số lượng từ đơn (từ có 1 âm tiết) và từ ghép (từ có nhiều hơn 1 âm tiết).
- Dòng thứ hai ghi danh sách các từ ghép có tần suất xuất hiện nhiều nhất.

Ví dụ:

Input	Output
"Hệ thống AI tiên tiến đang phát triển mạnh mẽ. Phát triển hệ thống không phải là điều dễ dàng."	2 4 phát triển, hệ thống

Bài 4.3: Thống kê từ theo mức độ phổ biến trong báo cáo kinh doanh

Cho nội dung một báo cáo kinh doanh, các từ trong báo cáo được phân cách bởi dấu cách hoặc dấu câu. Hãy thống kê từ theo mức độ phổ biến: từ

thông dụng, từ ít thông dụng, và từ hiếm gặp dựa trên tần suất xuất hiện của chúng.

Input:

- Nội dung báo cáo không quá 3000 ký tự.
- Mỗi từ không dài quá 40 ký tự.
- Số lượng từ tối đa là 800 từ.

Output:

- Dòng thứ nhất ghi số lượng từ thông dụng (xuất hiện từ 5 lần trở lên), từ ít thông dụng (xuất hiện từ 2 đến 4 lần), và từ hiếm gặp (xuất hiện 1 lần).
- Dòng thứ hai ghi danh sách các từ hiếm gặp.

Ví dụ:

Input	Output
"Báo cáo doanh thu cho thấy công ty đã tăng trưởng mạnh trong quý III. Tăng trưởng doanh thu là một dấu hiệu tốt cho sự phát triển."	1 2 10 quý, dấu, hiệu, tốt, sự, phát, triển, cho, trong, đã

ĐỀ SỐ 12

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Đếm cặp đôi	COUNT*	Bàn phím	Màn hình	4
Bài 2	Số chẵn lớn nhất	MAX*	Bàn phím	Màn hình	5
Bài 3	Xâu hoàn hảo	PERFECT*	Bàn phím	Màn hình	5
Bài 4	Mật khẩu	PASSWORD*	Bàn phím	Màn hình	6

Bài 1: Đếm cặp đôi

Cho dãy số A gồm n phần tử nguyên dương A_1, A_2, \dots, A_n . Mỗi phần tử có giá trị không vượt quá 10^9 và $n \leq 10^5$. Một cặp số được gọi là cặp tương đồng với x , nếu cặp số này có tổng bằng số x cho trước nào đó.

Yêu cầu: Hãy đếm xem trong dãy số A có bao nhiêu cặp số $(A_i; A_j)$ tương đồng với x (có nghĩa là $A_i + A_j = x$) với $i < j$.

Input

- Dòng đầu tiên chứa dãy số n, x ($n \leq 10^5, x \leq 10^6$).
- Dòng thứ 2 chứa n phần tử của dãy số A ($A_i \leq 10^9$).

Output

- Ghi ra một số nguyên là cặp đôi tương đồng của dãy số.

Ví dụ:

Input	Output
7 6 1 2 4 3 4 5 3	4

Bài 2: Số chẵn lớn nhất

Cho số nguyên dương n và dãy số a_1, a_2, \dots, a_n .

Yêu cầu: Hãy xác định xem liệu có tồn tại hai phần tử khác nhau sao cho tổng của chúng là số chẵn hay không. Nếu có hãy in ra số chẵn lớn nhất có thể.

Hai phần tử a_i và a_j được gọi là khác nhau nếu $i \neq j$.

Input

- Dòng đầu tiên chứa số nguyên dương n ($2 \leq n \leq 10^6$).
- Dòng thứ hai chứa dãy số a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$). Các số cách nhau một khoảng trắng.
- Dữ liệu vào đảm bảo rằng tất cả các phần tử trong dãy đều đôi một khác nhau.

Output

- In ra đáp án bài toán sau khi thực hiện yêu cầu đề bài. Nếu không tồn tại hai phần tử thỏa mãn yêu cầu đề bài hãy in ra -1.

Ví dụ:

Input	Output
3 2 3 4	6

Note: $a_1 + a_2 = 2 + 3 = 5$

$a_1 + a_3 = 2 + 4 = 6$

$a_2 + a_3 = 3 + 4 = 7$

Vậy 6 là số chẵn lớn nhất thỏa mãn yêu cầu đề bài.

Bài 3: Xâu hoàn hảo

Xâu hoàn hảo là xâu có độ dài lớn hơn hoặc bằng 2, trong đó kí tự đầu và kí tự cuối của xâu bằng nhau.

Cho một xâu S có độ dài N , đếm số lượng xâu hoàn hảo trong xâu S .

Input

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($n \leq 10^6$)
- Dòng thứ 2 là xâu S chỉ gồm các kí tự chữ cái latin in thường.

Output

- Ghi ra một số nguyên duy nhất là số lượng xâu hoàn hảo trong xâu S .

Ví dụ:

Input	Output
6	3

abcb	
------	--

Bài 4. Mật khẩu (Tin học trẻ Đà Nẵng bảng C2 – năm 2023)

Nhằm bảo mật thông tin trao đổi, người gửi và người nhận thông tin có thể đặt ra các quy tắc để tìm ra mật khẩu. Hiếu muốn trao đổi mật khẩu với Hùng nên đặt ra quy tắc như sau: Hiếu gửi cho Hùng một xâu ký tự, mật khẩu được tìm bằng cách tính tổng các chữ số chẵn có trong xâu rồi ghép với chiều dài của xâu đó.

Input:

- Gồm một dòng duy nhất chứa xâu ký tự có độ dài không quá 10^5 ký tự gồm các ký tự in hoa, in thường, ký tự số, khoảng trắng và các ký tự đặc biệt.

Output:

- In ra mật khẩu cần tìm.

Ví dụ:

Input	Output
Tin hoc tre 2023	416

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Đếm cặp đôi

1. Phân tích giải thuật:

1. Tạo danh sách để lưu số lượng các số xuất hiện.
2. Duyệt qua danh sách số, tính toán và cập nhật số lượng cặp số thỏa mãn.

2. Code tham khảo:

```
n,m=map(int,input().split())
k=list(map(int,input().split()))
s=[0]*(m+1)
kq=0
for i in k:
    if i<= m:
        kq+=s[m-i]
        s[i]+=1
print(kq)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm số lớn thứ hai

Cho một dãy số nguyên dương B gồm n phần tử. Hãy tìm số lớn thứ hai trong dãy số.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương phần tử của dãy số B (mỗi số không vượt quá 10^9).

Output:

- In ra số nguyên là số lớn thứ hai trong dãy số.

Ví dụ:

Input	Output
5 3 1 4 4 5	4

Bài 1.2: Số nhỏ nhất không xuất hiện

Cho một dãy số nguyên dương C gồm n phần tử. Hãy tìm số nguyên dương nhỏ nhất không xuất hiện trong dãy.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương phần tử của dãy số C (mỗi số không vượt quá 10^6).

Output:

- In ra số nguyên dương nhỏ nhất không xuất hiện trong dãy.

Ví dụ:

Input	Output
5 1 2 3 5 6	4

Bài 1.3: Đếm số lần xuất hiện

Cho một dãy số nguyên dương D gồm n phần tử và một số nguyên k . Hãy đếm số lần xuất hiện của k trong dãy.

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^6$).
- Dòng thứ hai chứa n số nguyên dương phần tử của dãy số D (mỗi số không vượt quá 10^9).

Output:

- In ra số lần xuất hiện của k trong dãy.

Ví dụ:

Input	Output
6 2 1 2 2 3 4 2	3

Bài 2: Số chẵn lớn nhất

1. Phân tích giải thuật:

1. Sắp xếp danh sách theo thứ tự giảm dần

2. Tạo ra 2 danh sách chẵn, lẻ để lưu những số chẵn, lẻ tương ứng trong danh sách. Kiểm tra nếu đã có ít nhất hai số chẵn và hai số lẻ, thì dừng vòng lặp sớm (để giảm thời gian chạy nếu đã tìm đủ số cần thiết).

3. Tạo ra biến tổng để lưu giá trị tổng lớn nhất của hai số chẵn hoặc hai số lẻ.

4. Cập nhật biến tổng nếu có ít nhất 2 số chẵn hoặc 2 số lẻ. Nếu tổng khác 0 in ra tổng, nếu tổng = 0 in ra -1

2. Code tham khảo

```
n=int(input())
a=list(map(int, input().split()))
a.sort(reverse=True)
chan=[]
le=[]
tong=0
for i in range(n):
    if a[i]%2==0:
        chan.append(a[i])
    else:
        le.append(a[i])
    if len(chan)>=2 and len(le)>=2:
        break
if len(chan)>=2:
    tong=chan[0]+chan[1]
if len(le)>=2:
    tong=max(tong, le[0]+le[1])
if tong!=0 :print (tong)
else:
    print (-1)
```

3. Bài tập mở rộng:

Bài 2.1: Số chính phương lớn nhất từ tổng hai số

Cho số nguyên dương n và dãy số a_1, a_2, \dots, a_n . Hãy tìm tổng lớn nhất của hai số khác nhau trong dãy sao cho tổng đó là một số chính phương.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($2 \leq n \leq 10^6$).

• Dòng thứ hai chứa dãy số a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$). Các số cách nhau một khoảng trắng.

Output:

• In ra tổng lớn nhất là số chính phương của hai phần tử khác nhau, hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
5 16 3 5 2 20	36

Bài 2.2: Tìm hai số có hiệu lớn nhất

Cho số nguyên dương n và dãy số a_1, a_2, \dots, a_n . Xác định hai số trong dãy sao cho hiệu của chúng là lớn nhất và là một số chẵn.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($2 \leq n \leq 10^6$).
- Dòng thứ hai chứa dãy số a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$). Các số cách nhau một khoảng trắng.

Output:

• In ra hiệu lớn nhất và là số chẵn của hai phần tử khác nhau, hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
6 10 2 7 5 14 20	18

Bài 2.3: Tìm tổng nhỏ nhất của ba phần tử

Cho số nguyên dương n và dãy số a_1, a_2, \dots, a_n . Hãy tìm tổng nhỏ nhất của ba phần tử khác nhau trong dãy sao cho tổng đó là một số nguyên tố.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($2 \leq n \leq 10^6$).
- Dòng thứ hai chứa dãy số a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$). Các số cách nhau một khoảng trắng.

Output:

● In ra tổng nhỏ nhất của ba phần tử khác nhau là số nguyên tố, hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
6 7 1 3 5 11 2	11

Bài 3: Xâu hoàn hảo

1. Phân tích giải thuật:

1. Khởi tạo biến kết quả để lưu trữ kết quả cuối cùng, tức tổng số cặp ký tự giống nhau trong chuỗi.

2. Duyệt vòng lặp chạy từ giá trị ASCII của ký tự 'a' đến 'z'. (Kiến thức liên quan: hàm ord() là một hàm trong Python trả về giá trị ASCII của một ký tự)

+ Chuyển giá trị ASCII thành ký tự tương ứng

+ Đếm số ký tự xuất hiện trong chuỗi

+ Cập nhật kết quả bằng cách thêm số cặp ký tự có thể hình thành, dựa trên công thức tổ hợp chập 2 của số lượng ký tự.

2. Code tham khảo

```
n=int(input())
s=input()
kq=0
for i in range(ord('a'),ord('z')+1):
    c=chr(i)
    so=s.count(c)
    kq=kq+so*(so-1)//2
print(kq)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm xâu con đối xứng

Cho một xâu ký tự S có độ dài N. Hãy đếm số lượng xâu con đối xứng trong xâu S. Xâu con đối xứng là xâu có thể đọc giống nhau từ trái qua phải và từ phải qua trái.

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu S ($1 \leq N \leq 10^6$).
- Dòng thứ hai là xâu S chỉ gồm các ký tự chữ cái Latinh thường.

Output:

- Ghi ra một số nguyên là số lượng xâu con đối xứng trong xâu S .

Ví dụ:

Input	Output
5 ababa	9

Bài 3.2: Xâu ký tự lặp lại

Cho một xâu ký tự T có độ dài N . Tìm xâu con lớn nhất xuất hiện nhiều lần trong xâu T .

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu T ($1 \leq N \leq 10^6$).
- Dòng thứ hai là xâu T chỉ gồm các ký tự chữ cái Latinh thường.

Output:

- Ghi ra xâu con lớn nhất xuất hiện nhiều hơn một lần. Nếu không có xâu con nào như vậy, in ra "-1".

Ví dụ:

Input	Output
7 abcbca	abc

Bài 3.3: Xâu có số ký tự khác nhau tối thiểu

Cho một xâu ký tự U có độ dài N . Tìm xâu con dài nhất có số ký tự khác nhau không quá K .

Input:

- Dòng đầu tiên là số nguyên dương N - độ dài của xâu T ($1 \leq N \leq 10^6$).
- Dòng thứ hai là số K ($1 \leq K \leq 26$).
- Dòng thứ ba là xâu U chỉ gồm các ký tự chữ cái Latinh thường.

Output:

- Ghi ra độ dài của xâu con dài nhất có số ký tự khác nhau không quá

K.

Ví dụ:

Input	Output
10	3
2	
aabcdeabcg	

Bài 4: Mật khẩu

1. Phân tích giải thuật

1. Sử dụng thư viện re (regular expression) để tìm tất cả các chuỗi con chứa các số (một hoặc nhiều chữ số) trong chuỗi. Kết quả được lưu trong danh sách mới dưới dạng các chuỗi số.

2. Tạo hàm tách các chữ số riêng biệt từ danh sách trên

3. Tính tổng và kết hợp độ dài của chuỗi

2. Code tham khảo

```
import re
s = input()
numbers = re.findall(r'\d+', s)
def split(numbers):
    ds = []
    for i in numbers:
        ds.extend([int(j) for j in str(i)])
    return ds
numbers = [int(i) for i in numbers]
list_number = split(numbers)
sum=0
for i in list_number:
    if i%2==0:
        sum+=i
print(str(sum) + str(len(s)))
```

3. Bài tập mở rộng:

Bài 4.1: Tổng số nguyên tố và số ký tự đặc biệt

Cho một xâu ký tự, tìm tổng các số nguyên tố trong xâu và ghép với số lượng ký tự đặc biệt (không phải chữ cái hay số).

Input:

- Một dòng duy nhất chứa xâu ký tự có thể bao gồm chữ cái, số, khoảng trắng và các ký tự đặc biệt. Độ dài xâu không quá 10^5 ký tự.

Output:

- Mật khẩu được tạo từ tổng các số nguyên tố trong xâu và số lượng ký tự đặc biệt.

Ví dụ:

Input	Output
abc 23 17 !@#	40 3

Bài 4.2: Số nguyên lớn nhất và số chữ cái

Cho một xâu ký tự, tìm số nguyên lớn nhất có trong xâu và ghép với số lượng chữ cái trong xâu.

Input:

- Một dòng duy nhất chứa xâu ký tự có thể bao gồm chữ cái, số, khoảng trắng và các ký tự đặc biệt. Độ dài xâu không quá 10^5 ký tự.

Output:

- Mật khẩu được tạo từ số nguyên lớn nhất trong xâu và số lượng chữ cái.

Ví dụ:

Input	Output
Hello 2024! Good 1995.	2024 9

Bài 4.3: Tổng các số chia hết cho 3 và số khoảng trắng

Cho một xâu ký tự, tìm tổng các số chia hết cho 3 trong xâu và ghép với số lượng khoảng trắng có trong xâu.

Input:

- Một dòng duy nhất chứa xâu ký tự có thể bao gồm chữ cái, số, khoảng trắng và các ký tự đặc biệt. Độ dài xâu không quá 10^5 ký tự.

Output:

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

- Mật khẩu được tạo từ tổng các số chia hết cho 3 trong xâu và số lượng khoảng trắng.

Ví dụ:

Input	Output
15 6 14 space 8	21 6

ĐỀ SỐ 13

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Xâu chẵn	STRING*	Bàn phím	Màn hình	4
Bài 2	Đếm k trong mảng 2 chiều	COUNT*	Bàn phím	Màn hình	5
Bài 3	Thập phân	ROUND*	Bàn phím	Màn hình	5
Bài 4	Ước số chung nhỏ nhất	DIVISOR*	Bàn phím	Màn hình	6

Bài 1: Xâu chẵn

Cho một xâu S được chỉ gồm các ký tự chữ cái thường 'a' ... 'z' được gọi là xâu chẵn nếu số lần xuất hiện của từng chữ cái trong xâu S là số chẵn.

Input

- Một dòng chứa duy nhất xâu S có số lượng ký tự không quá 255 ký tự.

Output

- Nếu xâu S là xâu chẵn thì in ra "Yes". Ngược lại thì in ra "No".

Ví dụ:

Input	Output
adccda	Yes
adcccdaa	No

Bài 2: Đếm k trong mảng hai chiều

Cho một bảng số a nguyên kích thước $m * n$ và số nguyên k. Hãy đếm xem trong dãy có bao nhiêu phần tử có giá trị k.

Input

- Dòng đầu tiên gồm 2 số nguyên dương m, n và số nguyên k: $m, n \leq 10^3$, $|k| \leq 10^3$, $|a[i][j]| \leq 10^3$

- m dòng tiếp theo, mỗi dòng gồm n số

Output

- Kết quả bài toán

Ví dụ:

Input	Output
2 3 4 2 4 6 6 4 2	2

Bài 3: Thập phân (Tin học trẻ C2 Đà Nẵng 2022)

Cho một số thực X có 4 chữ số sau dấu phẩy ($0 < X < 100$). Hãy làm tròn X trở thành số nguyên gần nhất và nhỏ nhất.

Input

- Dữ liệu đầu vào đảm bảo X có đúng 4 chữ số ở phần thập phân.

Output

- Số nguyên được làm tròn từ X.

Ví dụ:

Input	Output
0.1234	0
12.5000	12

Bài 4. Ước số chung nhỏ nhất

Ước số chung của dãy số nguyên dương là các số nguyên dương mà tất cả các số trong dãy đều chia hết cho nó. Có một dãy số A gồm N số nguyên dương, hãy tìm ước số chung nhỏ nhất khác 1. Nói cách khác, cần tìm số D nhỏ nhất, sao cho $D > 1$ và các số trong dãy số A đều chia hết cho số D này.

Input:

- Dòng đầu tiên chứa số nguyên dương N ($N \leq 10^5$)
- Dòng tiếp theo gồm N số nguyên dương A; là các phần tử của dãy A ($A_i \leq 10^6$).

Output:

- Một số nguyên dương ước chung nhỏ nhất của dãy số. Nếu không tồn tại số siêu nguyên dương nào, in ra -1.

Ví dụ:

Input	Output
3 1 2 3	-1
3 2 4 6	2

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Xâu chẵn

1. Phân tích giải thuật:

1. Khởi tạo một chuỗi các ký tự từ 'a' đến 'z'.
2. Duyệt qua từng ký tự trong a, đếm số lần xuất hiện các ký tự từ 'a' đến 'z' trong chuỗi s có lẽ không, nếu lẻ thoát vòng lặp, nếu không tiếp tục đếm.
3. Kiểm tra biến đếm và in kết quả.

2. Code tham khảo:

```
s = input()
a = 'abcdefghijklmnopqrstuvwxyz'
dem = 0
for i in a:
    if s.count(i)%2==1:
        dem=0
        break
    else:
        dem=1
if dem==1:
    print('Yes')
else:
    print('No')
```

3. Bài tập mở rộng:

Bài 1.1: Kiểm tra xâu chẵn trong khoảng con

Cho một xâu S chỉ gồm các ký tự chữ cái thường từ 'a' đến 'z'. Viết chương trình nhận input là một xâu S và hai số nguyên L và R ($0 \leq L \leq R < \text{độ dài của S}$). Kiểm tra xem xâu con từ L đến R có phải là xâu chẵn không.

Input:

- Dòng đầu tiên chứa xâu S.
- Dòng thứ hai chứa hai số nguyên L và R.

Output:

- Nếu xâu con từ L đến R là xâu chẵn, in ra "Yes". Ngược lại thì in ra "No".

Ví dụ:

Input	Output
abcdefabcde 2 8	No
zzxxccvvbb 2 7	Yes

Bài 1.2: Xâu con dài nhất

Cho một xâu S chỉ gồm các ký tự chữ cái thường từ 'a' đến 'z'. Tìm xâu con con dài nhất có thể tìm được trong S.

Input:

- Một dòng chứa xâu S.

Output:

- In ra độ dài của xâu con dài nhất.

Ví dụ:

Input	Output
abcabba	6

Bài 1.3: Tối ưu hóa xâu thành xâu con

Cho một xâu S chỉ gồm các ký tự chữ cái thường từ 'a' đến 'z'. Tìm số lượng tối thiểu các ký tự cần xóa để xâu S trở thành xâu con.

Input:

- Một dòng chứa xâu S.

Output:

- Số lượng ký tự tối thiểu cần xóa.

Ví dụ:

Input	Output
abcabc	0
aabbccdde	1

Bài 2: Đếm k trong mảng 2 chiều

1. Phân tích giải thuật:

1. Tạo 1 biến đếm.
2. Duyệt mảng 2 chiều, kiểm tra xem phần tử trong từng hàng có bằng k hay không, nếu bằng tăng biến đếm lên 1 đơn vị.
3. In biến đếm.

2. Code tham khảo

```
n, m, k = map(int, input().split())
d = 0
for i in range(n):
    row = list(map(int, input().split()))
    for j in range(m):
        if row[j] == k:
            d += 1
print(d)
```

3. Bài tập mở rộng:

Bài 2.1: Đếm phần tử lớn hơn k

Cho một bảng số nguyên a kích thước $m * n$ và số nguyên k. Hãy đếm xem trong bảng có bao nhiêu phần tử có giá trị lớn hơn k.

Input:

- Dòng đầu tiên chứa ba số nguyên m, n, và k.
- m dòng tiếp theo, mỗi dòng chứa n số nguyên.

Output:

- In ra số lượng phần tử có giá trị lớn hơn k.

Ví dụ:

Input	Output
2 3 4 5 3 7 8 2 4	3

Bài 2.2: Đếm phần tử bằng k theo hàng

Cho một bảng số nguyên a kích thước $m * n$ và số nguyên k. Đếm số lượng phần tử có giá trị bằng k trên mỗi hàng và in ra.

Input:

- Dòng đầu tiên chứa ba số nguyên m, n, và k.
- m dòng tiếp theo, mỗi dòng chứa n số nguyên.

Output:

- In ra m dòng, mỗi dòng là số lượng phần tử có giá trị bằng k trên hàng tương ứng.

Ví dụ:

Input	Output
2 3 2	2
2 3 2	0
3 1 4	

Bài 2.3: Đếm phần tử bằng k theo cột

Cho một bảng số nguyên a kích thước $m * n$ và số nguyên k. Đếm số lượng phần tử có giá trị bằng k trên mỗi cột và in ra.

Input:

- Dòng đầu tiên chứa ba số nguyên m, n, và k.
- m dòng tiếp theo, mỗi dòng chứa n số nguyên.

Output:

- In ra n dòng, mỗi dòng là số lượng phần tử có giá trị bằng k trên cột tương ứng.

Ví dụ:

Input	Output
3 2 1	1
1 0	2
2 1	
3 1	

Bài 3: Thập phân

1. Phân tích giải thuật:

- Cách 1:

1. So sánh x với $\text{int}(x) + 0.5$.
2. Nếu x chính xác bằng $\text{int}(x) + 0.5$, tức là x nằm đúng giữa hai số nguyên, in ra $\text{int}(x)$. Điều này xử lý các trường hợp như 2.5, 3.5, ở đây số này cần được làm tròn xuống.
3. Ngược lại, in ra giá trị nhỏ hơn trong hai giá trị: $\text{round}(x)$ và $\text{int}(x) + 1$. $\text{round}(x)$ làm tròn x đến số nguyên gần nhất theo quy tắc làm tròn tiêu chuẩn, và $\text{int}(x) + 1$ là số nguyên tiếp theo sau phần nguyên của x .

- Cách 2:

1. Làm tròn $a - 0.1$ rồi in ra.
2. Trừ 0.1 trước khi làm tròn có thể giúp thay đổi cách làm tròn của các số có phần thập phân là 0.5, vì theo quy tắc làm tròn của Python, nếu phần thập phân là chính xác 0.5, nó sẽ làm tròn đến số nguyên chẵn gần nhất. Trừ đi 0.1 trước khi làm tròn giúp đảm bảo các số như 2.5 được làm tròn xuống thành 2 thay vì lên thành 3.

2. Code tham khảo

- Cách 1:

```
x=float(input())
if x==int(x)+0.5:
    print(int(x))
else:
    print(min(round(x), int(x)+1))
```

- Cách 2:

```
a=float(input())
print(round(a-0.1))
```

3. Bài tập mở rộng:

Bài 3.1: Làm tròn thập phân theo ngưỡng

Cho một số thực X có 5 chữ số sau dấu phẩy ($0 < X < 100$). Hãy làm tròn X tới số nguyên gần nhất, nhưng chỉ làm tròn lên nếu phần thập phân sau làm tròn lớn hơn hoặc bằng 0.6.

Input:

- Một số thực X với đúng 5 chữ số ở phần thập phân.

Output:

- Ghi ra số nguyên được làm tròn từ X .

Ví dụ:

Input	Output
23.45678	23

Bài 3.2: Tìm số nguyên lớn nhất nhỏ hơn X

Cho một số thực X có đúng 3 chữ số sau dấu phẩy ($0 < X < 100$). Hãy tìm số nguyên lớn nhất nhưng vẫn nhỏ hơn X

Input:

- Một số thực X với đúng 3 chữ số ở phần thập phân.

Output:

- In ra số nguyên lớn nhất nhỏ hơn X .

Ví dụ:

Input	Output
45.123	45

Bài 3.3: Làm tròn với điều kiện

Cho một số thực X có 6 chữ số sau dấu phẩy ($0 < X < 100$). Làm tròn X đến số nguyên gần nhất, nhưng nếu phần thập phân thứ 6 là số lẻ, hãy giảm kết quả đi 1 đơn vị.

Input:

- Một số thực X với đúng 6 chữ số ở phần thập phân.

Output:

- In ra số nguyên được làm tròn từ X theo điều kiện đặc biệt.

Ví dụ:

Input	Output
12.345671	11

Bài 4: Ước số chung nhỏ nhất

1. Phân tích giải thuật

1. Nhập vào danh sách các số nguyên
2. Tìm ước chung lớn nhất của danh sách
3. Tìm ước nguyên tố nhỏ nhất nếu có
4. Nếu không có ước số nguyên tố nào khác ngoài 1, trả về -1; nếu không, trả về ước nguyên tố nhỏ nhất tìm được.

2. Code tham khảo

```
import math
def check(c):
    x = c[0]
    for i in range(1, len(c)):
        x = math.gcd(x, c[i])
    for i in range(2, int(math.sqrt(x))+1):
        if x%i==0:
            x=i
            break
    if x==1:
        x=-1
    return x
a=int(input())
n=list(map(int,input().split()))
kq = check(n)
print(kq)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm bội số chung nhỏ nhất của dãy

Cho dãy số nguyên dương A gồm N số, hãy tìm bội số chung nhỏ nhất (BCNN) của các số trong dãy và trả về kết quả sau khi lấy modulo 10^7 .

Input:

- Dòng đầu tiên chứa số nguyên dương N ($N \leq 10^5$).
- Dòng tiếp theo gồm N số nguyên dương A là các phần tử của dãy ($A_i \leq 10^6$).

Output:

- Một số nguyên dương là BCNN của dãy số, lấy kết quả modulo 10^7 .

Ví dụ:

Input	Output
3 3 5 7	105
4 100000 200000 300000 500000	3000000

Bài 4.2: Tính tổng ước số chung lớn nhất của từng cặp

Cho dãy số nguyên dương A gồm N số, hãy tính tổng ước số chung lớn nhất (UCLN) của từng cặp phần tử trong dãy.

Input:

- Dòng đầu tiên chứa số nguyên dương N ($N \leq 10^5$).
- Dòng tiếp theo gồm N số nguyên dương A là các phần tử của dãy ($A_i \leq 10^6$).

Output:

- Một số nguyên dương là tổng UCLN của từng cặp phần tử trong dãy.

Ví dụ:

Input	Output
3 6 8 12	16

Giải thích: UCLN của các cặp (6,8), (6,12), (8,12) lần lượt là 2, 6, 4. Tổng là 12)

Bài 4.3: Tìm các số chia hết cho ước số nguyên tố nhỏ nhất của dãy

Cho dãy số nguyên dương A gồm N số, hãy xác định và in ra các số trong dãy chia hết cho ước số nguyên tố nhỏ nhất của dãy.

Input:

- Dòng đầu tiên chứa số nguyên dương N ($N \leq 10^5$).
- Dòng tiếp theo gồm N số nguyên dương A là các phần tử của dãy ($A_i \leq 10^6$).

Output:

- In ra các số chia hết cho ước số nguyên tố nhỏ nhất của dãy, nếu không có số nào chia hết in ra -1.

Ví dụ:

Input	Output
3 15 25 35	15 25 35

Giải thích: ước số nguyên tố nhỏ nhất của dãy là 5.

ĐỀ SỐ 14
TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Luân lưu	SHOOTTOUT*	Bàn phím	Màn hình	4
Bài 2	Số ở giữa	CENTER*	Bàn phím	Màn hình	5
Bài 3	Dãy tăng nghiêm ngặt	INCREASE*	Bàn phím	Màn hình	5
Bài 4	Siêu nguyên tố	SUPER*	Bàn phím	Màn hình	6

Bài 1: Đá luân lưu (Tin học trẻ Đà Nẵng 2024 – Bảng C2)

Trong trận bóng đá trên sân nhà đội bóng AB FC đã có kết quả hòa với đội bạn trong 90 phút thi đấu chính thức. Vì thời gian thi đấu có hạn nên ban tổ chức trận đấu quyết định sử dụng loạt đá luân lưu để phân thắng bại. Có n quả đá luân lưu được sử dụng đến, theo quan sát lượt đá thứ i đội bóng AB FC sẽ thất bại nếu i là bội số của 3 và nếu ngược lại thì thành công.

Input

- Đọc từ tệp văn bản LUANLUU.INP một dòng duy nhất chứa số nguyên n ($1 \leq n \leq 10^6$).

Output

- Ghi ra tệp văn bản LUANLUU.OUT một chuỗi có độ dài n đại diện cho kết quả các lượt đá luân lưu của đội bóng AB FC. Kí tự thứ i ($1 \leq i \leq n$) sẽ là 'o' nếu thành công và 'x' nếu thất bại.

Ví dụ:

Input	Output
9	ooxooxoox

Bài 2: Số ở giữa

Cho 2 số nguyên A và B . Tìm số nguyên M nằm giữa A và B sao cho

khoảng cách giữa $A \times M$ và $B \times M$ là nhỏ nhất. M phải khác A và B .

Input

- Gồm 1 dòng duy nhất chứa hai số nguyên A và B ($-10^9 \leq A \leq B \leq 10^9$)
- Dữ liệu đảm bảo $A \leq B - 2$

Output

- Gồm 1 dòng duy nhất chứa số nguyên M cần tìm.

Ví dụ:

Input	Output
1 3	2

Bài 3: Dãy tăng nghiêm ngặt

Bạn được cho một dãy a gồm n số nguyên dương. Hỏi có cách nào sắp xếp dãy a thành dãy tăng nghiêm ngặt không?

Dãy tăng nghiêm ngặt có n phần tử là dãy có các phần tử thoả mãn điều kiện: $a_1 < a_2 < \dots < a_n$.

Input

- Dòng thứ nhất chứa một số nguyên dương T ($T \leq 10$) - số bộ test mà bạn phải trả lời.
- T bộ test tiếp theo có dạng như sau:
 - Dòng thứ nhất của bộ test chứa một số nguyên dương n ($n \leq 10^5$).
 - Dòng thứ hai của bộ test chứa n số nguyên dương a_1, a_2, \dots, a_n ($a_i \leq 10^6$).

Output

- Với mỗi bộ test, in ra YES nếu bạn có thể sắp xếp dãy số thoả mãn yêu cầu đề bài, ngược lại in ra NO.

Ví dụ:

Input	Output
2 4 1 1 1 1 5 8 7 1 3 4	NO YES

Bài 4. Siêu nguyên tố

Một số nguyên dương n được gọi là một số siêu nguyên tố nếu n là số nguyên tố và khi ta bỏ bao nhiêu chữ số tận cùng của n thì số tự nhiên mới tạo thành cũng là một số nguyên tố.

Ví dụ:

Số 317 là số siêu nguyên tố vì số 317 là số nguyên tố, số 31 (bỏ 1 chữ số tận cùng của 317) là số nguyên tố, số 3 (bỏ 2 chữ số tận cùng của 317) là số nguyên tố.

Số 61 không là số siêu nguyên tố vì số 6 (bỏ 1 chữ số tận cùng của 61) không là số nguyên tố.

Yêu cầu: Viết chương trình nhập vào từ bàn phím một số nguyên dương n ($0 < n < 10^9$) và in ra màn hình một từ khẳng định số n có phải là số siêu nguyên tố hay không.

Input:

- Số nguyên dương n nhập từ bàn phím ($0 < n < 10^9$).

Output:

- In ra màn hình một từ PHAI nếu n là số siêu nguyên tố; ngược lại, in ra màn hình một từ KHONG nếu n không phải là số siêu nguyên tố.

Ví dụ:

Input	Output
317	PHAI
61	KHONG

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Luân lưu

1. Phân tích giải thuật:

Cách 1:

1. Đọc file LUANLUU.INP lấy giá trị n .
2. Sử dụng vòng lặp for duyệt các số từ 1 đến n , nếu số nào chia hết cho 3 thì gán bằng 'x' ngược lại gán 'o' rồi đưa vào danh sách.
3. Đọc file LUANLUU.OUT in các phần tử trong danh sách. Lưu ý: Cách này có thể dùng với $n \leq 10^7$

Cách 2:

1. Đọc file LUANLUU.INP lấy giá trị n .
2. Sử dụng chia lấy nguyên và chia lấy dư để nhân chuỗi với các kí tự "oox" và "o".
3. Đọc file LUANLUU.OUT in các phần tử trong danh sách. Lưu ý: Cách này có thể dùng với $n \leq 10^8$

2. Code tham khảo:

Cách 1:

```
with open("LUANLUU.INP","r") as file:
    n = file.readline()

with open("LUANLUU.OUT","w") as file:
    ds=['x' if i % 3 == 0 else 'o' for i in range(1,
int(n) + 1)]
    file.write(' '.join(ds))
```

Cách 2:

```
with open("LUANLUU.INP","r") as file:
    n = file.readline()

with open("LUANLUU.OUT","w") as file:
    file.write((int(n)//3)*"oox"+"o"*(int(n)%3))
```

3. Bài tập mở rộng:

Bài 1.1: Xác định số ngày chẵn trong một tháng

Cho một tháng và năm, hãy xác định có bao nhiêu ngày chẵn trong tháng đó.

Input:

- Tháng và năm dưới dạng số nguyên (tháng 1-12, năm bất kỳ).

Output:

- In ra số lượng ngày chẵn trong tháng đó.

Ví dụ:

Input	Output
2 2024	14

Bài 1.2: Đếm số lần xuất hiện của từng phần tử trong mảng

Cho một mảng số nguyên, hãy đếm số lần xuất hiện của mỗi phần tử trong mảng đó.

Input:

- Một mảng số nguyên.

Output:

- Một danh sách các tuple, mỗi tuple chứa một phần tử và số lần xuất hiện của nó trong mảng.

Ví dụ:

Input	Output
[3, 5, 3, 3, 5]	[(3, 3), (5, 2)]

Bài 1.3: Xác định số đối xứng trong dãy số

Số đối xứng là số mà khi đọc từ trái sang phải hoặc từ phải sang trái đều giống nhau. Viết một hàm để xác định có bao nhiêu số đối xứng trong một danh sách số nguyên.

Input:

- Danh sách các số nguyên.

Output:

- Số lượng số đối xứng trong danh sách.

Ví dụ:

Input	Output
[123, 121, 454, 34543, 12321]	4

Bài 2: Số ở giữa

1. Phân tích giải thuật:

1. Nhận hai số nguyên từ người dùng.
2. Xác định điều kiện:
 - Nếu a âm và b dương, in '0'.
 - Nếu b không dương (bằng 0 hoặc âm), in b - 1.
 - Nếu a không âm (bằng 0 hoặc dương), in a + 1.

2. Code tham khảo

```
a,b=map(int,input().split())
if a<0 and b>0:
    print('0')
elif b<=0:
    print(b-1)
elif a>=0:
    print(a+1)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm số cân bằng trong dãy

Trong một mảng số nguyên, số cân bằng là chỉ số mà tại đó, tổng các phần tử ở bên trái của nó bằng tổng các phần tử ở bên phải của nó. Hãy tìm chỉ số cân bằng đầu tiên trong mảng. Nếu không tồn tại, trả về -1.

Input:

- Một dòng duy nhất chứa mảng số nguyên.

Output:

- In ra một dòng duy nhất chứa chỉ số (index) của số cân bằng đầu tiên trong mảng. Nếu không có chỉ số nào thỏa mãn, in ra -1.

Ví dụ:

Input	Output
[1, 7, 3, 6, 5, 6]	3

Giải thích: Chỉ số 3 (chỉ số bắt đầu từ 0) trong mảng là số 6, tại đó tổng các phần tử ở bên trái ($1+7+3$) bằng tổng các phần tử ở bên phải ($5+6$).

Bài 2.2: Đếm số cặp số nguyên tố cộng nhau

Cho một số nguyên N , tìm số lượng cặp số (i, j) mà i và j là số nguyên tố và $i + j = N$.

Input:

- Một số nguyên N .

Output:

- Số lượng cặp số thỏa mãn.

Ví dụ:

Input	Output
10	2

Giải thích: Có 2 cặp là $(3,7)$ và $(5,5)$.

Bài 2.3: Tìm số lớn nhất có thể tạo bằng cách hoán vị các chữ số

Cho một số nguyên không âm, hãy tìm số lớn nhất có thể tạo bằng cách hoán vị các chữ số của nó.

Input:

- Một số nguyên không âm.

Output:

- Số lớn nhất có thể tạo.

Ví dụ:

Input	Output
1120	2110
10000231002	32211010000

Bài 3: Dãy tăng nghiêm ngặt

1. Phân tích giải thuật:

Lập qua mỗi trường hợp kiểm tra:

1. Đọc số nguyên n.
2. Nhập danh sách a chứa n số nguyên.
3. Kiểm tra nếu có phần tử trùng lặp trong danh sách a:
 - Nếu có, in 'NO'.
 - Nếu không, in 'YES'.

2. Code tham khảo

```
T=int(input())
for i in range(T):
    n=int(input())
    a=list(map(int,input().split()))
    if len(set(a))!=len(a):
        print('NO')
    else:
        print('YES')
```

3. Bài tập mở rộng:

Bài 3.1: Sắp xếp theo bội số

Bạn được cung cấp một mảng số nguyên và một số K. Sắp xếp mảng sao cho các số là bội số của K được ưu tiên xếp trước và giữ nguyên thứ tự tương đối giữa chúng.

Input:

- Dãy số nguyên và số nguyên K.

Output:

- In ra mảng đã sắp xếp theo yêu cầu.

Ví dụ:

Input	Output
[1, 2, 3, 4, 5, 6] 2	[2, 4, 6, 1, 3, 5]

Bài 3.2: Tổng lớn nhất của dãy con không kề nhau

Cho một mảng số nguyên, tìm tổng lớn nhất mà bạn có thể đạt được bằng cách chọn các phần tử không kề nhau trong mảng.

Input:

- Một dãy số nguyên.

Output:

- Tổng lớn nhất của các phần tử không kề nhau.

Ví dụ:

Input	Output
[3, 2, 7, 10]	13

Bài 3.3: Tìm cặp số có khoảng cách nhỏ nhất

Cho một mảng số nguyên, tìm cặp số có khoảng cách (giá trị tuyệt đối của hiệu hai số) nhỏ nhất.

Input:

- Một mảng số nguyên.

Output:

- In ra khoảng cách nhỏ nhất giữa bất kỳ hai số nào trong mảng.

Ví dụ:

Input	Output
[4, 9, 1, 5, 6, 10]	1

Giải thích: Khoảng cách nhỏ nhất giữa 5 và 6

Bài 4: Siêu nguyên tố

1. Phân tích giải thuật

Định nghĩa hàm kiểm tra số nguyên tố:

1. Nhập chuỗi số n.
2. Khởi tạo biến check bằng 0.
3. Lặp qua mỗi tiền tố của n:

- Kiểm tra tiền tố đó khi chuyển sang số nguyên có phải là số nguyên tố không.

- Nếu không phải số nguyên tố, gán check bằng 1 và thoát vòng lặp.

4. In kết quả:

- "KHONG" nếu check bằng 1 (có tiền tố không phải số nguyên tố).

- "PHAI" nếu không có tiền tố nào không phải số nguyên tố.

2. Code tham khảo

```
def ktsnt(x):  
    if x <= 1:  
        return False  
    check = 0  
    for i in range(2, int(x**0.5)+1):  
        if x%i==0:  
            return False  
    return True  
n = input(); check = 0  
for i in range(1, len(n)+1):  
    if ktsnt(int(n[:i])) is False:  
        check = 1  
        break  
if check:  
    print("KHONG")  
else:  
    print("PHAI")
```

3. Bài tập mở rộng:

Bài 4.1: Số siêu nguyên tố mở rộng

Một số nguyên dương n được gọi là số siêu nguyên tố mở rộng nếu n là số nguyên tố và khi lấy bất kỳ chuỗi con nào của các chữ số của n (không chỉ là bỏ chữ số tận cùng) thì số mới tạo thành cũng là số nguyên tố.

Input:

- Số nguyên dương n nhập từ bàn phím ($0 < n < 10^9$).

Output:

- In ra "PHAI" nếu n là số siêu nguyên tố mở rộng; ngược lại, in ra "KHONG".

Ví dụ:

Input	Output
233	PHAI

Giải thích: Số 233 là PHAI vì, 2, 23, 233 đều là số nguyên tố.

Bài 4.2: Số nguyên tố kỳ lạ

Một số được gọi là số nguyên tố kỳ lạ nếu nó là số nguyên tố và tổng các chữ số của nó cũng là số nguyên tố.

Input:

- Số nguyên dương n nhập từ bàn phím ($0 < n < 10^9$).

Output:

- In ra "PHAI" nếu n là số nguyên tố kỳ lạ; ngược lại, in ra "KHONG".

Ví dụ:

Input	Output
23	PHAI

Giải thích: Số 23 là PHAI vì, $2+3=5$, 5 là số nguyên tố

Bài 4.3: Số nguyên tố mạnh

Một số nguyên tố được gọi là mạnh nếu nó lớn hơn trung bình cộng của số nguyên tố trước và sau nó.

Input:

- Số nguyên dương n nhập từ bàn phím ($0 < n < 10^9$).

Output:

- In ra "PHAI" nếu n là số nguyên tố mạnh; ngược lại, in ra "KHONG".

Ví dụ:

Input	Output
29	PHAI

Giải thích: Số 29 là PHAI vì 29 là số nguyên tố và lớn hơn trung bình cộng của 23 và 31

ĐỀ SỐ 15

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Chia xúc xích	DIVIDE*	Bàn phím	Màn hình	4
Bài 2	Siêu đối xứng	SUPER*	Bàn phím	Màn hình	5
Bài 3	Cặp số	PAIR*	Bàn phím	Màn hình	5
Bài 4	Số nguyên tố cân bằng	BALANCE*	Bàn phím	Màn hình	6

Bài 1: Chia xúc xích

Bi có n thanh xúc xích với chiều dài bằng nhau, hôm nay Bi tổ chức tiệc nhỏ cho m người bạn của mình và có một vấn đề cần giải quyết đó là làm sao cắt n thanh xúc xích trên để chia đều cho m người bạn, và để các thanh xúc xích đẹp khi chia đều thì cần cắt với ít lần cắt nhất có thể. Xét ví dụ $n = 2$, $m = 8$ mỗi xúc xích cắt 3 nhát ta có được 8 phần bằng nhau và chia đều cho 8 người, vậy mất 6 nhát cắt. Hãy lập trình giúp Bi giải quyết vấn đề trên.

Input

- Dòng duy nhất chứa hai số nguyên là n , m thỏa $1 \leq n, m \leq 100$.

Output

- In ra số lát cắt ít nhất có thể.

Ví dụ:

Input	Output
2 8	6
5 9	8
4 2	0

Bài 2: Siêu đối xứng (THT Đà Nẵng – Bảng B 2022)

Một số nguyên dương được gọi là siêu đối xứng nếu tất cả các chữ số của

nó giống nhau. Chẳng hạn số 777 hoặc 4444 là các số nguyên dương siêu đối xứng. Nhập từ bàn phím một số nguyên dương x . Hãy tìm và in ra màn hình số nguyên dương y nhỏ nhất sao cho tổng $x + y$ là một số nguyên dương siêu đối xứng.

Input

- Gồm 1 dòng duy nhất chứa số nguyên dương x ($1 \leq x \leq 10^{16}$).

Output

- In ra số nguyên dương y nhỏ nhất thỏa mãn.

Ví dụ:

Input	Output
45	10

Giải thích: $y = 10$ vì $45 + 10 = 55$

Bài 3: Cặp số

Cho bốn số nguyên dương A, B, C, D . Alice cần đếm xem có bao nhiêu cặp số (X, Y) thỏa mãn các điều kiện sau:

- $A \leq X < B$;
- $C \leq Y \leq D$;
- $X < Y$.

Input

- Dòng duy nhất chứa bốn số nguyên dương A, B, C, D thỏa ($B, D \leq 10^9$).

Output

- In ra số cần đếm.

Ví dụ:

Input	Output
2 4 4 5	5

Giải thích: Các cặp thỏa mãn là $(2, 4), (2, 5), (3, 4), (3, 5), (4, 5)$.

Bài 4. Số nguyên tố cân bằng

Một số được gọi là số nguyên tố cân bằng nếu nó là số nguyên tố có $2k + 1$ chữ số ($k \in \mathbb{N}^*$), trong đó có $2k$ chữ số giống nhau và có đúng 1 chữ số ở vị trí chính giữa (tức vị trí thứ $k + 1$ từ trái sang phải) là khác với các chữ số còn lại.

Ví dụ: Số 7778777 là số cân bằng.

Input:

- Nhập từ bàn phím 1 số nguyên dương k ($k \leq 7$).

Output:

- Hãy tính và in ra màn hình số lượng các số nguyên tố cân bằng có $2k + 1$ chữ số.

Ví dụ:

Input	Output
3	7

Giải thích: Có 7 số nguyên dương có $2 \times 3 + 1$ chữ số là số nguyên tố: 1114111; 1117111; 3331333; 3337333; 7772777; 7774777; 7778777

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Chia xúc xích

1. Phân tích giải thuật:

1. Tạo hàm tìm ước chung lớn nhất của 2 số nguyên: $ucln(n,m)$
2. Tính và in số lần cắt ít nhất: $m - UCLN(n,m)$

- Ý tưởng: có thể phân chia mỗi thanh xúc xích thành g nhóm, với g là UCLN của n và m , thì tổng số phần khi chia tất cả các thanh là ng . Để chia đều cho m người, mỗi người nhận được ng / m phần. Khi g bằng m , không cần cắt thêm (vì mỗi người sẽ nhận một thanh nguyên vẹn).

- Số lần cắt cần thiết sẽ là số phần cần có trừ đi số nhóm có thể tạo ra mà không cần cắt thêm, là $m - g$. Do đó, $m - ucln(n, m)$ in ra số lần cắt cần thiết để đạt được phân chia mong muốn.

2. Code tham khảo:

```
n, m = map(int, input().split())

def ucln(n,m):
    if m == 0:
        return n
    else:
        return ucln(m, n % m)

print(m - ucln(n,m))
```

3. Bài tập mở rộng:

Bài 1.1: Chia bánh pizza

Bạn có n chiếc bánh pizza và muốn chia chúng cho m người bạn sao cho mỗi người nhận được số miếng bánh bằng nhau. Mỗi bánh pizza có thể được cắt thành các miếng. Tìm số lần cắt ít nhất để chia đều bánh cho mọi người.

Input:

- Một dòng chứa hai số nguyên n và m ($1 \leq n, m \leq 100$).

Output:

- Số lần cắt ít nhất.

Ví dụ:

Input	Output
3 10	4

Bài 1.2: Chia công việc

Bạn có n công việc cần phải hoàn thành và m nhân viên. Mỗi công việc có thể được chia nhỏ thành các tác vụ nhỏ hơn để phân công cho nhân viên. Hãy tìm số lần phân chia công việc ít nhất để mỗi nhân viên có số tác vụ bằng nhau.

Input:

- Một dòng chứa hai số nguyên n và m ($1 \leq n, m \leq 100$).

Output:

- Số lần phân chia công việc ít nhất.

Ví dụ:

Input	Output
10 4	1

Bài 1.3: Chia quà

Bì có n món quà và muốn phân chia đều cho m bạn bè. Mỗi món quà có thể được đóng gói lại thành các gói nhỏ hơn. Hãy xác định số lần đóng gói lại ít nhất để chia đều quà cho mọi người.

Input:

- Một dòng chứa hai số nguyên n và m ($1 \leq n, m \leq 100$).

Output:

- Số lần đóng gói lại ít nhất.

Ví dụ:

Input	Output
8 5	2

Bài 2: Siêu đối xứng

1. Phân tích giải thuật:

1. Khởi tạo 1 danh sách rỗng

2. Tạo số siêu đối xứng, dùng hai vòng lặp lồng nhau: vòng ngoài chạy từ 1 đến 9 để định chữ số cơ bản, và vòng trong từ 1 đến 16 để tạo ra số siêu đối xứng từ chữ số đó, bằng cách lặp lại và thêm chữ số vào cuối.

3. Tìm và in y:

+ Sắp xếp danh sách.

+ Duyệt qua danh sách, tìm số siêu đối xứng đầu tiên lớn hơn n và tính y bằng cách lấy số đó trừ đi n.

+ In giá trị của y và dừng vòng lặp ngay sau khi tìm thấy.

2. Code tham khảo

```
a = []
n = int(input())
for i in range(1, 10):
    x = i
    for j in range(1, 17):
        x = 10 * x + i
        a.append(x)
a.sort()
for i in a:
    if i > n:
        print(i-n)
        break
```

3. Bài tập mở rộng:

Bài 2.1: Số nguyên tố đối xứng nhỏ nhất

Nhập từ bàn phím một số nguyên dương x. Hãy tìm và in ra màn hình số nguyên dương y nhỏ nhất sao cho $x + y$ là một số nguyên tố đối xứng.

Input:

- Một dòng chứa số nguyên dương x ($1 \leq x \leq 10^6$).

Output:

- In ra số nguyên dương y nhỏ nhất thoả mãn.

Ví dụ:

Input	Output
89	12

Giải thích: $y = 12$ vì $89 + 12 = 101$, và 101 là một số nguyên tố đối xứng.

Bài 2.2: Tìm số chia hết nhỏ nhất

Nhập từ bàn phím một số nguyên dương x . Hãy tìm và in ra màn hình số nguyên dương y nhỏ nhất sao cho $x + y$ là số nguyên tố gần nhất lớn hơn x .

Input:

- Một dòng chứa số nguyên dương x ($1 \leq x \leq 10^6$).

Output:

- In ra số nguyên dương y nhỏ nhất thoả mãn.

Ví dụ:

Input	Output
24	6

Giải thích: $y = 6$ vì $24 + 6 = 30$, và 30 chia hết cho 10

Bài 2.3: Tìm số lũy thừa của 3 gần nhất

Nhập từ bàn phím một số nguyên dương x . Hãy tìm và in ra màn hình số nguyên dương y nhỏ nhất sao cho tổng của x và y là lũy thừa của 3 gần nhất lớn hơn x .

Input:

- Một dòng chứa số nguyên dương x ($1 \leq x \leq 10^6$).

Output:

- In ra số nguyên dương y nhỏ nhất thoả mãn.

Ví dụ:

Input	Output
25	2

Giải thích: $y = 2$ vì $25 + 2 = 27$, và $27 = 3^3$

Bài 3: Cặp số

1. Phân tích giải thuật:

1. Khởi tạo các biến:

- count: biến đếm số cặp thỏa mãn.
- l1: số lượng các giá trị khả dụng cho X từ A đến B-1.
- l2: số lượng các giá trị khả dụng cho Y từ C đến D.
- t: biến tạm để tính số cặp cần loại trừ khi X không nhỏ hơn Y.

2. Xét trường hợp không có cặp số thỏa mãn điều kiện 3 ($X < Y$): Nếu giá trị lớn nhất của Y (D) không lớn hơn giá trị nhỏ nhất của X (A), không có cặp nào có thể thỏa mãn $X < Y$, do đó kết quả là 0.

3. Tính toán số cặp số khi các khoảng có giao nhau.

4. Tính tổng số cặp thỏa mãn.

2. Code tham khảo

```
a,b,c,d = map(int,input().split())
count = 0
l1 = b-a +1
l2 = d-c +1
t = 0

if d <= a:
    print(0)
else:
    if b >= c:
        t = b-c+1
        if c < a:
            t -= a-c
        if d < b:
            t -= b-d
    if t == 0:
        count = l2*l1
    else:
        count = (l1-t)*l2
        count += round((2*d-2*a-2*l1+t+1)*(t/2))
```

```
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Chọn số may mắn

Cho hai khoảng số nguyên dương $[A, B]$ và $[C, D]$. Hãy đếm xem có bao nhiêu cặp số (X, Y) thỏa mãn các điều kiện sau: $A \leq X \leq B$; $C \leq Y \leq D$; X là ước của Y .

Input:

- Một dòng duy nhất chứa bốn số nguyên dương A, B, C, D .

Output:

- In ra số lượng cặp số thỏa mãn.

Ví dụ:

Input	Output
1 4 1 10	18

Bài 3.2: Sắp xếp khoảng

Cho hai khoảng số nguyên dương $[A, B]$ và $[C, D]$. Hãy tìm số nguyên dương nhỏ nhất X sao cho khi cộng X vào mỗi số trong khoảng $[A, B]$, khoảng mới này không giao nhau với khoảng $[C, D]$.

Input:

- Một dòng duy nhất chứa bốn số nguyên dương A, B, C, D .

Output:

- Số nguyên dương X nhỏ nhất.

Ví dụ:

Input	Output
3 5 6 8	2

Bài 3.3: Đếm số nguyên tố

Cho hai khoảng số nguyên dương $[A, B]$ và $[C, D]$. Hãy đếm xem có bao nhiêu cặp số (X, Y) thỏa mãn các điều kiện sau: $A \leq X \leq B$; $C \leq Y \leq D$; X và Y đều là số nguyên tố.

Input:

- Một dòng duy nhất chứa bốn số nguyên dương A, B, C, D.

Output:

- In ra số lượng cặp số thỏa mãn.

Ví dụ:

Input	Output
2 10 5 15	14

Bài 4: Số nguyên tố cân bằng

1. Phân tích giải thuật

1. Tạo hàm kiểm tra số nguyên tố
2. Tạo và kiểm tra các số cân bằng:
 - Kiểm tra nếu $(k - 1) // 2 = 7$ thì trực tiếp in 7;
 - Nếu không, sử dụng vòng lặp với i chạy từ 1 đến 9, chỉ xét các số lẻ (vì số chẵn sẽ không thể là số nguyên tố khi có hơn 2 chữ số);
 - Trong mỗi lần lặp, xây dựng số bằng cách thay đổi chữ số giữa từ i sang các giá trị từ 0 đến 9, trừ trường hợp giá trị đó bằng i hoặc số tạo thành khi chia hết cho 3 (không thể là số nguyên tố);
 - Kiểm tra mỗi số có phải là số nguyên tố bằng hàm và đếm nếu đúng.
3. In kết quả

2. Code tham khảo

```
def isprime(n):  
    if n == 2 or n == 3: return True  
    if n % 2 == 0 or n % 3 == 0 or n < 2: return  
False  
    i = 5  
    while i * i <= n:  
        if n % i == 0 or n % (i + 2) == 0: return  
False  
        i += 6  
    return True  
k, dem, goc, i = int(input()), 0, 0, 1  
congthem = 10 ** k
```

```
k = k * 2 + 1
for x in range(k): goc = goc * 10 + 1
if (k - 1) // 2 == 7: print(7)
else:
    while(i <= 9):
        so = goc * i - conghem * (i + 1)
        for x in range(10):
            so += conghem
            if x == i or (i * (k - 1) + x) % 3 == 0:
                continue
            if isprime(so): dem += 1
        i += 2
    print(dem)
```

3. Bài tập mở rộng:

Bài 4.1: Số nguyên tố kết thúc bằng chữ số đặc biệt

Tìm số lượng số nguyên tố có n chữ số kết thúc bằng chữ số m.

Giải thích: Yêu cầu số đó phải kết thúc bằng một chữ số cụ thể. Ví dụ, nếu $m = 7$, các số như 157 và 967 ... là số nguyên tố kết thúc bằng 7.

Input:

- Một số nguyên dương k ($k \leq 7$).

Output:

- In ra số lượng các số nguyên tố thỏa mãn.

Ví dụ:

Input	Output
3 7	42

Bài 4.2: Số nguyên tố có tổng chữ số nhất định

Cho số nguyên dương n, S. Tìm số lượng số nguyên tố số lượng chữ số là n và tổng là S.

Input:

- Hai số nguyên dương n, S

Output:

- In ra số lượng các số nguyên tố thỏa mãn.

Ví dụ:

Input	Output
3 10	1 2

Bài 4.3: Số nguyên tố có tích chữ số nhất định

Cho số nguyên dương n, S . Tìm số lượng số nguyên tố số lượng chữ số là n và tổng là S .

Input:

- Hai số nguyên dương n, S .

Output:

- In ra số lượng các số nguyên tố thỏa mãn.

Ví dụ:

Input	Output
3 24	2

Giải thích: Có 2 số nguyên tố có 3 chữ số và tích các chữ số là 24 là 239 và 389.

ĐỀ SỐ 16

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Đếm số lượng chữ số trong danh sách	COUNTDIGITS*	Bàn phím	Màn hình	4
Bài 2	Tìm dãy con liên tiếp có tổng chia hết cho k	SUMDIVISIBLEK*	Bàn phím	Màn hình	5
Bài 3	Đếm số lượng từ có số lượng ký tự chẵn	COUNTEVENWORDS*	Bàn phím	Màn hình	5
Bài 4	Tìm từ dài nhất không chứa ký tự lặp lại	LONGESTUNIQUEWORD*	Bàn phím	Màn hình	6

Bài 1: Đếm số lượng chữ số trong danh sách

Cho một danh sách các số nguyên dương. Hãy đếm số lượng chữ số xuất hiện trong danh sách.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra số lượng chữ số (0-9) xuất hiện trong danh sách.

Ví dụ:

Input	Output
5 123 456 789 101 202	5

Bài 2: Tìm dãy con liên tiếp có tổng chia hết cho k

Cho một dãy số nguyên. Hãy tìm dãy con liên tiếp có tổng chia hết cho k và dài nhất.

Input

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).

- Dòng thứ hai chứa n số nguyên.

Output

- In ra tổng của dãy con liên tiếp dài nhất mà tổng chia hết cho k . Nếu có nhiều dãy con có cùng độ dài, in ra dãy con có tổng lớn nhất.

Ví dụ:

Input	Output
6 3	12
2 3 4 5 6 1	

Bài 3: Đếm số lượng từ có số lượng ký tự chẵn

Cho một chuỗi ký tự chứa các từ được phân cách bởi khoảng trắng. Hãy đếm số lượng từ có số lượng ký tự chẵn.

Input

- Một dòng duy nhất chứa chuỗi ký tự S (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng từ có số lượng ký tự chẵn.

Ví dụ:

Input	Output
hello world abcde	1

Bài 4. Tìm từ dài nhất không chứa ký tự lặp lại

Cho một chuỗi ký tự S . Hãy tìm từ dài nhất không chứa ký tự lặp lại.

Input:

- Một dòng duy nhất chứa chuỗi ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất không chứa ký tự lặp lại. Nếu có nhiều từ có cùng độ dài, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
hello abcde world	abcde

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Đếm số lượng chữ số trong danh sách

1. Phân tích giải thuật:

1. Duyệt qua từng số trong danh sách và tách từng chữ số.
2. Đếm tổng số lượng chữ số xuất hiện.

2. Code tham khảo:

```
n = int(input())
arr = list(map(int, input().split()))

digit_count = 0

for num in arr:
    digit_count += len(str(num))

print(digit_count)
```

3. Bài tập mở rộng:

Bài 1.1: Đếm số lượng số nguyên tố có chữ số bằng n

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lượng số nguyên tố trong danh sách có đúng n chữ số.

Ví dụ:

Input	Output
5	1
123 456 789 101 202	

Bài 1.2: Đếm số lượng số nguyên dương chia hết cho 5 và có số chữ số lẻ

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số lượng số nguyên dương chia hết cho 5 và có số chữ số lẻ.

Ví dụ:

Input	Output
5 15 30 25 40 100	2

Bài 1.3: Tìm chữ số xuất hiện nhiều nhất trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra chữ số xuất hiện nhiều nhất trong danh sách. Nếu có nhiều chữ số cùng xuất hiện nhiều nhất, in ra chữ số lớn nhất.

Ví dụ:

Input	Output
6 555 123 444 555 333 222	5

Bài 2: Tìm dãy con liên tiếp có tổng chia hết cho k

Cho một dãy số nguyên. Hãy tìm dãy con liên tiếp có tổng chia hết cho k và dài nhất.

1. Phân tích giải thuật:

1. Sử dụng mảng cộng dồn (cumulative sum) để tính tổng của các dãy con liên tiếp.

2. Duyệt qua dãy số và kiểm tra tổng có chia hết cho k không.

2. Code tham khảo

```
n, k = map(int, input().split())
arr = list(map(int, input().split()))

max_sum = 0
current_sum = 0
start = 0
```

```
for i in range(n):
    current_sum += arr[i]
    if current_sum % k == 0:
        max_sum = max(max_sum, current_sum)
    else:
        start = i + 1
        current_sum = 0

print(max_sum)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm dãy con liên tiếp có tổng chia hết cho k và ngắn nhất

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng của dãy con liên tiếp ngắn nhất mà tổng chia hết cho k.

Ví dụ:

Input	Output
6 3 2 3 4 5 6 1	3

Bài 2.2: Tìm dãy con liên tiếp có tổng chia hết cho k mà không chứa số âm

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng của dãy con liên tiếp mà tổng chia hết cho k và không chứa số âm.

Ví dụ:

Input	Output
6 3	15

2 3 4 5 6 1	
-------------	--

Bài 2.3: Tìm dãy con liên tiếp có tổng chia hết cho k với số lượng phần tử lẻ

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng của dãy con liên tiếp mà tổng chia hết cho k và có số lượng phần tử lẻ.

Ví dụ:

Input	Output
6 3 1 2 3 4 5 6	9

Bài 3: Đếm số lượng từ có số lượng ký tự chẵn

1. Phân tích giải thuật:

1. Sử dụng phương pháp tách từ bằng cách sử dụng hàm split().
2. Duyệt qua từng từ và kiểm tra số lượng ký tự. Đếm số lượng từ có số lượng ký tự chẵn.

2. Code tham khảo

```
s = input().split()
count = sum(1 for word in s if len(word) % 2 == 0)
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng từ có số lượng ký tự lẻ

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ có số lượng ký tự lẻ.

Ví dụ:

Input	Output
hello world abcde	2

Bài 3.2: Tìm từ dài nhất có số lượng ký tự chẵn trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất có số lượng ký tự chẵn trong xâu. Nếu có nhiều từ có cùng độ dài, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
hello wonderful abcde	wonderful

Bài 3.3: Đếm số lượng từ có số lượng ký tự chẵn và không chứa ký tự lặp lại

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ có số lượng ký tự chẵn và không chứa ký tự lặp lại.

Ví dụ:

Input	Output
hello world abcde	0

Bài 4: Tìm từ dài nhất không chứa ký tự lặp lại

1. Phân tích giải thuật

1. Tách xâu thành các từ bằng cách sử dụng hàm `split()`.
2. Kiểm tra từng từ xem có ký tự lặp lại không. Tìm từ dài nhất thỏa mãn điều kiện.

2. Code tham khảo

```
s = input().split()
max_word = ""
```

```
for word in s:
    if len(word) == len(set(word)):
        if len(word) > len(max_word):
            max_word = word

print(max_word)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm từ dài nhất có chứa tất cả các ký tự là nguyên âm

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất có chứa tất cả các ký tự là nguyên âm. Nếu có nhiều từ có cùng độ dài, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
education umbrella	education

Bài 4.2: Tìm từ dài nhất có chứa cả ký tự viết hoa và viết thường

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất có chứa cả ký tự viết hoa và viết thường. Nếu có nhiều từ có cùng độ dài, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
Hello World AbCdeF	AbCdeF

Bài 4.3: Tìm từ dài nhất mà không chứa ký tự nào là số

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất mà không chứa ký tự nào là số. Nếu có nhiều từ có cùng độ dài, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
hello world abc123	hello

ĐỀ SỐ 17

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tìm số lớn thứ năm trong danh sách	FIFTHLARGEST*	Bàn phím	Màn hình	4
Bài 2	Tìm dãy con liên tiếp có tổng bằng một giá trị cho trước	SUMEQUALTOX*	Bàn phím	Màn hình	5
Bài 3	Đếm số lượng từ có chứa số lượng nguyên âm lớn nhất	MAXVOWELS*	Bàn phím	Màn hình	5
Bài 4	Tìm dãy con không chứa số nguyên tố dài nhất	LONGESTNONPRIMESUBARRAY*	Bàn phím	Màn hình	6

Bài 1: Tìm số lớn thứ năm trong danh sách

Cho một danh sách các số nguyên dương. Hãy tìm phần tử lớn thứ năm trong danh sách. Nếu không có phần tử lớn thứ năm, in ra -1.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra phần tử lớn thứ năm trong danh sách hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
7	2
7 1 4 3 2 5 6	

Bài 2: Tìm dãy con liên tiếp có tổng bằng một giá trị cho trước

Cho một dãy số nguyên và một số nguyên dương x . Hãy tìm dãy con liên tiếp có tổng bằng x .

Input

- Dòng đầu tiên chứa hai số nguyên dương n và x ($1 \leq n \leq 10^5$, $1 \leq x \leq 10^9$).
- Dòng thứ hai chứa n số nguyên.

Output

- In ra dãy con liên tiếp có tổng bằng x . Nếu không tìm thấy, in ra -1.

Ví dụ:

Input	Output
6 9	3 2 4
2 3 4 5 6 7	

Bài 3: Đếm số lượng từ có chứa số lượng nguyên âm lớn nhất

Cho một chuỗi ký tự chứa các từ được phân cách bởi khoảng trắng. Hãy đếm số lượng từ có chứa số lượng nguyên âm lớn nhất.

Input

- Một dòng duy nhất chứa chuỗi ký tự S (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng từ có chứa số lượng nguyên âm lớn nhất.

Ví dụ:

Input	Output
hello education umbrella	1

Bài 4. Tìm dãy con không chứa số nguyên tố dài nhất

Cho một dãy số nguyên. Hãy tìm dãy con liên tiếp không chứa số nguyên tố và có độ dài lớn nhất.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra độ dài của dãy con liên tiếp không chứa số nguyên tố và có độ dài lớn nhất.

Ví dụ:

Input	Output
10 4 6 8 9 10 3 4 6 8 12	5

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tìm số lớn thứ năm trong danh sách

1. Phân tích giải thuật:

1. Duyệt qua danh sách để xác định phần tử lớn nhất, lớn thứ hai, lớn thứ ba, lớn thứ tư và lớn thứ năm.

2. Nếu không có đủ năm phần tử khác nhau, in ra -1.

2. Code tham khảo:

```
n = int(input())
arr = list(map(int, input().split()))

first = second = third = fourth = fifth = float('-inf')

for num in arr:
    if num > first:
        fifth = fourth
        fourth = third
        third = second
        second = first
        first = num
    elif num > second and num != first:
        fifth = fourth
        fourth = third
        third = second
        second = num
    elif num > third and num != second and num != first:
        fifth = fourth
        fourth = third
        third = num
    elif num > fourth and num != third and num != second and num != first:
        fifth = fourth
        fourth = num
    elif num > fifth and num != fourth and num != third and num != second and num != first:
        fifth = num

print(fifth if fifth != float('-inf') else -1)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm phần tử nhỏ thứ năm trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra phần tử nhỏ thứ năm trong danh sách hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
7 7 1 4 3 2 5 6	3

Bài 1.2: Tìm phần tử lớn thứ năm mà là số nguyên tố trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra phần tử lớn thứ năm là số nguyên tố trong danh sách hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
8 11 13 5 7 17 19 3 2	3

Bài 1.3: Tìm phần tử lớn thứ năm mà chia hết cho k trong danh sách

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra phần tử lớn thứ năm mà chia hết cho k trong danh sách hoặc -1 nếu không tìm thấy.

Ví dụ:

Input	Output
10 3	3
15 9 12 6 18 3 21 24 27 30	

Bài 2: Tìm dãy con liên tiếp có tổng bằng một giá trị cho trước

1. Phân tích giải thuật:

1. Sử dụng mảng cộng dồn (cumulative sum) và hai con trỏ để tìm dãy con có tổng bằng x.

2. Duyệt qua dãy số và kiểm tra tổng của các dãy con liên tiếp.

2. Code tham khảo

```
n, x = map(int, input().split())
arr = list(map(int, input().split()))

current_sum = 0
start = 0

for i in range(n):
    current_sum += arr[i]
    while current_sum > x and start <= i:
        current_sum -= arr[start]
        start += 1
    if current_sum == x:
        print(*arr[start:i+1])
        break
else:
    print(-1)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm dãy con liên tiếp có tổng nhỏ nhất lớn hơn hoặc bằng x

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và x ($1 \leq n \leq 10^5$, $1 \leq x \leq 10^9$).

- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng nhỏ nhất của dãy con liên tiếp mà lớn hơn hoặc bằng x .

Ví dụ:

Input	Output
6 10 2 3 4 5 6 7	10

Bài 2.2: Tìm dãy con liên tiếp có tổng nhỏ nhất bằng x

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và x ($1 \leq n \leq 10^5$, $1 \leq x \leq 10^9$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tổng nhỏ nhất của dãy con liên tiếp bằng x . Nếu không tìm thấy, in ra -1.

Ví dụ:

Input	Output
6 9 2 3 4 5 6 7	9

Bài 2.3: Tìm dãy con liên tiếp có tổng lớn nhất bằng x mà không có số âm

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và x ($1 \leq n \leq 10^5$, $1 \leq x \leq 10^9$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra dãy con liên tiếp có tổng lớn nhất bằng x mà không có số âm. Nếu không tìm thấy, in ra -1.

Ví dụ:

Input	Output
6 10	3 7

3 7 -2 5 1 -6	
---------------	--

Bài 3: Đếm số lượng từ có chứa số lượng nguyên âm lớn nhất

1. Phân tích giải thuật:

1. Sử dụng phương pháp tách từ bằng cách sử dụng hàm `split()`.
2. Duyệt qua từng từ và đếm số lượng nguyên âm trong từ đó. Tìm từ có số lượng nguyên âm lớn nhất và đếm số lượng từ thỏa mãn điều kiện này.

2. Code tham khảo

```
s = input().split()
vowels = set("aeiou")
max_vowels = 0
count = 0

for word in s:
    vowel_count = sum(1 for char in word if char in vowels)
    if vowel_count > max_vowels:
        max_vowels = vowel_count
        count = 1
    elif vowel_count == max_vowels:
        count += 1

print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Tìm từ có số lượng nguyên âm ít nhất trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra từ có số lượng nguyên âm ít nhất trong xâu. Nếu có nhiều từ có cùng số lượng nguyên âm, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
hello world umbrella	world

Bài 3.2: Đếm số lượng từ có chứa tất cả các nguyên âm

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ có chứa tất cả các nguyên âm (a, e, i, o, u).

Ví dụ:

Input	Output
hello education umbrella	1

Bài 3.3: Tìm từ có số lượng nguyên âm lớn nhất mà không chứa ký tự lặp lại

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra từ có số lượng nguyên âm lớn nhất mà không chứa ký tự lặp lại. Nếu có nhiều từ có cùng số lượng nguyên âm, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
education abcde umbrella	abcde

Bài 4: Tìm dãy con không chứa số nguyên tố dài nhất

1. Phân tích giải thuật

1. Sử dụng thuật toán phân loại số nguyên tố và số không nguyên tố.
2. Duyệt qua dãy số và tìm dãy con liên tiếp không chứa số nguyên tố.

2. Code tham khảo

```
def is_prime(x):  
    if x < 2:  
        return False
```

```
def is_prime(x):
    for i in range(2, int(x ** 0.5) + 1):
        if x % i == 0:
            return False
    return True

n = int(input())
arr = list(map(int, input().split()))

max_len = 0
current_len = 0

for num in arr:
    if not is_prime(num):
        current_len += 1
        max_len = max(max_len, current_len)
    else:
        current_len = 0

print(max_len)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm dãy con không chứa số nguyên tố dài nhất mà có tổng các phần tử chia hết cho k

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra độ dài của dãy con liên tiếp không chứa số nguyên tố và có tổng chia hết cho k.

Ví dụ:

Input	Output
10 3	3
4 6 8 9 10 3 4 6 8 12	

Bài 4.2: Tìm dãy con không chứa số nguyên tố dài nhất mà không chứa số chẵn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra độ dài của dãy con liên tiếp không chứa số nguyên tố và không chứa số chẵn

Ví dụ:

Input	Output
10 1 3 5 7 9 11 13 15 17 19	2

Bài 4.3: Tìm dãy con không chứa số nguyên tố dài nhất mà không chứa số lẻ

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra độ dài của dãy con liên tiếp không chứa số nguyên tố và không chứa số lẻ.

Ví dụ:

Input	Output
10 2 4 6 8 10 12 14 16 18 20	5

ĐỀ SỐ 18

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tìm số xuất hiện nhiều nhất trong danh sách	MOSTFREQNUM*	Bàn phím	Màn hình	4
Bài 2	Tìm đoạn con liên tiếp có tích lớn nhất	MAXPRODUCTSUBARR*	Bàn phím	Màn hình	5
Bài 3	Đếm số lượng chuỗi con đối xứng trong xâu	COUNTPALINDROME*	Bàn phím	Màn hình	5
Bài 4	Tìm chuỗi con có tổng số nguyên tố ASCII lớn nhất	MAXPRIMEASCII SUM*	Bàn phím	Màn hình	6

Bài 1: Tìm số xuất hiện nhiều nhất trong danh sách

Cho một danh sách các số nguyên dương. Hãy tìm số xuất hiện nhiều nhất trong danh sách. Nếu có nhiều số có cùng số lần xuất hiện, in ra số lớn nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra số xuất hiện nhiều nhất trong danh sách. Nếu có nhiều số có cùng số lần xuất hiện, in ra số lớn nhất.

Ví dụ:

Input	Output
6 2 4 4 2 4 3	4

Bài 2: Tìm đoạn con liên tiếp có tích lớn nhất

Cho một dãy số nguyên dương và âm. Hãy tìm đoạn con liên tiếp có tích lớn nhất trong dãy số.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output

- In ra tích lớn nhất của một đoạn con liên tiếp trong dãy số.

Ví dụ:

Input	Output
5 -1 2 -3 4 5	120

Bài 3: Đếm số lượng chuỗi con đối xứng trong xâu

Cho một xâu ký tự. Hãy đếm số lượng chuỗi con đối xứng trong xâu.

Input

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng chuỗi con đối xứng trong xâu.

Ví dụ:

Input	Output
ababa	9

Bài 4. Tìm chuỗi con có tổng số nguyên tố ASCII lớn nhất

Cho một xâu ký tự S . Hãy tìm chuỗi con liên tiếp có tổng giá trị ASCII của các ký tự là số nguyên tố lớn nhất.

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng giá trị ASCII lớn nhất của chuỗi con liên tiếp mà tổng là số nguyên tố.

Ví dụ:

Input	Output
abcdef	197

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tìm số xuất hiện nhiều nhất trong danh sách

1. Phân tích giải thuật:

1. Sử dụng một từ điển để đếm số lần xuất hiện của mỗi phần tử.
2. Tìm phần tử có số lần xuất hiện nhiều nhất và nếu có nhiều phần tử có cùng số lần xuất hiện, chọn phần tử lớn nhất.

2. Code tham khảo:

```
n = int(input())
arr = list(map(int, input().split()))

freq = {}
for num in arr:
    if num in freq:
        freq[num] += 1
    else:
        freq[num] = 1

max_freq = max(freq.values())
max_num = max(num for num in freq if freq[num] ==
max_freq)

print(max_num)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm số xuất hiện ít nhất trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số xuất hiện ít nhất trong danh sách. Nếu có nhiều số có cùng số lần xuất hiện, in ra số nhỏ nhất.

Ví dụ:

Input	Output
6 2 4 4 2 4 3	3

Bài 1.2: Tìm số xuất hiện nhiều nhất trong danh sách nhưng phải là số nguyên tố

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số nguyên tố xuất hiện nhiều nhất trong danh sách. Nếu có nhiều số có cùng số lần xuất hiện, in ra số lớn nhất. Nếu không có số nào là số nguyên tố, in ra -1.

Ví dụ:

Input	Output
6 2 4 4 2 5 3	5

Bài 1.3: Tìm số xuất hiện nhiều nhất trong danh sách và là số chẵn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra số chẵn xuất hiện nhiều nhất trong danh sách. Nếu có nhiều số có cùng số lần xuất hiện, in ra số lớn nhất. Nếu không có số chẵn nào, in ra -1.

Ví dụ:

Input	Output
6 2 4 4 2 4 3	4

Bài 2: Tìm đoạn con liên tiếp có tích lớn nhất

1. Phân tích giải thuật:

1. Sử dụng thuật toán Kadane phiên bản tích để tìm tích lớn nhất của một đoạn con liên tiếp.

2. Cần lưu ý cả tích lớn nhất và tích nhỏ nhất tại mỗi vị trí, do số âm có thể làm đảo dấu.

2. Code tham khảo

```
n = int(input())
arr = list(map(int, input().split()))

max_prod = arr[0]
min_prod = arr[0]
result = arr[0]

for i in range(1, n):
    if arr[i] < 0:
        max_prod, min_prod = min_prod, max_prod
    max_prod = max(arr[i], max_prod * arr[i])
    min_prod = min(arr[i], min_prod * arr[i])
    result = max(result, max_prod)

print(result)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm đoạn con liên tiếp có tích nhỏ nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tích nhỏ nhất của một đoạn con liên tiếp trong dãy số.

Ví dụ:

Input	Output
4 1 2 -3 4	-6

Bài 2.2: Tìm đoạn con liên tiếp có tích lớn nhất mà không có số âm

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).

- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tích lớn nhất của một đoạn con liên tiếp trong dãy số mà không có số âm.

Ví dụ:

Input	Output
5 -1 2 3 4 5	20

Bài 2.3: Tìm đoạn con liên tiếp có tích lớn nhất với độ dài lẻ

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên.

Output:

- In ra tích lớn nhất của một đoạn con liên tiếp trong dãy số có độ dài lẻ.

Ví dụ:

Input	Output
6 1 2 3 -4 5 6	60

Bài 3: Đếm số lượng chuỗi con đối xứng trong xâu

1. Phân tích giải thuật:

1. Sử dụng hai vòng lặp để tạo ra tất cả các chuỗi con có thể có trong xâu.
2. Kiểm tra từng chuỗi con có phải là đối xứng không và đếm số lượng chuỗi con đối xứng.

2. Code tham khảo

```
s = input()
n = len(s)
count = 0
```

```
for i in range(n):  
    for j in range(i, n):  
        sub = s[i:j+1]  
        if sub == sub[::-1]:  
            count += 1  
  
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng chuỗi con đối xứng có độ dài lẻ

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng chuỗi con đối xứng có độ dài lẻ trong xâu.

Ví dụ:

Input	Output
ababa	7

Bài 3.2: Đếm số lượng chuỗi con đối xứng có độ dài chẵn

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng chuỗi con đối xứng có độ dài chẵn trong xâu.

Ví dụ:

Input	Output
ababa	2

Bài 3.3: Tìm chuỗi con đối xứng dài nhất trong xâu

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra chuỗi con đối xứng dài nhất trong xâu.

Ví dụ:

Input	Output
ababa	ababa

Bài 4: Tìm chuỗi con có tổng số nguyên tố ASCII lớn nhất

1. Phân tích giải thuật

1. Sử dụng thuật toán Kadane để tìm tổng giá trị ASCII lớn nhất của chuỗi con liên tiếp.
2. Kiểm tra tổng đó có phải là số nguyên tố không.

2. Code tham khảo

```
def is_prime(x):  
    if x < 2:  
        return False  
    for i in range(2, int(x ** 0.5) + 1):  
        if x % i == 0:  
            return False  
    return True  
  
s = input()  
n = len(s)  
max_prime_sum = 0  
  
for i in range(n):  
    current_sum = 0  
    for j in range(i, n):  
        current_sum += ord(s[j])  
        if is_prime(current_sum):  
            max_prime_sum = max(max_prime_sum,  
current_sum)  
  
print(max_prime_sum)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm chuỗi con có tổng giá trị ASCII là số nguyên tố nhỏ nhất

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng giá trị ASCII nhỏ nhất của chuỗi con liên tiếp mà tổng là số nguyên tố.

Ví dụ:

Input	Output
abcdef	101

Bài 4.2: Tìm chuỗi con có tổng giá trị ASCII là số nguyên tố lớn nhất và độ dài lẻ

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng giá trị ASCII lớn nhất của chuỗi con liên tiếp mà tổng là số nguyên tố và có độ dài lẻ.

Ví dụ:

Input	Output
abcdefg	263

Bài 4.3: Tìm chuỗi con có tổng giá trị ASCII là số nguyên tố lớn nhất nhưng không chứa ký tự trùng lặp

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng giá trị ASCII lớn nhất của chuỗi con liên tiếp mà tổng là số nguyên tố và không chứa ký tự trùng lặp.

Ví dụ:

Input	Output
-------	--------

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

abcabcabc	197
-----------	-----

ĐỀ SỐ 19

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tìm dãy con liên tiếp có tổng lớn nhất với độ dài tối thiểu k	MAXSUMK*	Bàn phím	Màn hình	4
Bài 2	Tìm số lượng số nguyên tố trong một đoạn	COUNTPRIMES*	Bàn phím	Màn hình	5
Bài 3	Đếm số lượng từ không chứa ký tự lặp lại	COUNTONCE*	Bàn phím	Màn hình	5
Bài 4	Tìm xâu con có tổng giá trị ASCII chia hết cho k	MAXASCIIDIVK*	Bàn phím	Màn hình	6

Bài 1: Tìm dãy con liên tiếp có tổng lớn nhất với độ dài tối thiểu k

Cho một dãy số nguyên dương. Hãy tìm dãy con liên tiếp có tổng lớn nhất với độ dài tối thiểu k.

Input

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output

- In ra tổng lớn nhất của dãy con liên tiếp có độ dài ít nhất là k.

Ví dụ:

Input	Output
7 3 1 2 3 4 5 1 1	15

Bài 2: Tìm số lượng số nguyên tố trong một đoạn

Cho một đoạn [a, b]. Hãy tìm số lượng số nguyên tố trong đoạn đó.

Input

- Một dòng duy nhất chứa hai số nguyên dương a và b ($1 \leq a \leq b \leq 10^6$).

Output

- In ra số lượng số nguyên tố trong đoạn $[a, b]$.

Ví dụ:

Input	Output
10 20	4

Bài 3: Đếm số lượng từ không chứa ký tự lặp lại

Cho một chuỗi ký tự chứa các từ được phân cách bởi khoảng trắng. Hãy đếm số lượng từ không chứa ký tự lặp lại.

Input

- Một dòng duy nhất chứa chuỗi ký tự S (độ dài không quá 10^5 ký tự).

Output

- In ra số lượng từ không chứa ký tự lặp lại.

Ví dụ:

Input	Output
hello world abc	2

Bài 4. Tìm chuỗi con có tổng giá trị ASCII chia hết cho k

Cho một chuỗi ký tự S . Hãy tìm chuỗi con liên tiếp có tổng giá trị ASCII của các ký tự chia hết cho k lớn nhất.

Input:

- Dòng đầu tiên chứa số nguyên dương k ($1 \leq k \leq 100$).
- Dòng thứ hai chứa chuỗi ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng giá trị ASCII lớn nhất của chuỗi con liên tiếp mà chia hết cho k .

Ví dụ:

Input	Output
5	100
abcde	

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tìm dãy con liên tiếp có tổng lớn nhất với độ dài tối thiểu k

1. Phân tích giải thuật:

1. Sử dụng kỹ thuật Sliding Window kết hợp với mảng cộng dồn (cumulative sum) để tìm tổng lớn nhất của dãy con có độ dài tối thiểu là k.
2. Duyệt qua dãy số và tính tổng của các dãy con.

2. Code tham khảo:

```
n, k = map(int, input().split())
arr = list(map(int, input().split()))

cumulative_sum = [0] * (n + 1)
for i in range(1, n + 1):
    cumulative_sum[i] = cumulative_sum[i - 1] + arr[i - 1]

max_sum = float('-inf')
for i in range(k, n + 1):
    for j in range(i, n + 1):
        current_sum = cumulative_sum[j] - cumulative_sum[j - i]
        max_sum = max(max_sum, current_sum)

print(max_sum)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm dãy con liên tiếp có tổng nhỏ nhất với độ dài tối thiểu k

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq k \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng nhỏ nhất của dãy con liên tiếp có độ dài ít nhất là k.

Ví dụ:

Input	Output
6 2	3
1 2 3 1 2 3	

Bài 1.2: Tìm dãy con liên tiếp có tổng lớn nhất với độ dài chẵn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp có độ dài chẵn.

Ví dụ:

Input	Output
7 4 2 3 1 4 1 2 3	10

Bài 1.3: Tìm dãy con liên tiếp có tổng lớn nhất mà không có số nào lặp lại

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$).
- Dòng thứ hai chứa n số nguyên dương.

Output:

- In ra tổng lớn nhất của dãy con liên tiếp mà không có số nào lặp lại.

Ví dụ:

Input	Output
7 2 3 1 4 1 2 3	10

Bài 2: Tìm số lượng số nguyên tố trong một đoạn

1. Phân tích giải thuật:

1. Sử dụng Sàng Eratosthenes để tìm tất cả các số nguyên tố trong khoảng từ 1 đến b .
2. Đếm số lượng số nguyên tố trong đoạn $[a, b]$.

2. Code tham khảo

```
a, b = map(int, input().split())
```

```
def sieve_of_eratosthenes(max_num):
    sieve = [True] * (max_num + 1)
    sieve[0] = sieve[1] = False
    p = 2
    while p * p <= max_num:
        if sieve[p]:
            for i in range(p * p, max_num + 1, p):
                sieve[i] = False
            p += 1
    return sieve

sieve = sieve_of_eratosthenes(b)
count = sum(1 for i in range(a, b + 1) if sieve[i])

print(count)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm tổng các số nguyên tố trong đoạn [a, b]

Input:

- Một dòng duy nhất chứa hai số nguyên dương a và b ($1 \leq a \leq b \leq 10^6$).

Output:

- In ra tổng các số nguyên tố trong đoạn [a, b].

Ví dụ:

Input	Output
10 20	60

Bài 2.2: Tìm số nguyên tố lớn nhất trong đoạn [a, b]

Input:

- Một dòng duy nhất chứa hai số nguyên dương a và b ($1 \leq a \leq b \leq 10^6$).

Output:

- In ra số nguyên tố lớn nhất trong đoạn [a, b]. Nếu không có số nguyên tố nào trong đoạn, in ra -1.

Ví dụ:

Input	Output
20 30	29

Bài 2.3: Tìm số nguyên tố có tổng các chữ số là lớn nhất trong đoạn [a, b]

Input:

- Một dòng duy nhất chứa hai số nguyên dương a và b ($1 \leq a \leq b \leq 10^6$).

Output:

- In ra số nguyên tố có tổng các chữ số lớn nhất trong đoạn [a, b].

Ví dụ:

Input	Output
10 50	41

Bài 3: Đếm số lượng từ không chứa ký tự lặp lại

1. Phân tích giải thuật:

1. Sử dụng phương pháp tách từ bằng cách sử dụng hàm split().
2. Kiểm tra từng từ xem có ký tự lặp lại không và đếm số lượng từ không chứa ký tự lặp lại.

2. Code tham khảo

```
s = input().split()
count = sum(1 for word in s if len(set(word)) == len(word))
print(count)
```

3. Bài tập mở rộng:

Bài 3.1: Đếm số lượng từ chứa tất cả các ký tự là nguyên âm

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ chứa tất cả các ký tự là nguyên âm.

Ví dụ:

Input	Output
aeioo uaeio abc	1

Bài 3.2: Đếm số lượng từ chứa ít nhất một ký tự lặp lại

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra số lượng từ chứa ít nhất một ký tự lặp lại.

Ví dụ:

Input	Output
hello world abc	1

Bài 3.3: Tìm từ dài nhất mà không chứa ký tự lặp lại

Input:

- Một dòng duy nhất chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất mà không chứa ký tự lặp lại. Nếu có nhiều từ có cùng độ dài, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
hello abcdefghi	abcdefghi

Bài 4: Tìm xâu con có tổng giá trị ASCII chia hết cho k

1. Phân tích giải thuật

1. Duyệt qua tất cả các chuỗi con của xâu và tính tổng giá trị ASCII của chúng.
2. Kiểm tra tổng giá trị đó có chia hết cho k không và giữ lại tổng lớn nhất.

2. Code tham khảo

```
k = int(input())  
s = input()
```

```
n = len(s)
max_sum = 0

for i in range(n):
    current_sum = 0
    for j in range(i, n):
        current_sum += ord(s[j])
        if current_sum % k == 0:
            max_sum = max(max_sum, current_sum)

print(max_sum)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm xâu con có tổng giá trị ASCII chia hết cho k nhỏ nhất

Input:

- Dòng đầu tiên chứa số nguyên dương k ($1 \leq k \leq 100$).
- Dòng thứ hai chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng giá trị ASCII nhỏ nhất của chuỗi con liên tiếp mà chia hết cho k .

Ví dụ:

Input	Output
5 abcdef	100

Bài 4.2: Tìm xâu con có tổng giá trị ASCII chia hết cho k và độ dài lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương k ($1 \leq k \leq 100$).
- Dòng thứ hai chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng giá trị ASCII lớn nhất của chuỗi con liên tiếp mà chia hết cho k và có độ dài lớn nhất.

Ví dụ:

Input	Output
5 abcdefg	100

Bài 4.3: Tìm xâu con có tổng giá trị ASCII chia hết cho k nhưng không có ký tự nào trùng lặp

Input:

- Dòng đầu tiên chứa số nguyên dương k ($1 \leq k \leq 100$).
- Dòng thứ hai chứa xâu ký tự S (độ dài không quá 10^5 ký tự).

Output:

- In ra tổng giá trị ASCII lớn nhất của chuỗi con liên tiếp mà chia hết cho k và không có ký tự nào trùng lặp

Ví dụ:

Input	Output
5 abcabcabc	95

ĐỀ SỐ 20

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Đếm số lần xuất hiện của từ trong văn bản	WORDCOUNT*	Bàn phím	Màn hình	4
Bài 2	Tìm dãy ngày có tổng doanh thu cao nhất	MAXREVENUE*	Bàn phím	Màn hình	5
Bài 3	Phân loại học sinh theo điểm số	GRADECLASSIFICATION*	Bàn phím	Màn hình	5
Bài 4	Sắp xếp nhân viên theo tuổi và lương	SORTEMPLOYEES*	Bàn phím	Màn hình	6

Bài 1: Đếm số lần xuất hiện của từ trong văn bản

Cho một đoạn văn bản dài chứa nhiều câu và từ. Hãy đếm số lần xuất hiện của một từ cho trước trong văn bản.

Input

- Dòng đầu tiên chứa một xâu ký tự T (độ dài không quá 10^5 ký tự).
- Dòng thứ hai chứa một từ w cần đếm số lần xuất hiện.

Output

- In ra số lần xuất hiện của từ w trong văn bản T.

Ví dụ:

Input	Output
"hello world, hello everyone. hello!" "hello"	3

Bài 2: Tìm dãy ngày có tổng doanh thu cao nhất

Một cửa hàng ghi lại doanh thu hàng ngày trong một khoảng thời gian. Hãy tìm dãy ngày liên tiếp có tổng doanh thu cao nhất.

Input

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số ngày.

- Dòng thứ hai chứa n số nguyên là doanh thu của từng ngày.

Output

- In ra tổng doanh thu cao nhất của dãy ngày liên tiếp.

Ví dụ:

Input	Output
7 3 8 -2 5 -1 7 3	18

Bài 3: Phân loại học sinh theo điểm số

Trường học cần phân loại học sinh thành các nhóm A, B, C, D, F dựa trên điểm số của họ. Bạn hãy giúp trường phân loại học sinh dựa trên các tiêu chí sau:

- A: 85-100 điểm
- B: 70-84 điểm
- C: 55-69 điểm
- D: 40-54 điểm
- F: Dưới 40 điểm

Input

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số học sinh.
- Dòng thứ hai chứa n số nguyên là điểm số của từng học sinh.

Output

- In ra số lượng học sinh thuộc từng nhóm A, B, C, D, F theo thứ tự.

Ví dụ:

Input	Output
5 90 85 75 60 35	2 1 1 0 1

Bài 4. Sắp xếp nhân viên theo tuổi và lương

Một công ty cần sắp xếp danh sách nhân viên theo hai tiêu chí: tuổi và lương. Hãy sắp xếp danh sách theo tuổi tăng dần, nếu hai nhân viên có cùng tuổi thì sắp xếp theo lương giảm dần.

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số nhân viên.
- Dòng tiếp theo chứa n bộ ba giá trị: tên, tuổi và lương của từng nhân viên. Tên là một chuỗi ký tự không quá 100 ký tự, tuổi và lương là các số nguyên dương.

Output:

- In ra danh sách nhân viên sau khi đã sắp xếp theo yêu cầu.

Ví dụ:

Input	Output
3	John 30 4000
John 30 4000	Mike 30 3500
Mike 30 3500	Alice 25 5000
Alice 25 5000	

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Đếm số lần xuất hiện của từ trong văn bản

1. Phân tích giải thuật:

1. Sử dụng hàm tách từ `split()` để tách văn bản thành các từ riêng biệt.
2. Duyệt qua danh sách từ để đếm số lần xuất hiện của từ cần tìm.

2. Code tham khảo:

```
T = input()
w = input()
words = T.split()
count = words.count(w)
print(count)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm từ xuất hiện nhiều nhất trong văn bản

Input:

- Một dòng chứa văn bản T (độ dài không quá 10^5 ký tự).

Output:

- In ra từ xuất hiện nhiều nhất và số lần xuất hiện của nó.

Ví dụ:

Input	Output
"hello world, hello everyone. hello!"	"hello 3"

Bài 1.2: Đếm số lần xuất hiện của các từ trong một văn bản và sắp xếp theo thứ tự giảm dần

Input:

- Một dòng chứa văn bản T (độ dài không quá 10^5 ký tự).

Output:

- In ra các từ và số lần xuất hiện của chúng, sắp xếp theo thứ tự giảm dần.

Ví dụ:

Input	Output
"hello world, hello everyone. hello!"	"hello 3, world 1, everyone 1"

Bài 1.3: Tìm từ dài nhất trong văn bản và số lần xuất hiện của nó

Input:

- Một dòng chứa văn bản T (độ dài không quá 10^5 ký tự).

Output:

- In ra từ dài nhất và số lần xuất hiện của nó. Nếu có nhiều từ có cùng độ dài, in ra từ xuất hiện đầu tiên.

Ví dụ:

Input	Output
"hello world, hello everyone. hello!"	"everyone 1"

Bài 2: Tìm dãy ngày có tổng doanh thu cao nhất

1. Phân tích giải thuật:

1. Sử dụng kỹ thuật Sliding Window để tính tổng doanh thu của dãy ngày liên tiếp.
2. Duyệt qua danh sách doanh thu và cập nhật tổng doanh thu cao nhất.

2. Code tham khảo

```
n = int(input())
revenues = list(map(int, input().split()))

max_revenue = float('-inf')
current_revenue = 0

for rev in revenues:
    current_revenue = max(rev, current_revenue + rev)
    max_revenue = max(max_revenue, current_revenue)

print(max_revenue)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm dãy ngày có tổng doanh thu thấp nhất

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số ngày.
- Dòng thứ hai chứa n số nguyên là doanh thu của từng ngày.

Output:

- In ra tổng doanh thu thấp nhất của dãy ngày liên tiếp.

Ví dụ:

Input	Output
7 3 8 -2 5 -1 7 -10	-3

Bài 2.2: Tìm dãy ngày có tổng doanh thu lớn nhất mà không có ngày âm

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số ngày.
- Dòng thứ hai chứa n số nguyên là doanh thu của từng ngày.

Output:

- In ra tổng doanh thu lớn nhất của dãy ngày liên tiếp mà không có ngày âm.

Ví dụ:

Input	Output
7 3 8 -2 5 -1 7 3	13

Bài 2.3: Tìm dãy ngày có tổng doanh thu lớn nhất trong tháng, biết rằng tháng đó có 30 ngày

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số ngày.
- Dòng thứ hai chứa n số nguyên là doanh thu của từng ngày.

Output:

- In ra tổng doanh thu lớn nhất của dãy 30 ngày liên tiếp.

Ví dụ:

Input	Output
31 10 20 30 40 50 60 70 80 90 100 -10 -20 -30 -40 -50 -60 -70 -80 -90 -100 5 10 15 20 25 30 35 40 45 50	200

Bài 3: Phân loại học sinh theo điểm số

1. Phân tích giải thuật:

1. Sử dụng các cấu trúc điều kiện if-elif-else để phân loại từng điểm số vào nhóm tương ứng.
2. Đếm số lượng học sinh trong từng nhóm.

2. Code tham khảo

```
n = int(input())
scores = list(map(int, input().split()))

A = B = C = D = F = 0

for score in scores:
    if score >= 85:
        A += 1
    elif score >= 70:
        B += 1
    elif score >= 55:
        C += 1
    elif score >= 40:
        D += 1
    else:
        F += 1

print(A, B, C, D, F)
```

3. Bài tập mở rộng:

Bài 3.1: Tìm nhóm có số lượng học sinh ít nhất

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số học sinh.
- Dòng thứ hai chứa n số nguyên là điểm số của từng học sinh.

Output:

- In ra nhóm có số lượng học sinh ít nhất và số lượng của nhóm đó.

Ví dụ:

Input	Output
5 90 85 75 60 35	F 1

Bài 3.2: Phân loại học sinh và in ra nhóm có điểm trung bình cao nhất

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số học sinh.
- Dòng thứ hai chứa n số nguyên là điểm số của từng học sinh.

Output:

- In ra nhóm có điểm trung bình cao nhất và điểm trung bình của nhóm đó.

Ví dụ:

Input	Output
5 90 85 75 60 35	A 87.5

Bài 3.3: Tìm nhóm có số lượng học sinh lớn nhất và in ra số lượng học sinh trong nhóm đó

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số học sinh.
- Dòng thứ hai chứa n số nguyên là điểm số của từng học sinh.

Output:

- In ra nhóm có số lượng học sinh lớn nhất và số lượng của nhóm đó.

Ví dụ:

Input	Output
6 90 85 70 65 50 35	A 2

Bài 4: Sắp xếp nhân viên theo tuổi và lương

1. Phân tích giải thuật

Sử dụng hàm `sorted()` hoặc `sort()` để sắp xếp danh sách nhân viên theo tuổi tăng dần, sau đó sắp xếp theo lương giảm dần nếu tuổi bằng nhau.

2. Code tham khảo

```
n = int(input())
employees = [input().split() for _ in range(n)]

# Chuyển đổi tuổi và lương thành số nguyên để sắp xếp
for emp in employees:
    emp[1] = int(emp[1])
    emp[2] = int(emp[2])

# Sắp xếp theo tuổi tăng dần, nếu tuổi bằng nhau thì
# sắp xếp theo lương giảm dần
employees.sort(key=lambda x: (x[1], -x[2]))

# In ra danh sách sau khi sắp xếp
for emp in employees:
    print(*emp)
```

3. Bài tập mở rộng:

Bài 4.1: Sắp xếp nhân viên theo tên trong trường hợp cùng tuổi và cùng lương

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số nhân viên.
- Dòng tiếp theo chứa n bộ ba giá trị: tên, tuổi và lương của từng nhân viên. Tên là một chuỗi ký tự không quá 100 ký tự, tuổi và lương là các số nguyên dương.

Output:

- In ra danh sách nhân viên sau khi sắp xếp theo tuổi, lương và tên (nếu cùng tuổi và lương).

Ví dụ:

Input	Output
3 John 30 5000 Alice 30 5000 Mike 30 4000	Alice 30 5000 John 30 4000 Mike 30 4000

Bài 4.2: Tìm nhân viên có lương cao nhất trong nhóm tuổi cụ thể

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số nhân viên.
- Dòng tiếp theo chứa n bộ ba giá trị: tên, tuổi và lương của từng nhân viên.
- Dòng cuối cùng chứa một số nguyên là tuổi cụ thể.

Output:

- In ra nhân viên có lương cao nhất trong nhóm tuổi cụ thể đó.

Ví dụ:

Input	Output
3 John 30 4000 Mike 30 3500 Alice 25 5000 30	Alice 5000

Bài 4.3: Tìm nhóm tuổi có tổng lương cao nhất và in ra tổng lương của nhóm đó

Input:

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^5$) là số nhân viên.
- Dòng tiếp theo chứa n bộ ba giá trị: tên, tuổi và lương của từng nhân viên.

Output:

- In ra nhóm tuổi có tổng lương cao nhất và tổng lương của nhóm đó.

Ví dụ:

Input	Output
5 John 30 4000 Mike 30 3500 Alice 25 5000 Bob 30 3000 Tom 40 2000	30 10500

ĐỀ SỐ 21

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Phân tích dữ liệu bán hàng theo tuần	WEEKLYSALES*	Bàn phím	Màn hình	4
Bài 2	Quản lý lượng khách hàng vào cửa hàng	CUSTOMERFLOW*	Bàn phím	Màn hình	5
Bài 3	Tìm sản phẩm bán chạy nhất trong tháng	TOPPRODUCT*	Bàn phím	Màn hình	5
Bài 4	Tối ưu hóa quy trình sản xuất	PRODUCTIONOPTIMIZATION*	Bàn phím	Màn hình	6

Bài 1: Phân tích dữ liệu bán hàng theo tuần

Một công ty cần phân tích dữ liệu bán hàng theo tuần để hiểu rõ hơn về xu hướng mua sắm của khách hàng. Bạn được yêu cầu tính tổng doanh thu mỗi tuần và tìm tuần có doanh thu cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày.
- Dòng tiếp theo chứa n số nguyên là doanh thu của từng ngày.

Output

- In ra tuần có tổng doanh thu cao nhất (giả sử tuần bắt đầu từ thứ Hai và có 7 ngày) và tổng doanh thu của tuần đó. Nếu có nhiều tuần có cùng doanh thu, in ra tuần xuất hiện đầu tiên.

Ví dụ:

Input	Output
14 300 400 500 600 700 800 900 300 400 500 600 700 800 900	Week 1 2400

Bài 2: Quản lý lượng khách hàng vào cửa hàng

Một cửa hàng cần quản lý lượng khách hàng vào cửa hàng mỗi giờ trong

ngày để tối ưu hóa hoạt động. Hãy giúp cửa hàng xác định khoảng thời gian có lượng khách hàng cao nhất và thấp nhất trong ngày.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số giờ trong ngày.
- Dòng thứ hai chứa n số nguyên là số lượng khách hàng vào cửa hàng trong từng giờ.

Output

- In ra khoảng thời gian (bắt đầu từ giờ) có lượng khách hàng cao nhất và thấp nhất, cùng với số lượng khách hàng tương ứng.

Ví dụ:

Input	Output
6	1-3 150
50 50 50 30 20 10	4-6 60

Bài 3: Tìm sản phẩm bán chạy nhất trong tháng

Cửa hàng cần xác định sản phẩm bán chạy nhất trong tháng và so sánh với các sản phẩm khác để lập kế hoạch nhập hàng. Bạn được yêu cầu tìm sản phẩm bán chạy nhất, sản phẩm bán ít nhất và tổng số sản phẩm bán được trong tháng.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số loại sản phẩm.
- Dòng tiếp theo chứa n số nguyên là số lượng bán được của từng sản phẩm.

Output

- In ra số lượng sản phẩm bán chạy nhất, bán ít nhất và tổng số sản phẩm bán được trong tháng.

Ví dụ:

Input	Output
5	500 100 1500
100 200 300 400 500	

Bài 4. Tối ưu hóa quy trình sản xuất

Một nhà máy cần tối ưu hóa quy trình sản xuất để đạt hiệu quả cao nhất. Hãy tìm quy trình sản xuất tối ưu bằng cách tối đa hóa sản lượng mà không vượt quá giới hạn nguyên vật liệu cho trước.

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và m ($1 \leq n, m \leq 10^5$) là số loại sản phẩm và giới hạn nguyên vật liệu.
- Dòng tiếp theo chứa n số nguyên là lượng nguyên vật liệu cần cho mỗi sản phẩm.
- Dòng cuối cùng chứa n số nguyên là sản lượng mỗi loại sản phẩm có thể đạt được.

Output:

- In ra sản lượng tối ưu nhất có thể đạt được mà không vượt quá giới hạn nguyên vật liệu m .

Ví dụ:

Input	Output
3 100	90
30 40 50	60 80 100

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Phân tích dữ liệu bán hàng theo tuần

1. Phân tích giải thuật:

1. Sử dụng một vòng lặp để tính tổng doanh thu cho từng tuần.
2. So sánh và tìm tuần có tổng doanh thu cao nhất.

2. Code tham khảo:

```
n = int(input())
sales = list(map(int, input().split()))

max_week_sum = 0
best_week = 0

for i in range(0, n, 7):
    week_sum = sum(sales[i:i+7])
    if week_sum > max_week_sum:
        max_week_sum = week_sum
        best_week = i // 7 + 1

print(f"Week {best_week} {max_week_sum}")
```

3. Bài tập mở rộng:

Bài 1.1: Tìm tuần có doanh thu thấp nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày.
- Dòng tiếp theo chứa n số nguyên là doanh thu của từng ngày.

Output:

- In ra tuần có doanh thu thấp nhất và tổng doanh thu của tuần đó

Ví dụ:

Input	Output
14 900 800 700 600 500 400 300 900 800 700 600 500 400 300	Week 2 2400

Bài 1.2: Tìm tổng doanh thu của nửa tháng đầu tiên và nửa tháng cuối cùng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày.

- Dòng tiếp theo chứa n số nguyên là doanh thu của từng ngày.

Output:

- In ra tổng doanh thu của nửa tháng đầu tiên và nửa tháng cuối cùng.

Ví dụ:

Input	Output
30 300 400 500 600 700 800 900 300 400 500 600 700 800 900 300 400 500 600 700 800 900 300 400 500 600 700 800 900	7500 7500

Bài 1.3: Phân tích doanh thu trung bình mỗi tuần và tìm tuần có mức tăng trưởng cao nhất so với tuần trước

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày.
- Dòng tiếp theo chứa n số nguyên là doanh thu của từng ngày.

Output:

- In ra tuần có mức tăng trưởng cao nhất so với tuần trước đó và tỷ lệ phần trăm tăng trưởng.

Ví dụ:

Input	Output
14 300 400 500 600 700 800 900 400 500 600 700 800 900 1000	Week 2 33.33%

Bài 2: Quản lý lượng khách hàng vào cửa hàng

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng lượng khách hàng trong từng khoảng thời gian (có thể là từng 3 giờ).
2. So sánh và tìm khoảng thời gian có lượng khách hàng cao nhất và thấp nhất.

2. Code tham khảo

```
n = int(input())
customers = list(map(int, input().split()))

max_customers = 0
min_customers = float('inf')
best_period = ""
worst_period = ""

for i in range(0, n, 3):
    period_sum = sum(customers[i:i+3])
    if period_sum > max_customers:
        max_customers = period_sum
        best_period = f"{i+1}-{i+3}"
    if period_sum < min_customers:
        min_customers = period_sum
        worst_period = f"{i+1}-{i+3}"

print(f"{best_period} {max_customers}")
print(f"{worst_period} {min_customers}")
```

3. Bài tập mở rộng:

Bài 2.1: Tìm khoảng thời gian có lượng khách hàng ổn định nhất (không chênh lệch nhiều giữa các giờ)

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số giờ trong ngày.
- Dòng thứ hai chứa n số nguyên là số lượng khách hàng vào cửa hàng trong từng giờ.

Output:

- In ra khoảng thời gian có lượng khách hàng ổn định nhất và mức độ ổn định đó (sự chênh lệch nhỏ nhất).

Ví dụ:

Input	Output
-------	--------

6	1-3 0
50 50 50 60 70 80	

Bài 2.2: Phân tích lượng khách hàng trong ngày và tìm giờ cao điểm (giờ có lượng khách hàng cao nhất)

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số giờ trong ngày.
- Dòng thứ hai chứa n số nguyên là số lượng khách hàng vào cửa hàng trong từng giờ.

Output:

- In ra giờ cao điểm và lượng khách hàng trong giờ đó.

Ví dụ:

Input	Output
6 20 30 60 40 50 10	3 60

Bài 2.3: Tìm khoảng thời gian liên tiếp dài nhất có lượng khách hàng dưới một ngưỡng cho trước

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và k ($1 \leq n \leq 24$, $1 \leq k \leq 10^5$) là số giờ trong ngày và ngưỡng khách hàng.
- Dòng thứ hai chứa n số nguyên là số lượng khách hàng vào cửa hàng trong từng giờ.

Output:

- In ra khoảng thời gian liên tiếp dài nhất có lượng khách hàng dưới ngưỡng k và tổng số lượng khách hàng trong khoảng thời gian đó.

Ví dụ:

Input	Output
6 50 20 30 40 10 20 60	4-6 90

Bài 3: Tìm sản phẩm bán chạy nhất trong tháng

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tìm số lượng sản phẩm bán chạy nhất và bán ít nhất.
2. Tính tổng số sản phẩm bán được.

2. Code tham khảo

```
n = int(input())
sales = list(map(int, input().split()))

max_sales = max(sales)
min_sales = min(sales)
total_sales = sum(sales)

print(max_sales, min_sales, total_sales)
```

3. Bài tập mở rộng:

Bài 3.1: Tìm sản phẩm bán chạy nhất trong tuần (giả sử tuần có 7 ngày)

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số loại sản phẩm.
- Dòng tiếp theo chứa n số nguyên là số lượng bán được của từng sản phẩm.

Output:

- In ra số lượng sản phẩm bán chạy nhất trong tuần.

Ví dụ:

Input	Output
7 100 200 300 400 500 600 700	500

Bài 3.2: Tìm sản phẩm bán chạy nhất trong tháng với điều kiện có ít nhất k

sản phẩm được bán

Input:

- Dòng đầu tiên chứa số nguyên dương n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^5$) là số loại sản phẩm và số lượng tối thiểu.
- Dòng tiếp theo chứa n số nguyên là số lượng bán được của từng sản phẩm.

Output:

- In ra số lượng sản phẩm bán chạy nhất với điều kiện có ít nhất k sản phẩm được bán.

Ví dụ:

Input	Output
5 300 100 200 300 400 500	500

Bài 3.3: So sánh doanh thu của sản phẩm bán chạy nhất với sản phẩm bán ít nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số loại sản phẩm.
- Dòng tiếp theo chứa n số nguyên là số lượng bán được của từng sản phẩm.

Output:

- In ra sự chênh lệch giữa sản phẩm bán chạy nhất và bán ít nhất.

Ví dụ:

Input	Output
5 100 200 300 400 500	400

Bài 4: Tối ưu hóa quy trình sản xuất

1. Phân tích giải thuật

1. Sử dụng phương pháp tham lam để chọn các sản phẩm có tỷ lệ sản lượng trên nguyên vật liệu cao nhất trước.

2. Tính tổng sản lượng tối ưu mà không vượt quá giới hạn nguyên vật liệu.

2. Code tham khảo

```
n, m = map(int, input().split())
materials = list(map(int, input().split()))
outputs = list(map(int, input().split()))

ratios = [(outputs[i] / materials[i], materials[i],
outputs[i]) for i in range(n)]
ratios.sort(reverse=True)

total_output = 0
total_materials = 0

for ratio, material, output in ratios:
    if total_materials + material <= m:
        total_output += output
        total_materials += material

print(total_output)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm sản lượng tối đa có thể đạt được với một loại sản phẩm nhất định trong trường hợp giới hạn nguyên vật liệu thay đổi

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và m ($1 \leq n, m \leq 10^5$) là số loại sản phẩm và giới hạn nguyên vật liệu.
- Dòng tiếp theo chứa n số nguyên là lượng nguyên vật liệu cần cho mỗi sản phẩm.
- Dòng cuối cùng chứa n số nguyên là sản lượng mỗi loại sản phẩm có thể đạt được.

Output:

- In ra sản lượng tối đa có thể đạt được của một sản phẩm cụ thể khi giới hạn nguyên vật liệu thay đổi.

Ví dụ:

Input	Output
3 100	90
30 40 50	60 80 100

Bài 4.2: Tìm phương án sản xuất tối ưu khi thêm một giới hạn về số lượng sản phẩm có thể sản xuất

Input:

- Dòng đầu tiên chứa ba số nguyên dương n , m , và k ($1 \leq n, m \leq 10^5$, $1 \leq k \leq n$) là số loại sản phẩm, giới hạn nguyên vật liệu và số lượng tối đa sản phẩm có thể sản xuất.

- Dòng tiếp theo chứa n số nguyên là lượng nguyên vật liệu cần cho mỗi sản phẩm.

- Dòng cuối cùng chứa n số nguyên là sản lượng mỗi loại sản phẩm có thể đạt được.

Output:

- In ra sản lượng tối ưu nhất có thể đạt được khi chỉ có thể sản xuất tối đa k sản phẩm.

Ví dụ:

Input	Output
3 100 2	140
30 40 50	60 80 100

Bài 4.3: So sánh sản lượng tối ưu trong hai tình huống giới hạn nguyên vật liệu khác nhau

Input:

- Dòng đầu tiên chứa ba số nguyên dương n , m_1 , m_2 ($1 \leq n \leq 10^5$, $1 \leq m_1, m_2 \leq 10^5$) là số loại sản phẩm và hai giới hạn nguyên vật liệu khác nhau.

- Dòng tiếp theo chứa n số nguyên là lượng nguyên vật liệu cần cho mỗi sản phẩm.

- Dòng cuối cùng chứa n số nguyên là sản lượng mỗi loại sản phẩm có thể đạt được.

Output:

● In ra sản lượng tối ưu nhất trong từng trường hợp giới hạn nguyên vật liệu m_1 và m_2 .

Ví dụ:

Input	Output
3 100 150	90 170
30 40 50	60 80 100

ĐỀ SỐ 22

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Phân tích hành vi người dùng trên website	USERBEHAVIOR*	Bàn phím	Màn hình	4
Bài 2	Tính toán tối ưu chi phí sản xuất	COSTOPTIMIZATION*	Bàn phím	Màn hình	5
Bài 3	Dự đoán lượng mưa theo chuỗi thời gian	RAINPREDICTION*	Bàn phím	Màn hình	5
Bài 4	Xây dựng lịch làm việc tối ưu	WORKSCHEDULE*	Bàn phím	Màn hình	6

Bài 1: Phân tích hành vi người dùng trên website

Một website cần phân tích hành vi của người dùng thông qua thời gian họ truy cập vào các trang khác nhau. Bạn được yêu cầu tính tổng thời gian truy cập của người dùng trong mỗi phiên và xác định phiên có thời gian truy cập dài nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số phiên truy cập.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của một phiên truy cập (tính bằng phút).

Output

- In ra thời gian truy cập dài nhất và tổng thời gian của tất cả các phiên truy cập.

Ví dụ:

Input	Output
3 30 60 45 90	60 150

15 45	
-------	--

Bài 2: Tính toán tối ưu chi phí sản xuất

Một công ty cần tính toán chi phí sản xuất của các sản phẩm. Bạn được yêu cầu tính tổng chi phí và tìm sản phẩm có chi phí sản xuất cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số sản phẩm.
- Mỗi dòng tiếp theo chứa hai số nguyên là số lượng sản phẩm và chi phí sản xuất mỗi đơn vị sản phẩm.

Output

- In ra chi phí sản xuất cao nhất và tổng chi phí sản xuất của tất cả các sản phẩm.

Ví dụ:

Input	Output
3 100 50 200 20 150 30	5000 12000

Bài 3: Dự đoán lượng mưa theo chuỗi thời gian

Một hệ thống dự báo thời tiết cần dự đoán lượng mưa trong một chuỗi thời gian. Bạn được yêu cầu tìm khoảng thời gian có lượng mưa cao nhất và dự đoán lượng mưa cho các khoảng thời gian tương lai dựa trên dữ liệu hiện tại.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng mưa trong ngày đó (tính bằng mm).

Output

- In ra khoảng thời gian có lượng mưa cao nhất và tổng lượng mưa trong khoảng thời gian đó.

Ví dụ:

Input	Output
7 20 30 50 10 20 30 40	3-5 120

Bài 4. Xây dựng lịch làm việc tối ưu

Một công ty cần xây dựng lịch làm việc cho nhân viên sao cho tối ưu hóa thời gian làm việc và nghỉ ngơi. Bạn được yêu cầu phân chia lịch làm việc của nhân viên theo cách tối ưu nhất dựa trên các yêu cầu về thời gian làm việc và nghỉ ngơi.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày làm việc.
- Mỗi dòng tiếp theo chứa hai số nguyên là số giờ làm việc tối thiểu và tối đa của mỗi ngày.

Output:

- In ra lịch làm việc tối ưu nhất sao cho tổng số giờ làm việc gần với một giá trị cho trước nhất.

Ví dụ:

Input	Output
3 8 10 7 9 6 8	25

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Phân tích hành vi người dùng trên website

1. Phân tích giải thuật:

1. Dùng vòng lặp để tính thời gian truy cập của từng phiên (thời gian kết thúc trừ thời gian bắt đầu).
2. Tìm thời gian truy cập dài nhất và tính tổng thời gian của tất cả các phiên.

2. Code tham khảo:

```
n = int(input())
max_time = 0
total_time = 0

for _ in range(n):
    start, end = map(int, input().split())
    session_time = end - start
    total_time += session_time
    if session_time > max_time:
        max_time = session_time

print(max_time, total_time)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm phiên truy cập ngắn nhất và dài nhất, sau đó tính khoảng cách thời gian giữa chúng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số phiên truy cập.

Output:

- In ra khoảng cách thời gian giữa phiên ngắn nhất và dài nhất.

Ví dụ:

Input	Output
3	45
30 60	
45 90	
15 45	

Bài 1.2: Xác định phiên truy cập bắt đầu sớm nhất và kết thúc muộn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số phiên truy cập.

Output:

- In ra thời gian bắt đầu của phiên sớm nhất và thời gian kết thúc của phiên muộn nhất.

Ví dụ:

Input	Output
3 30 60 45 90 15 45	15 90

Bài 1.3: Tính tổng thời gian truy cập của những phiên bắt đầu trong khoảng thời gian nhất định

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số phiên truy cập.

- Dòng thứ hai chứa hai số nguyên dương l và r là khoảng thời gian giới hạn (tính bằng phút).

Output:

- In ra tổng thời gian truy cập của các phiên bắt đầu trong khoảng thời gian từ l đến r .

Ví dụ:

Input	Output
3 15 45 30 60 45 90 15 45	90

Bài 2: Tính toán tối ưu chi phí sản xuất

1. Phân tích giải thuật:

1. Dùng vòng lặp để tính chi phí sản xuất cho mỗi sản phẩm (số lượng nhân với chi phí đơn vị).
2. Tìm chi phí sản xuất cao nhất và tính tổng chi phí của tất cả các sản phẩm.

2. Code tham khảo

```
n = int(input())
max_cost = 0
total_cost = 0

for _ in range(n):
    quantity, cost_per_unit = map(int,
input().split())
    product_cost = quantity * cost_per_unit
    total_cost += product_cost
    if product_cost > max_cost:
        max_cost = product_cost

print(max_cost, total_cost)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm sản phẩm có chi phí sản xuất thấp nhất và cao nhất, sau đó tính chênh lệch chi phí

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số sản phẩm.
- Mỗi dòng tiếp theo chứa hai số nguyên là số lượng sản phẩm và chi phí sản xuất mỗi đơn vị sản phẩm.

Output:

- In ra chênh lệch chi phí sản xuất giữa sản phẩm cao nhất và thấp nhất.

Ví dụ:

Input	Output
3	2000
100 50	
200 20	
150 30	

Bài 2.2: Tính chi phí sản xuất trung bình cho tất cả các sản phẩm

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số sản phẩm.
- Mỗi dòng tiếp theo chứa hai số nguyên là số lượng sản phẩm và chi phí sản xuất mỗi đơn vị sản phẩm.

Output:

- In ra chi phí sản xuất trung bình cho tất cả các sản phẩm.

Ví dụ:

Input	Output
3	4000
100 50	
200 20	
150 30	

Bài 2.3: Tính tổng chi phí sản xuất khi thêm một khoản chi phí cố định cho mỗi sản phẩm

Input:

- Dòng đầu tiên chứa số nguyên dương n và f ($1 \leq n \leq 10^5$, $1 \leq f \leq 10^9$) là số sản phẩm và chi phí cố định cho mỗi sản phẩm.
- Mỗi dòng tiếp theo chứa hai số nguyên là số lượng sản phẩm và chi phí sản xuất mỗi đơn vị sản phẩm.

Output:

• In ra tổng chi phí sản xuất khi thêm khoản chi phí cố định f vào mỗi sản phẩm.

Ví dụ:

Input	Output
3 10	12300
100 50	
200 20	
150 30	

Bài 3: Dự đoán lượng mưa theo chuỗi thời gian

1. Phân tích giải thuật:

1. Sử dụng phương pháp Sliding Window để tìm khoảng thời gian có lượng mưa cao nhất.

2. Tính tổng lượng mưa trong khoảng thời gian đó.

2. Code tham khảo

```
n = int(input())
rainfall = list(map(int, input().split()))

max_rain = 0
current_rain = 0
best_start = 0
best_end = 0

for i in range(n):
    current_rain += rainfall[i]
    if i >= 2: # Xét khoảng thời gian dài 3 ngày
        if current_rain > max_rain:
            max_rain = current_rain
            best_start = i - 2
            best_end = i
        current_rain -= rainfall[i - 2]
```

```
print(f"{best_start+1}-{best_end+1} {max_rain}")
```

3. Bài tập mở rộng:

Bài 3.1: Tìm khoảng thời gian có lượng mưa thấp nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng mưa trong ngày đó (tính bằng mm).

Output:

- In ra khoảng thời gian có lượng mưa thấp nhất và tổng lượng mưa trong khoảng thời gian đó.

Ví dụ:

Input	Output
7 20 30 50 10 20 30 40	6-7 70

Bài 3.2: Dự đoán lượng mưa trong 3 ngày tiếp theo dựa trên lượng mưa trung bình của 3 ngày gần nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng mưa trong ngày đó (tính bằng mm).

Output:

- In ra dự đoán lượng mưa cho 3 ngày tiếp theo.

Ví dụ:

Input	Output
7 20 30 50 10 20 30 40	30

Bài 3.3: So sánh lượng mưa trong hai khoảng thời gian liên tiếp và tìm khoảng

thời gian có sự thay đổi lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng mưa trong ngày đó (tính bằng mm).

Output:

- In ra khoảng thời gian có sự thay đổi lượng mưa lớn nhất và mức độ thay đổi.

Ví dụ:

Input	Output
7 20 30 50 10 20 30 40	3-4 40

Bài 4: Xây dựng lịch làm việc tối ưu

1. Phân tích giải thuật

1. Sử dụng vòng lặp để tính tổng giờ làm việc cho mỗi ngày dựa trên giá trị gần với yêu cầu.
2. Tối ưu hóa tổng giờ làm việc sao cho gần với giá trị yêu cầu nhất.

2. Code tham khảo

```
n = int(input())
total_hours = 0

for _ in range(n):
    min_hours, max_hours = map(int, input().split())
    # Giả sử cần tối ưu gần 8 giờ mỗi ngày
    if (max_hours + min_hours) // 2 <= 8:
        total_hours += max_hours
    else:
        total_hours += min_hours

print(total_hours)
```

3. Bài tập mở rộng:

Bài 4.1: Tìm lịch làm việc tối ưu sao cho tổng số giờ làm việc không vượt quá một giới hạn cho trước

Input:

- Dòng đầu tiên chứa số nguyên dương n và h ($1 \leq n \leq 10^5$, $1 \leq h \leq 10^9$) là số ngày làm việc và giới hạn giờ làm việc.
- Mỗi dòng tiếp theo chứa hai số nguyên là số giờ làm việc tối thiểu và tối đa của mỗi ngày.

Output:

- In ra lịch làm việc tối ưu sao cho tổng số giờ làm việc không vượt quá h .

Ví dụ:

Input	Output
3 24 8 10 7 9 6 8	24

Bài 4.2: Tìm lịch làm việc tối ưu khi nhân viên chỉ có thể làm việc vào một số ngày nhất định trong tuần

Input:

- Dòng đầu tiên chứa số nguyên dương n và m ($1 \leq n \leq 10^5$, $1 \leq m \leq 7$) là số ngày làm việc và số ngày được chọn để làm việc.
- Mỗi dòng tiếp theo chứa hai số nguyên là số giờ làm việc tối thiểu và tối đa của mỗi ngày.

Output:

- In ra lịch làm việc tối ưu cho những ngày làm việc được chọn.

Ví dụ:

Input	Output
7 3 8 10 7 9	24

6 8	
7 8	
6 7	
5 6	
8 10	

Bài 4.3: Tối ưu hóa lịch làm việc sao cho khoảng cách giữa ngày làm việc và ngày nghỉ là nhỏ nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số ngày làm việc.
- Mỗi dòng tiếp theo chứa hai số nguyên là số giờ làm việc tối thiểu và tối đa của mỗi ngày.

Output:

- In ra lịch làm việc tối ưu sao cho khoảng cách giữa ngày làm việc và ngày nghỉ là nhỏ nhất.

Ví dụ:

Input	Output
3 8 10 7 9 6 8	2 4

ĐỀ SỐ 23

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Phân tích số lượng đơn hàng theo từng giờ	ORDERANALYSIS*	Bàn phím	Màn hình	4
Bài 2	Tối ưu hóa lộ trình giao hàng	DELIVERYROUTE*	Bàn phím	Màn hình	5
Bài 3	Dự đoán lượng khách hàng trong tuần tới	CUSTOMERPREDICTION*	Bàn phím	Màn hình	5
Bài 4	Phân bổ ngân sách cho chiến dịch quảng cáo	ADVERTBUDGET*	Bàn phím	Màn hình	6

Bài 1: Phân tích số lượng đơn hàng theo từng giờ

Một cửa hàng online cần phân tích số lượng đơn hàng theo từng giờ để cải thiện dịch vụ khách hàng. Bạn được yêu cầu tính tổng số lượng đơn hàng trong mỗi giờ và tìm ra giờ có số lượng đơn hàng cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số lượng đơn hàng trong ngày.
- Mỗi dòng tiếp theo chứa một số nguyên là thời điểm (giờ) mà một đơn hàng được đặt.

Output

- In ra giờ có số lượng đơn hàng cao nhất và tổng số đơn hàng trong giờ đó. Nếu có nhiều giờ có cùng số lượng đơn hàng cao nhất, in ra giờ nhỏ nhất.

Ví dụ:

Input	Output
6 1 4	1 4

1 3	
1 4	
1 2	
1 4	
1 3	

Bài 2: Tối ưu hóa lộ trình giao hàng

Một công ty giao hàng cần tối ưu hóa lộ trình giao hàng sao cho tổng khoảng cách di chuyển là ngắn nhất. Bạn được yêu cầu tính tổng khoảng cách giao hàng và tối ưu hóa lộ trình.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số điểm giao hàng.
- Mỗi dòng tiếp theo chứa một số nguyên là khoảng cách từ điểm giao hàng trước đến điểm giao hàng tiếp theo.

Output

- In ra tổng khoảng cách di chuyển ngắn nhất có thể đạt được.

Ví dụ:

Input	Output
5 20 30 10 50 40	150

Bài 3: Dự đoán lượng khách hàng trong tuần tới

Một cửa hàng cần dự đoán lượng khách hàng trong tuần tới dựa trên dữ liệu tuần trước. Bạn được yêu cầu tính tổng lượng khách hàng dự đoán cho tuần tới dựa trên trung bình lượng khách hàng của tuần trước.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày trong tuần.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng khách hàng trong mỗi ngày.

Output

- In ra tổng lượng khách hàng dự đoán cho tuần tới dựa trên trung bình lượng khách hàng của tuần trước.

Ví dụ:

Input	Output
7	490
70	
60	
80	
50	
90	
60	
80	

Bài 4. Phân bổ ngân sách cho chiến dịch quảng cáo

Một công ty cần phân bổ ngân sách cho chiến dịch quảng cáo trên các kênh khác nhau. Bạn được yêu cầu tính toán tổng ngân sách cần thiết và phân bổ ngân sách tối ưu sao cho mỗi kênh nhận được ít nhất một khoản ngân sách nhất định.

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và m ($1 \leq n \leq 10$, $1 \leq m \leq 10^5$) là số kênh quảng cáo và tổng ngân sách.
- Mỗi dòng tiếp theo chứa hai số nguyên là mức ngân sách tối thiểu và tối đa cho mỗi kênh.

Output:

- In ra tổng ngân sách cần thiết và phân bổ tối ưu cho các kênh quảng cáo.

Ví dụ:

Input	Output
3 100	100
20 50	
30 40	
10 20	

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Phân tích số lượng đơn hàng theo từng giờ

1. Phân tích giải thuật:

1. Sử dụng một mảng để lưu trữ số lượng đơn hàng theo từng giờ.
2. Duyệt qua danh sách đơn hàng và cập nhật số lượng đơn hàng cho từng giờ.
3. Tìm giờ có số lượng đơn hàng cao nhất.

2. Code tham khảo:

```
n = int(input())
orders = [0] * 24

for _ in range(n):
    hour = int(input())
    orders[hour] += 1

max_orders = max(orders)
max_hour = orders.index(max_orders)

print(max_hour, max_orders)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm giờ có số lượng đơn hàng thấp nhất và in ra giờ đó cùng số lượng đơn hàng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số lượng đơn hàng trong ngày.
- Mỗi dòng tiếp theo chứa một số nguyên là thời điểm (giờ) mà một đơn hàng được đặt.

Output:

- In ra giờ có số lượng đơn hàng thấp nhất và tổng số đơn hàng trong giờ đó.

Ví dụ:

Input	Output
6 14 13	12 1

1 4	
1 2	
1 4	
1 3	

Bài 1.2: Tính tổng số lượng đơn hàng trong khoảng thời gian từ giờ l đến giờ r

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số lượng đơn hàng trong ngày.
- Dòng thứ hai chứa hai số nguyên l và r là khoảng thời gian (giờ) cần tính.

Output:

- In ra tổng số lượng đơn hàng trong khoảng thời gian từ giờ l đến giờ r .

Ví dụ:

Input	Output
6 1 2 1 4 1 4 1 3 1 4 1 2 1 4 1 3	5

Bài 1.3: Phân tích sự thay đổi số lượng đơn hàng giữa các giờ liên tiếp và tìm giờ có sự thay đổi lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số lượng đơn hàng trong ngày.

- Mỗi dòng tiếp theo chứa một số nguyên là thời điểm (giờ) mà một đơn hàng được đặt.

Output:

- In ra giờ có sự thay đổi lớn nhất về số lượng đơn hàng so với giờ trước đó và mức độ thay đổi.

Ví dụ:

Input	Output
6	13 1
14	
13	
14	
12	
14	
13	

Bài 2: Tối ưu hóa lộ trình giao hàng

1. Phân tích giải thuật:

1. Sử dụng phương pháp tham lam (greedy) để tối ưu hóa tổng khoảng cách di chuyển.

2. Sắp xếp các điểm giao hàng theo khoảng cách từ nhỏ đến lớn và tính tổng khoảng cách.

2. Code tham khảo

```
n = int(input())
distances = [int(input()) for _ in range(n)]

distances.sort()
min_distance = sum(distances)

print(min_distance)
```

3. Bài tập mở rộng:

Bài 2.1: Tính tổng khoảng cách giao hàng khi ưu tiên giao những điểm có

khoảng cách xa nhất trước

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số điểm giao hàng.
- Mỗi dòng tiếp theo chứa một số nguyên là khoảng cách từ điểm giao hàng trước đến điểm giao hàng tiếp theo.

Output:

- In ra tổng khoảng cách di chuyển khi ưu tiên những điểm xa nhất trước.

Ví dụ:

Input	Output
5	150
20	
30	
10	
50	
40	

Bài 2.2: Phân tích khoảng cách giao hàng và tìm điểm giao hàng tối ưu (có khoảng cách trung bình so với các điểm khác)

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số điểm giao hàng.
- Mỗi dòng tiếp theo chứa một số nguyên là khoảng cách từ điểm giao hàng trước đến điểm giao hàng tiếp theo.

Output:

- In ra điểm giao hàng có khoảng cách trung bình so với các điểm khác và khoảng cách đó.

Ví dụ:

Input	Output
5	30

20	
30	
10	
50	
40	

Bài 2.3: Tối ưu hóa lộ trình giao hàng sao cho tổng thời gian di chuyển là ngắn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số điểm giao hàng.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian di chuyển từ điểm giao hàng trước đến điểm giao hàng tiếp theo.

Output:

- In ra tổng thời gian di chuyển ngắn nhất có thể đạt được.

Ví dụ:

Input	Output
5 20 30 10 50 40	150

Bài 3: Dự đoán lượng khách hàng trong tuần tới

1. Phân tích giải thuật:

1. Tính trung bình lượng khách hàng trong tuần trước.
2. Nhân kết quả trung bình với 7 để dự đoán lượng khách hàng cho tuần tới.

2. Code tham khảo

```
n = int(input())
customers = [int(input()) for _ in range(n)]

average_customers = sum(customers) // n
predicted_customers = average_customers * 7

print(predicted_customers)
```

3. Bài tập mở rộng:

Bài 3.1: Dự đoán lượng khách hàng cho tuần tới nếu lượng khách hàng tăng đều 10% mỗi ngày

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày trong tuần.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng khách hàng trong mỗi ngày.

Output:

- In ra tổng lượng khách hàng dự đoán cho tuần tới nếu lượng khách hàng tăng đều 10% mỗi ngày.

Ví dụ:

Input	Output
7	539
70	
60	
80	
50	
90	
60	
80	

Bài 3.2: Dự đoán lượng khách hàng cho tuần tới nếu lượng khách hàng giảm 5% mỗi ngày

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày trong tuần.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng khách hàng trong mỗi ngày.

Output:

- In ra tổng lượng khách hàng dự đoán cho tuần tới nếu lượng khách hàng giảm 5% mỗi ngày.

Ví dụ:

Input	Output
7	4 6 6
7 0	
6 0	
8 0	
5 0	
9 0	
6 0	
8 0	

Bài 3.3: So sánh lượng khách hàng dự đoán giữa hai tuần và tìm tuần có lượng khách hàng tăng trưởng lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày trong tuần.
- Dòng thứ hai chứa n số nguyên là lượng khách hàng tuần trước.
- Dòng thứ ba chứa n số nguyên là lượng khách hàng tuần này.

Output:

- In ra tuần có lượng khách hàng tăng trưởng lớn nhất và mức độ tăng trưởng.

Ví dụ:

Input	Output
7 70 60 80 50 90 60 80 80 70 90 55 100 65 90	14%

Bài 4: Phân bổ ngân sách cho chiến dịch quảng cáo

1. Phân tích giải thuật

1. Sử dụng phương pháp tham lam (greedy) để phân bổ ngân sách tối thiểu cho mỗi kênh trước.

2. Phân bổ ngân sách còn lại theo cách tối ưu cho các kênh sao cho không vượt quá tổng ngân sách.

2. Code tham khảo

```
n, m = map(int, input().split())
budget_min_max = [tuple(map(int, input().split()))
for _ in range(n)]

total_budget = sum(min_budget for min_budget, _ in
budget_min_max)
remaining_budget = m - total_budget

allocation = [min_budget for min_budget, _ in
budget_min_max]

for i in range(n):
    min_budget, max_budget = budget_min_max[i]
    additional_budget = min(max_budget - min_budget,
remaining_budget)
    allocation[i] += additional_budget
    remaining_budget -= additional_budget

print(sum(allocation))
```

3. Bài tập mở rộng:

Bài 4.1: Phân bổ ngân sách tối ưu sao cho tất cả các kênh đều có mức ngân sách gần bằng nhau nhất

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và m ($1 \leq n \leq 10$, $1 \leq m \leq 10^5$) là số kênh quảng cáo và tổng ngân sách.
- Mỗi dòng tiếp theo chứa hai số nguyên là mức ngân sách tối thiểu và tối đa cho mỗi kênh.

Output:

- In ra phân bổ ngân sách tối ưu sao cho tất cả các kênh đều có mức ngân sách gần bằng nhau nhất.

Ví dụ:

Input	Output
3 100 20 50 30 40 10 20	34 33 33

Bài 4.2: Tìm mức ngân sách tối thiểu để tất cả các kênh đều đạt được mức ngân sách tối đa

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và m ($1 \leq n \leq 10$, $1 \leq m \leq 10^5$) là số kênh quảng cáo và tổng ngân sách.
- Mỗi dòng tiếp theo chứa hai số nguyên là mức ngân sách tối thiểu và tối đa cho mỗi kênh.

Output:

- In ra mức ngân sách tối thiểu để tất cả các kênh đều đạt được mức ngân sách tối đa.

Ví dụ:

Input	Output
3 110 20 50	110

30 40	
10 20	

Bài 4.3: Tính toán phân bổ ngân sách khi mỗi kênh cần một mức ngân sách cố định và phần còn lại được chia đều

Input:

- Dòng đầu tiên chứa hai số nguyên dương n và m ($1 \leq n \leq 10$, $1 \leq m \leq 10^5$) là số kênh quảng cáo và tổng ngân sách.
- Mỗi dòng tiếp theo chứa hai số nguyên là mức ngân sách tối thiểu và tối đa cho mỗi kênh.

Output:

- In ra phân bổ ngân sách tối ưu khi mỗi kênh cần một mức ngân sách cố định và phần còn lại được chia đều.

Ví dụ:

Input	Output
3 120 20 50 30 40 10 20	40 40 40

ĐỀ SỐ 24

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tính toán chi phí điện nước hàng tháng	UTILITYCOST*	Bàn phím	Màn hình	4
Bài 2	Quản lý danh sách mua sắm	SHOPPINGLIST*	Bàn phím	Màn hình	5
Bài 3	Xây dựng bảng điểm học sinh	GRADEBOOK*	Bàn phím	Màn hình	5
Bài 4	Dự đoán thu nhập theo chu kỳ kinh doanh	INCOMEPREDICTION*	Bàn phím	Màn hình	6

Bài 1: Tính toán chi phí điện nước hàng tháng

Mỗi tháng, một gia đình cần tính toán chi phí điện nước dựa trên mức tiêu thụ của từng thiết bị trong nhà. Bạn được yêu cầu tính tổng chi phí hàng tháng và xác định thiết bị nào tiêu tốn nhiều nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số thiết bị trong nhà.
- Mỗi dòng tiếp theo chứa hai số nguyên là mức tiêu thụ điện nước (kWh hoặc m^3) và chi phí cho mỗi đơn vị tiêu thụ (đơn vị tiền tệ).

Output

- In ra tổng chi phí hàng tháng và tên thiết bị tiêu tốn nhiều nhất.

Ví dụ:

Input	Output
4 100 50 200 30 150 40	9500

50 100	
--------	--

Bài 2: Quản lý danh sách mua sắm

Một gia đình cần quản lý danh sách mua sắm hàng tuần và tính toán tổng chi phí dự kiến. Bạn được yêu cầu lập danh sách mua sắm và tính toán tổng chi phí.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số mặt hàng cần mua.
- Mỗi dòng tiếp theo chứa tên mặt hàng, số lượng và giá tiền mỗi đơn vị của mặt hàng đó.

Output

- In ra tổng chi phí dự kiến và tên mặt hàng đắt nhất.

Ví dụ:

Input	Output
3 Sữa 2 500 Gạo 10 200 Thịt 5 400	3500

Bài 3: Xây dựng bảng điểm học sinh

Một giáo viên cần xây dựng bảng điểm của học sinh trong lớp. Bạn được yêu cầu tính toán tổng điểm của mỗi học sinh và xếp loại học sinh dựa trên tổng điểm đó.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số học sinh trong lớp.
- Mỗi dòng tiếp theo chứa tên học sinh và điểm của các môn học (tối đa 5 môn học).

Output

- In ra tên học sinh có tổng điểm cao nhất và xếp loại của học sinh đó.

Ví dụ:

Input	Output
3 Minh 9 8 10 Hằng 7 8 9 An 8 7 9	Minh Giỏi

Bài 4. Dự đoán thu nhập theo chu kỳ kinh doanh

Một doanh nghiệp cần dự đoán thu nhập cho các chu kỳ kinh doanh tiếp theo dựa trên dữ liệu thu nhập trước đó. Bạn được yêu cầu tính toán tổng thu nhập dự đoán cho chu kỳ tiếp theo.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số chu kỳ kinh doanh.
- Mỗi dòng tiếp theo chứa một số nguyên là thu nhập trong mỗi chu kỳ.

Output:

- In ra tổng thu nhập dự đoán cho chu kỳ tiếp theo dựa trên trung bình thu nhập của các chu kỳ trước đó.

Ví dụ:

Input	Output
6 120 150 130 140 150 150	840

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tính toán chi phí điện nước hàng tháng

1. Phân tích giải thuật:

- Sử dụng vòng lặp để tính chi phí tiêu thụ cho mỗi thiết bị (mức tiêu thụ nhân với chi phí mỗi đơn vị).
- Tìm chi phí tiêu thụ cao nhất và tổng chi phí của tất cả các thiết bị.

2. Code tham khảo:

```
n = int(input())
max_cost = 0
total_cost = 0

for _ in range(n):
    consumption, cost_per_unit = map(int,
input().split())
    device_cost = consumption * cost_per_unit
    total_cost += device_cost
    if device_cost > max_cost:
        max_cost = device_cost

print(total_cost, max_cost)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm thiết bị tiêu thụ ít nhất và in ra chi phí của thiết bị đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số thiết bị trong nhà.
- Mỗi dòng tiếp theo chứa hai số nguyên là mức tiêu thụ điện nước (kWh hoặc m^3) và chi phí cho mỗi đơn vị tiêu thụ (đơn vị tiền tệ).

Output:

- In ra chi phí của thiết bị tiêu thụ ít nhất.

Ví dụ:

Input	Output
4 100 50 200 30 150 40	5000

50 100	
--------	--

Bài 1.2: Tính chi phí trung bình của tất cả các thiết bị trong nhà

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số thiết bị trong nhà.
- Mỗi dòng tiếp theo chứa hai số nguyên là mức tiêu thụ điện nước (kWh hoặc m^3) và chi phí cho mỗi đơn vị tiêu thụ (đơn vị tiền tệ).

Output:

- In ra chi phí trung bình của tất cả các thiết bị trong nhà.

Ví dụ:

Input	Output
4 100 50 200 30 150 40 50 100	6000

Bài 1.3: So sánh chi phí tiêu thụ của tháng này với tháng trước và tính phần trăm chênh lệch

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số thiết bị trong nhà.
- Dòng thứ hai chứa n số nguyên là chi phí tiêu thụ của tháng trước.
- Dòng thứ ba chứa n số nguyên là chi phí tiêu thụ của tháng này.

Output:

- In ra phần trăm chênh lệch chi phí tiêu thụ giữa tháng này và tháng trước.

Ví dụ:

Input	Output
-------	--------

4	10%
1000 2000 3000 4000	
1100 2200 3300 4400	

Bài 2: Quản lý danh sách mua sắm

1. Phân tích giải thuật:

1. Dùng vòng lặp để tính toán chi phí của từng mặt hàng (số lượng nhân với giá tiền mỗi đơn vị).
2. Tìm mặt hàng có chi phí cao nhất và tổng chi phí dự kiến.

2. Code tham khảo

```
n = int(input())
total_cost = 0
max_cost = 0
max_item = ""

for _ in range(n):
    item, quantity, price_per_unit = input().split()
    quantity = int(quantity)
    price_per_unit = int(price_per_unit)
    item_cost = quantity * price_per_unit
    total_cost += item_cost
    if item_cost > max_cost:
        max_cost = item_cost
        max_item = item

print(total_cost, max_item)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm mặt hàng rẻ nhất trong danh sách mua sắm và in ra tên mặt hàng đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số mặt hàng cần mua.

● Mỗi dòng tiếp theo chứa tên mặt hàng, số lượng và giá tiền mỗi đơn vị của mặt hàng đó.

Output:

- In ra tên mặt hàng rẻ nhất và chi phí của mặt hàng đó.

Ví dụ:

Input	Output
3 Sữa 2 500 Gạo 10 200 Thịt 5 400	Gạo 2000

Bài 2.2: Tính tổng chi phí dự kiến khi giảm giá một số mặt hàng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số mặt hàng cần mua.
- Dòng tiếp theo chứa n số nguyên là phần trăm giảm giá của từng mặt hàng.

Output:

- In ra tổng chi phí dự kiến sau khi áp dụng giảm giá.

Ví dụ:

Input	Output
3 Sữa 2 500 10 Gạo 10 200 5 Thịt 5 400 20	3150

Bài 2.3: Tìm tên và chi phí của mặt hàng có phần trăm giảm giá cao nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số mặt hàng cần mua.

- Dòng tiếp theo chứa n số nguyên là phần trăm giảm giá của từng mặt hàng.

Output:

- In ra tên mặt hàng có phần trăm giảm giá cao nhất và chi phí của mặt hàng đó sau khi áp dụng giảm giá.

Ví dụ:

Input	Output
3 Sữa 2 500 10 Gạo 10 200 5 Thịt 5 400 20	Thịt 1600

Bài 3: Xây dựng bảng điểm học sinh

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng điểm của từng học sinh.
2. Dựa trên tổng điểm để xếp loại học sinh và tìm học sinh có tổng điểm cao nhất.

2. Code tham khảo

```
n = int(input())
max_total = 0
top_student = ""
grade = ""

for _ in range(n):
    data = input().split()
    name = data[0]
    scores = list(map(int, data[1:]))
    total = sum(scores)

    if total > max_total:
        max_total = total
```



```
top_student = name
if total >= 25:
    grade = "Giỏi"
elif total >= 15:
    grade = "Khá"
else:
    grade = "Trung bình"

print(top_student, grade)
```

3. Bài tập mở rộng:

Bài 3.1: Tìm học sinh có tổng điểm thấp nhất và xếp loại học sinh đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số học sinh trong lớp.
- Mỗi dòng tiếp theo chứa tên học sinh và điểm của các môn học (tối đa 5 môn học).

Output:

- In ra tên học sinh có tổng điểm thấp nhất và xếp loại của học sinh đó.

Ví dụ:

Input	Output
3 Minh 9 8 10 Hằng 7 8 9 An 8 7 9	An Trung bình

Bài 3.2: Tính điểm trung bình của cả lớp và xếp loại theo điểm trung bình

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số học sinh trong lớp.

● Mỗi dòng tiếp theo chứa tên học sinh và điểm của các môn học (tối đa 5 môn học).

Output:

- In ra điểm trung bình của cả lớp và xếp loại theo điểm trung bình.

Ví dụ:

Input	Output
3 Minh 9 8 10 Hằng 7 8 9 An 8 7 9	8.3 Khá

Bài 3.3: So sánh điểm trung bình của hai học kỳ và xác định học sinh có sự tiến bộ lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số học sinh trong lớp.
- Dòng thứ hai chứa n số nguyên là điểm trung bình học kỳ 1 của các học sinh.
- Dòng thứ ba chứa n số nguyên là điểm trung bình học kỳ 2 của các học sinh.

Output:

- In ra tên học sinh có sự tiến bộ lớn nhất và mức độ tiến bộ.

Ví dụ:

Input	Output
3 8.7 7.5 8.0 10.0 8.5 9.0	Minh 1.3

Bài 4: Dự đoán thu nhập theo chu kỳ kinh doanh

1. Phân tích giải thuật

1. Tính trung bình thu nhập của các chu kỳ trước đó.

2. Dự đoán thu nhập cho chu kỳ tiếp theo dựa trên trung bình thu nhập.

2. Code tham khảo

```
n = int(input())
incomes = list(map(int, input().split()))

average_income = sum(incomes) // n
predicted_income = average_income

print(predicted_income)
```

3. Bài tập mở rộng:

Bài 4.1: Dự đoán thu nhập cho chu kỳ tiếp theo nếu thu nhập tăng đều 5% mỗi chu kỳ

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số chu kỳ kinh doanh.
- Mỗi dòng tiếp theo chứa một số nguyên là thu nhập trong mỗi chu kỳ.

Output:

- In ra tổng thu nhập dự đoán cho chu kỳ tiếp theo nếu thu nhập tăng đều 5% mỗi chu kỳ.

Ví dụ:

Input	Output
6 120 150 130 140 150 150	882

Bài 4.2: Dự đoán thu nhập cho chu kỳ tiếp theo nếu thu nhập giảm 10% mỗi chu kỳ

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số chu kỳ kinh doanh.
- Mỗi dòng tiếp theo chứa một số nguyên là thu nhập trong mỗi chu kỳ.

Output:

- In ra tổng thu nhập dự đoán cho chu kỳ tiếp theo nếu thu nhập giảm 10% mỗi chu kỳ.

Ví dụ:

Input	Output
6 120 150 130 140 150 150	756

Bài 4.3: So sánh thu nhập giữa hai chu kỳ liên tiếp và xác định chu kỳ có sự thay đổi lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số chu kỳ kinh doanh.
- Mỗi dòng tiếp theo chứa một số nguyên là thu nhập trong mỗi chu kỳ.

Output:

- In ra chu kỳ có sự thay đổi thu nhập lớn nhất và mức độ thay đổi.

Ví dụ:

Input	Output
6 120 150 130 140 150 150	4-5 10

ĐỀ SỐ 25

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Phân tích dữ liệu người dùng trên ứng dụng	USERANALYSIS*	Bàn phím	Màn hình	4
Bài 2	Tối ưu hóa danh sách công việc hàng ngày	TASKOPTIMIZATION*	Bàn phím	Màn hình	5
Bài 3	Quản lý và sắp xếp danh sách sản phẩm	PRODUCTMANAGEMENT*	Bàn phím	Màn hình	5
Bài 4	Dự đoán doanh số bán hàng dựa trên dữ liệu quá khứ	SALESPREDICTION*	Bàn phím	Màn hình	6

Bài 1: Phân tích dữ liệu người dùng trên ứng dụng

Một ứng dụng cần phân tích dữ liệu người dùng để cải thiện trải nghiệm người dùng. Bạn được yêu cầu tính tổng thời gian người dùng sử dụng ứng dụng và xác định khoảng thời gian sử dụng ứng dụng phổ biến nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số phiên đăng nhập của người dùng.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của mỗi phiên (đơn vị tính là phút).

Output

- In ra tổng thời gian sử dụng ứng dụng và khoảng thời gian sử dụng phổ biến nhất.

Ví dụ:

Input	Output
3 10 30	240 30-60

50 80	
30 60	

Bài 2: Tối ưu hóa danh sách công việc hàng ngày

Một nhân viên cần quản lý và tối ưu hóa danh sách công việc hàng ngày để hoàn thành công việc một cách hiệu quả nhất. Bạn được yêu cầu tính tổng thời gian hoàn thành công việc và sắp xếp công việc theo thứ tự ưu tiên.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số công việc.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian hoàn thành công việc và mức độ ưu tiên (1 là cao nhất, 3 là thấp nhất).

Output

- In ra tổng thời gian hoàn thành công việc và danh sách công việc được sắp xếp theo thứ tự ưu tiên.

Ví dụ:

Input	Output
3 5 1 3 2 7 1	15

Bài 3: Quản lý và sắp xếp danh sách sản phẩm

Một cửa hàng cần quản lý danh sách sản phẩm và sắp xếp sản phẩm theo thứ tự giá từ thấp đến cao để thu hút khách hàng. Bạn được yêu cầu tính tổng giá trị hàng hóa và sắp xếp danh sách sản phẩm.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số sản phẩm.
- Mỗi dòng tiếp theo chứa tên sản phẩm và giá tiền của sản phẩm đó.

Output

- In ra tổng giá trị hàng hóa và danh sách sản phẩm đã được sắp xếp theo thứ tự giá từ thấp đến cao.

Ví dụ:

Input	Output
3 Táo 500 Bánh mì 200 Sữa 800	1500

Bài 4. Dự đoán doanh số bán hàng dựa trên dữ liệu quá khứ

Một doanh nghiệp cần dự đoán doanh số bán hàng cho tháng tới dựa trên dữ liệu bán hàng của các tháng trước. Bạn được yêu cầu tính tổng doanh số dự đoán cho tháng tới.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh số của từng tháng.

Output:

- In ra tổng doanh số dự đoán cho tháng tới dựa trên trung bình doanh số của các tháng trước đó.

Ví dụ:

Input	Output
6 1500 1600 1700 1500 1600 1500	1600

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Phân tích dữ liệu người dùng trên ứng dụng

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng thời gian sử dụng ứng dụng.
2. Dùng mảng để lưu số phiên đăng nhập tại từng khoảng thời gian và xác định khoảng thời gian phổ biến nhất.

2. Code tham khảo:

```
n = int(input())
total_time = 0
time_slots = [0] * 1440 # Mảng lưu số phiên tại mỗi
phút trong ngày

for _ in range(n):
    start, end = map(int, input().split())
    total_time += (end - start)
    for i in range(start, end):
        time_slots[i] += 1

# Tìm khoảng thời gian phổ biến nhất
max_users = max(time_slots)
popular_time = time_slots.index(max_users)

# Tìm khoảng thời gian liên tục phổ biến nhất
start_time = popular_time
end_time = popular_time
for i in range(popular_time, 1440):
    if time_slots[i] == max_users:
        end_time = i
    else:
        break

print(total_time, f"{start_time}-{end_time + 1}")
```

3. Bài tập mở rộng:

Bài 1.1: Tính thời gian trung bình mỗi phiên đăng nhập

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số phiên đăng nhập.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của mỗi phiên (đơn vị tính là phút).

Output:

- In ra thời gian trung bình mỗi phiên đăng nhập.

Ví dụ:

Input	Output
3 10 30 50 80 30 60	80

Bài 1.2: Xác định thời gian mà số người dùng đồng thời nhiều nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số phiên đăng nhập.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của mỗi phiên (đơn vị tính là phút).

Output:

- In ra thời điểm mà số người dùng đồng thời cao nhất.

Ví dụ:

Input	Output
3 10 30 50 80 30 60	30

Bài 1.3: So sánh thời gian sử dụng ứng dụng giữa hai nhóm người dùng khác nhau

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 105$) là số phiên đăng nhập.

- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của mỗi phiên (đơn vị tính là phút).

Output:

- In ra thời gian sử dụng ứng dụng của từng nhóm và nhóm có thời gian sử dụng dài hơn.

Ví dụ:

Input	Output
3 2 10 30 50 80 30 60 100 150 120 170	240 150 Group 1

Bài 2: Tối ưu hóa danh sách công việc hàng ngày

1. Phân tích giải thuật:

1. Tính tổng thời gian hoàn thành công việc.
2. Sắp xếp danh sách công việc theo mức độ ưu tiên.

2. Code tham khảo

```
n = int(input())
tasks = []

for _ in range(n):
    time, priority = map(int, input().split())
    tasks.append((priority, time))

tasks.sort() # Sắp xếp theo mức độ ưu tiên

total_time = sum(time for _, time in tasks)
sorted_tasks = [time for _, time in tasks]
```

```
print(total_time)
print(*sorted_tasks)
```

3. Bài tập mở rộng:

Bài 2.1: Tìm công việc có thời gian hoàn thành dài nhất trong danh sách

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số công việc.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian hoàn thành công việc và mức độ ưu tiên (1 là cao nhất, 3 là thấp nhất).

Output:

- In ra thời gian hoàn thành của công việc dài nhất.

Ví dụ:

Input	Output
3 5 1 3 2 7 1	7

Bài 2.2: Tính tổng thời gian hoàn thành công việc khi có thêm một khoản thời gian nghỉ giữa các công việc

Input:

- Dòng đầu tiên chứa số nguyên dương n và t ($1 \leq n \leq 10^5$, $1 \leq t \leq 100$) là số công việc và thời gian nghỉ giữa các công việc.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian hoàn thành công việc và mức độ ưu tiên (1 là cao nhất, 3 là thấp nhất).

Output:

- In ra tổng thời gian hoàn thành công việc bao gồm cả thời gian nghỉ.

Ví dụ:

Input	Output
3 2	19

5 1	
3 2	
7 1	

Bài 2.3: Tìm công việc có mức độ ưu tiên thấp nhất và thời gian hoàn thành ngắn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số công việc.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian hoàn thành công việc và mức độ ưu tiên (1 là cao nhất, 3 là thấp nhất).

Output:

- In ra thời gian hoàn thành của công việc có mức độ ưu tiên thấp nhất.

Ví dụ:

Input	Output
3 5 1 3 2 7 1	3

Bài 3: Quản lý và sắp xếp danh sách sản phẩm

1. Phân tích giải thuật:

1. Tính tổng giá trị hàng hóa.
2. Sắp xếp danh sách sản phẩm theo giá tiền.

2. Code tham khảo

```
n = int(input())
products = []

for _ in range(n):
```

```
name, price = input().split()
price = int(price)
products.append((price, name))

products.sort()

total_value = sum(price for price, _ in products)
sorted_products = [name for _, name in products]

print(total_value)
print(*sorted_products)
```

3. Bài tập mở rộng:

Bài 3.1: Tìm sản phẩm có giá tiền cao nhất và in ra tên sản phẩm đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số sản phẩm.
- Mỗi dòng tiếp theo chứa tên sản phẩm và giá tiền của sản phẩm đó.

Output:

- In ra tên sản phẩm có giá tiền cao nhất.

Ví dụ:

Input	Output
3 Táo 500 Bánh mì 200 Sữa 800	Sữa

Bài 3.2: Tính tổng giá trị hàng hóa sau khi giảm giá 10% cho tất cả các sản phẩm

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số sản phẩm.

- Mỗi dòng tiếp theo chứa tên sản phẩm và giá tiền của sản phẩm đó.

Output:

- In ra tổng giá trị hàng hóa sau khi giảm giá 10% cho tất cả các sản phẩm.

Ví dụ:

Input	Output
3 Táo 500 Bánh mì 200 Sữa 800	1350

Bài 3.3: So sánh giá trị hàng hóa trước và sau khi giảm giá 10% và tính phần trăm chênh lệch

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số sản phẩm.

- Mỗi dòng tiếp theo chứa tên sản phẩm và giá tiền của sản phẩm đó.

Output:

- In ra phần trăm chênh lệch giá trị hàng hóa trước và sau khi giảm giá 10%.

Ví dụ:

Input	Output
3 Táo 500 Bánh mì 200 Sữa 800	10%

Bài 4: Dự đoán doanh số bán hàng dựa trên dữ liệu quá khứ

1. Phân tích giải thuật

1. Tính trung bình doanh số của các tháng trước đó.

2. Dự đoán doanh số cho tháng tới dựa trên trung bình doanh số.

2. Code tham khảo

```
n = int(input())
sales = list(map(int, input().split()))

average_sales = sum(sales) // n
predicted_sales = average_sales

print(predicted_sales)
```

3. Bài tập mở rộng:

Bài 4.1: Dự đoán doanh số cho tháng tới nếu doanh số tăng đều 5% mỗi tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh số của từng tháng.

Output:

- In ra tổng doanh số dự đoán cho tháng tới nếu doanh số tăng đều 5% mỗi tháng.

Ví dụ:

Input	Output
6 1500 1600 1700 1500 1600 1500	1680

Bài 4.2: Dự đoán doanh số cho tháng tới nếu doanh số giảm 10% mỗi tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh số của từng tháng.

Output:

- In ra tổng doanh số dự đoán cho tháng tới nếu doanh số giảm 10% mỗi tháng.

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

Ví dụ:

Input	Output
6 1500 1600 1700 1500 1600 1500	1440

Bài 4.3: So sánh doanh số giữa hai tháng gần nhất và xác định tháng có doanh số cao hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh số của từng tháng.

Output:

- In ra tháng có doanh số cao hơn và mức chênh lệch doanh số giữa hai tháng gần nhất.

Ví dụ:

Input	Output
6 1500 1600 1700 1500 1600 1500	100

ĐỀ SỐ 26

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tính toán chi phí sản xuất	PRODUCTIONCO ST*	Bàn phím	Màn hình	4
Bài 2	Quản lý lịch trình chuyến bay	FLIGHTSCHEDUL E*	Bàn phím	Màn hình	5
Bài 3	Theo dõi và phân tích doanh số bán hàng	SALESANALYSIS *	Bàn phím	Màn hình	5
Bài 4	Dự đoán lưu lượng truy cập website	TRAFFICPREDICT ION*	Bàn phím	Màn hình	6

Bài 1: Tính toán chi phí sản xuất

Một nhà máy sản xuất nhiều loại sản phẩm và cần tính toán tổng chi phí sản xuất hàng tháng. Bạn được yêu cầu tính tổng chi phí sản xuất và xác định sản phẩm có chi phí sản xuất cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số loại sản phẩm.
- Mỗi dòng tiếp theo chứa hai số nguyên là số lượng sản phẩm và chi phí sản xuất mỗi đơn vị sản phẩm.

Output

- In ra tổng chi phí sản xuất và loại sản phẩm có chi phí sản xuất cao nhất.

Ví dụ:

Input	Output
4 100 50 200 80 150 60 50 200	36000

Bài 2: Quản lý lịch trình chuyến bay

Một sân bay cần quản lý lịch trình các chuyến bay hàng ngày. Bạn được yêu cầu tính tổng số thời gian bay và tìm chuyến bay dài nhất trong ngày.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số chuyến bay trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian khởi hành và thời gian hạ cánh của một chuyến bay (đơn vị tính là giờ).

Output

- In ra tổng số giờ bay và chuyến bay có thời gian dài nhất.

Ví dụ:

Input	Output
3 2 5 6 9 3 8	14 5-9

Bài 3: Theo dõi và phân tích doanh số bán hàng

Một cửa hàng cần theo dõi doanh số bán hàng hàng tháng để phân tích hiệu suất kinh doanh. Bạn được yêu cầu tính tổng doanh số và xác định tháng có doanh số cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng cần theo dõi.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh số của cửa hàng trong tháng đó.

Output

- In ra tổng doanh số trong n tháng và tháng có doanh số cao nhất.

Ví dụ:

Input	Output
6	105000

15000	
18000	
20000	
12000	
18000	
22000	

Bài 4. Dự đoán lưu lượng truy cập website

Một website cần dự đoán lưu lượng truy cập cho tháng tới dựa trên dữ liệu truy cập của các tháng trước đó. Bạn được yêu cầu tính toán tổng lưu lượng truy cập dự đoán cho tháng tới.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là lưu lượng truy cập trong từng tháng (đơn vị: lượt truy cập).

Output:

- In ra tổng lưu lượng truy cập dự đoán cho tháng tới dựa trên trung bình lưu lượng của các tháng trước đó.

Ví dụ:

Input	Output
5 200000 210000 220000 230000 240000	220000

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tính toán chi phí sản xuất

1. Phân tích giải thuật:

1. Dùng vòng lặp để tính chi phí sản xuất cho mỗi loại sản phẩm (số lượng nhân với chi phí đơn vị).
2. Tìm chi phí sản xuất cao nhất và tổng chi phí của tất cả các sản phẩm.

2. Code tham khảo:

```
n = int(input())
total_cost = 0
max_cost = 0

for _ in range(n):
    quantity, unit_cost = map(int, input().split())
    product_cost = quantity * unit_cost
    total_cost += product_cost
    if product_cost > max_cost:
        max_cost = product_cost

print(total_cost, max_cost)
```

3. Bài tập mở rộng:

Bài 1.1: Tìm sản phẩm có chi phí sản xuất thấp nhất và in ra chi phí của sản phẩm đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số loại sản phẩm.
- Mỗi dòng tiếp theo chứa hai số nguyên là số lượng sản phẩm và chi phí sản xuất mỗi đơn vị sản phẩm.

Output:

- In ra chi phí của sản phẩm có chi phí sản xuất thấp nhất.

Ví dụ:

Input	Output
4 100 50 200 80	5000

150 60	
50 200	

Bài 1.2: Tính tổng chi phí sản xuất khi có thêm một khoản phí cố định cho mỗi sản phẩm

Input:

- Dòng đầu tiên chứa số nguyên dương n và f ($1 \leq n \leq 10^5$, $1 \leq f \leq 10000$) là số loại sản phẩm và phí cố định cho mỗi sản phẩm.
- Mỗi dòng tiếp theo chứa hai số nguyên là số lượng sản phẩm và chi phí sản xuất mỗi đơn vị sản phẩm.

Output:

- In ra tổng chi phí sản xuất sau khi thêm phí cố định.

Ví dụ:

Input	Output
4 1000 100 50 200 80 150 60 50 200	38000

Bài 1.3: So sánh chi phí sản xuất giữa hai sản phẩm và xác định sản phẩm nào tốn kém hơn

Input:

- Dòng đầu tiên chứa hai số nguyên dương m và n ($1 \leq m, n \leq 10^5$) là hai loại sản phẩm.
- Mỗi dòng tiếp theo chứa hai số nguyên là số lượng sản phẩm và chi phí sản xuất mỗi đơn vị sản phẩm.

Output:

- In ra loại sản phẩm nào có chi phí sản xuất cao hơn và mức chênh lệch.

Ví dụ:

Input	Output
200 100 80	8000
150 200 50	

Bài 2: Quản lý lịch trình chuyến bay

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính thời gian bay của mỗi chuyến bay.
2. Tìm chuyến bay dài nhất và tổng thời gian bay của tất cả các chuyến.

2. Code tham khảo

```
n = int(input())
total_flight_time = 0
max_flight_time = 0
max_flight_range = ""

for _ in range(n):
    start, end = map(int, input().split())
    flight_time = end - start
    total_flight_time += flight_time
    if flight_time > max_flight_time:
        max_flight_time = flight_time
        max_flight_range = f"{start}-{end}"

print(total_flight_time, max_flight_range)
```

3. Bài tập mở rộng:

Bài 2.1: Tính thời gian bay trung bình cho tất cả các chuyến bay trong ngày

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số chuyến bay trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian khởi hành và thời gian hạ cánh của một chuyến bay (đơn vị tính là giờ).

Output:

- In ra thời gian bay trung bình cho tất cả các chuyến bay.

Ví dụ:

Input	Output
3	4 . 6 7
2 5	
6 9	
3 8	

Bài 2.2: Xác định khoảng thời gian trong ngày có nhiều chuyến bay nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số chuyến bay trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian khởi hành và thời gian hạ cánh của một chuyến bay (đơn vị tính là giờ).

Output:

- In ra khoảng thời gian có nhiều chuyến bay nhất và số lượng chuyến bay trong khoảng đó.

Ví dụ:

Input	Output
3	3 – 8 2
2 5	
6 9	
3 8	

Bài 2.3: So sánh thời gian bay giữa hai chuyến bay và xác định chuyến bay dài hơn

Input:

- Dòng đầu tiên chứa số nguyên dương m và n ($1 \leq m, n \leq 10^5$) là thời gian khởi hành và hạ cánh của hai chuyến bay.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian khởi hành và thời gian hạ cánh của một chuyến bay (đơn vị tính là giờ).

Output:

● In ra chuyến bay nào dài hơn và mức chênh lệch thời gian bay giữa hai chuyến.

Ví dụ:

Input	Output
2 5 3 8	1

Bài 3: Theo dõi và phân tích doanh số bán hàng

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng doanh số của tất cả các tháng.
2. Tìm tháng có doanh số cao nhất.

2. Code tham khảo

```
n = int(input())
total_sales = 0
max_sales = 0
max_sales_month = 0

for i in range(n):
    sales = int(input())
    total_sales += sales
    if sales > max_sales:
        max_sales = sales
        max_sales_month = i + 1

print(total_sales, max_sales_month)
```

3. Bài tập mở rộng:

Bài 3.1: Tính doanh số trung bình hàng tháng trong n tháng

Input:

● Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng cần theo dõi.

- Mỗi dòng tiếp theo chứa một số nguyên là doanh số của cửa hàng trong tháng đó.

Output:

- In ra doanh số trung bình hàng tháng.

Ví dụ:

Input	Output
6 15000 18000 20000 12000 18000 22000	17500

Bài 3.2: So sánh doanh số giữa hai tháng liên tiếp và xác định tháng có sự tăng trưởng lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng cần theo dõi.

- Mỗi dòng tiếp theo chứa một số nguyên là doanh số của cửa hàng trong tháng đó.

Output:

- In ra tháng có sự tăng trưởng doanh số lớn nhất và mức chênh lệch doanh số.

Ví dụ:

Input	Output
6 15000 18000 20000	5-6 4000

12000	
18000	
22000	
6	5-6 4000
15000	

Bài 3.3: Xác định tháng có doanh số thấp nhất và in ra mức doanh số của tháng đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng cần theo dõi.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh số của cửa hàng trong tháng đó.

Output:

- In ra tháng có doanh số thấp nhất và mức doanh số của tháng đó.

Ví dụ:

Input	Output
6	4 12000
15000	
18000	
20000	
12000	
18000	
22000	

Bài 4: Dự đoán lưu lượng truy cập website

1. Phân tích giải thuật

1. Tính trung bình lưu lượng truy cập của các tháng trước đó.
2. Dự đoán lưu lượng truy cập cho tháng tới dựa trên trung bình này.

2. Code tham khảo

```
n = int(input())
traffic = list(map(int, input().split()))

average_traffic = sum(traffic) // n
predicted_traffic = average_traffic

print(predicted_traffic)
```

3. Bài tập mở rộng:

Bài 4.1: Dự đoán lưu lượng truy cập nếu lưu lượng tăng đều 10% mỗi tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là lưu lượng truy cập trong từng tháng (đơn vị: lượt truy cập).

Output:

- In ra lưu lượng truy cập dự đoán cho tháng tới nếu lưu lượng tăng đều 10% mỗi tháng.

Ví dụ:

Input	Output
5	264000
200000	
210000	
220000	
230000	
240000	

Bài 4.2: So sánh lưu lượng truy cập giữa hai tháng và xác định tháng nào có lưu lượng cao hơn

Input:

- Dòng đầu tiên chứa số nguyên dương m và n ($1 \leq m, n \leq 12$) là lưu lượng truy cập của hai tháng.

- Mỗi dòng tiếp theo chứa một số nguyên là lưu lượng truy cập trong từng tháng (đơn vị: lượt truy cập).

Output:

- In ra tháng nào có lưu lượng truy cập cao hơn và mức chênh lệch.

Ví dụ:

Input	Output
220000 230000	10000

Bài 4.3: Tính lưu lượng truy cập trung bình hàng tháng trong n tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.

- Mỗi dòng tiếp theo chứa một số nguyên là lưu lượng truy cập trong từng tháng (đơn vị: lượt truy cập).

Output:

- In ra lưu lượng truy cập trung bình hàng tháng.

Ví dụ:

Input	Output
5 200000 210000 220000 230000 240000	220000

ĐỀ SỐ 27

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Quản lý lượng khách hàng hàng ngày	DAILYCUSTOMERS*	Bàn phím	Màn hình	4
Bài 2	Tối ưu hóa số lượng nhân viên làm việc	EMPLOYEEOPTIMIZATION*	Bàn phím	Màn hình	5
Bài 3	Tính toán doanh thu từ cửa hàng trực tuyến	ONLINEREVENUE*	Bàn phím	Màn hình	5
Bài 4	Dự đoán thời gian hoàn thành dự án	PROJECTPREDICTION*	Bàn phím	Màn hình	6

Bài 1: Quản lý lượng khách hàng hàng ngày

Một cửa hàng cần quản lý và theo dõi lượng khách hàng mỗi ngày để điều chỉnh chiến lược kinh doanh. Bạn được yêu cầu tính tổng số lượng khách hàng trong tuần và xác định ngày có lượng khách hàng đông nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày cần theo dõi.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng khách hàng của cửa hàng trong từng ngày.

Output

- In ra tổng số lượng khách hàng trong tuần và ngày có lượng khách hàng đông nhất.

Ví dụ:

Input	Output
7 20 15 30	140 3

25	
20	
15	
15	

Bài 2: Tối ưu hóa số lượng nhân viên làm việc

Một công ty cần tối ưu hóa số lượng nhân viên làm việc để đạt hiệu quả cao nhất mà vẫn đảm bảo đủ nhân lực. Bạn được yêu cầu tính tổng số giờ làm việc trong tuần và xác định ca làm việc có số giờ cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ca làm việc trong tuần.
- Mỗi dòng tiếp theo chứa một số nguyên là số giờ làm việc của mỗi ca.

Output

- In ra tổng số giờ làm việc trong tuần và ca làm việc có số giờ cao nhất.

Ví dụ:

Input	Output
7 6 8 6 5 8 6 3	42 6

Bài 3: Tính toán doanh thu từ cửa hàng trực tuyến

Một cửa hàng trực tuyến cần tính toán doanh thu hàng tháng từ các đơn hàng. Bạn được yêu cầu tính tổng doanh thu trong tháng và xác định đơn hàng có giá trị cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số đơn hàng trong tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là giá trị của từng đơn hàng.

Output

- In ra tổng doanh thu trong tháng và giá trị của đơn hàng cao nhất.

Ví dụ:

Input	Output
5 2000 3000 4000 1000 5000	15000 4000

Bài 4. Dự đoán thời gian hoàn thành dự án

Một nhóm làm việc cần dự đoán thời gian hoàn thành một dự án dựa trên các công việc đã hoàn thành. Bạn được yêu cầu tính tổng thời gian hoàn thành dự án dựa trên thời gian của từng công việc đã hoàn thành.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số công việc đã hoàn thành.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian hoàn thành của từng công việc (đơn vị: giờ).

Output:

- In ra tổng thời gian hoàn thành dự án và công việc mất nhiều thời gian nhất.

Ví dụ:

Input	Output
4 5	30 12

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

8	
12	
5	

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Quản lý lượng khách hàng hàng ngày

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng số lượng khách hàng của tất cả các ngày.
2. Tìm ngày có lượng khách hàng đông nhất.

2. Code tham khảo:

```
n = int(input())
total_customers = 0
max_customers = 0
max_day = 0

for i in range(n):
    customers = int(input())
    total_customers += customers
    if customers > max_customers:
        max_customers = customers
        max_day = i + 1

print(total_customers, max_day)
```

3. Bài tập mở rộng:

Bài 1.1: Tính số lượng khách hàng trung bình mỗi ngày trong tuần

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày cần theo dõi.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng khách hàng của cửa hàng trong từng ngày.

Output:

- In ra số lượng khách hàng trung bình mỗi ngày trong tuần.

Ví dụ:

Input	Output
7	20
20	
15	
30	

25	
20	
15	
15	

Bài 1.2: Xác định ngày có lượng khách hàng ít nhất và in ra số lượng khách hàng trong ngày đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày cần theo dõi.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng khách hàng của cửa hàng trong từng ngày.

Output:

- In ra số lượng khách hàng của ngày có lượng khách hàng ít nhất.

Ví dụ:

Input	Output
7	15
20	
15	
30	
25	
20	
15	
15	

Bài 1.3: So sánh lượng khách hàng giữa hai ngày liên tiếp và xác định ngày có sự tăng trưởng lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày cần theo dõi.

- Mỗi dòng tiếp theo chứa một số nguyên là lượng khách hàng của cửa hàng trong từng ngày.

Output:

- In ra ngày có sự tăng trưởng lượng khách hàng lớn nhất và mức độ tăng trưởng.

Ví dụ:

Input	Output
7	4-5 10
20	
15	
30	
25	
20	
15	
15	

Bài 2: Tối ưu hóa số lượng nhân viên làm việc

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng số giờ làm việc của tất cả các ca.
2. Tìm ca làm việc có số giờ cao nhất.

2. Code tham khảo

```
n = int(input())
total_hours = 0
max_hours = 0
max_shift = 0

for i in range(n):
    hours = int(input())
    total_hours += hours
    if hours > max_hours:
```

```
max_hours = hours
max_shift = i + 1

print(total_hours, max_shift)
```

3. Bài tập mở rộng:

Bài 2.1: Tính số giờ làm việc trung bình của mỗi ca trong tuần

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ca làm việc trong tuần.
- Mỗi dòng tiếp theo chứa một số nguyên là số giờ làm việc của mỗi ca.

Output:

- In ra số giờ làm việc trung bình của mỗi ca trong tuần.

Ví dụ:

Input	Output
7	6
6	
8	
6	
5	
8	
6	
3	

Bài 2.2: Xác định ca làm việc có số giờ ít nhất và in ra số giờ của ca đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ca làm việc trong tuần.
- Mỗi dòng tiếp theo chứa một số nguyên là số giờ làm việc của mỗi ca.

Output:

- In ra số giờ làm việc của ca có số giờ ít nhất.

Ví dụ:

Input	Output
7 6 8 6 5 8 6 3	3

Bài 2.3: So sánh số giờ làm việc giữa hai ca liên tiếp và xác định ca có sự tăng trưởng lớn nhất về số giờ làm việc

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ca làm việc trong tuần.
- Mỗi dòng tiếp theo chứa một số nguyên là số giờ làm việc của mỗi ca.

Output:

- In ra ca có sự tăng trưởng số giờ làm việc lớn nhất và mức độ tăng trưởng.

Ví dụ:

Input	Output
7 6 8 6 5	3-4 2

8	
6	
3	

Bài 3: Tính toán doanh thu từ cửa hàng trực tuyến

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng doanh thu của tất cả các đơn hàng.
2. Tìm đơn hàng có giá trị cao nhất.

2. Code tham khảo

```
n = int(input())
total_revenue = 0
max_order_value = 0

for i in range(n):
    order_value = int(input())
    total_revenue += order_value
    if order_value > max_order_value:
        max_order_value = order_value

print(total_revenue, max_order_value)
```

3. Bài tập mở rộng:

Bài 3.1: Tính giá trị trung bình của các đơn hàng trong tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số đơn hàng trong tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là giá trị của từng đơn hàng.

Output:

- In ra giá trị trung bình của các đơn hàng trong tháng.

Ví dụ:

Input	Output
5	3000
2000	
3000	
4000	
1000	
5000	

Bài 3.2: Xác định đơn hàng có giá trị thấp nhất và in ra giá trị của đơn hàng đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số đơn hàng trong tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là giá trị của từng đơn hàng.

Output:

- In ra giá trị của đơn hàng có giá trị thấp nhất.

Ví dụ:

Input	Output
5	1000
2000	
3000	
4000	
1000	
5000	

Bài 3.3: So sánh giá trị đơn hàng giữa hai đơn hàng liên tiếp và xác định đơn hàng có sự tăng trưởng lớn nhất

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số đơn hàng trong tháng.

- Mỗi dòng tiếp theo chứa một số nguyên là giá trị của từng đơn hàng.

Output:

- In ra đơn hàng có sự tăng trưởng lớn nhất về giá trị và mức độ tăng trưởng.

Ví dụ:

Input	Output
5	3-4 1000
2000	
3000	
4000	
1000	
5000	

Bài 4: Dự đoán thời gian hoàn thành dự án

1. Phân tích giải thuật

1. Sử dụng vòng lặp để tính tổng thời gian hoàn thành của tất cả các công việc.
2. Tìm công việc có thời gian hoàn thành dài nhất.

2. Code tham khảo

```
n = int(input())
total_time = 0
max_time = 0

for i in range(n):
    time = int(input())
    total_time += time
    if time > max_time:
        max_time = time
print(total_time, max_time)
```

3. Bài tập mở rộng:

Bài 4.1: Tính thời gian trung bình hoàn thành mỗi công việc

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số công việc đã hoàn thành.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian hoàn thành của từng công việc (đơn vị: giờ).

Output:

- In ra thời gian trung bình hoàn thành mỗi công việc.

Ví dụ:

Input	Output
4	7.5
5	
8	
12	
5	

Bài 4.2: Xác định công việc có thời gian hoàn thành ngắn nhất và in ra thời gian của công việc đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số công việc đã hoàn thành.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian hoàn thành của từng công việc (đơn vị: giờ).

Output:

- In ra thời gian của công việc có thời gian hoàn thành ngắn nhất.

Ví dụ:

Input	Output
4	5
5	
8	

12	
5	

Bài 4.3: So sánh thời gian hoàn thành giữa hai công việc liên tiếp và xác định công việc nào có thời gian dài hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số công việc đã hoàn thành.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian hoàn thành của từng công việc (đơn vị: giờ).

Output:

- In ra công việc nào có thời gian hoàn thành dài hơn giữa hai công việc liên tiếp và mức chênh lệch.

Ví dụ:

Input	Output
4	2 – 3 4
5	
8	
12	
5	

ĐỀ SỐ 28

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Tính toán lượng nước tiêu thụ trong hộ gia đình	WATERCONSUMPTION*	Bàn phím	Màn hình	4
Bài 2	Quản lý thời gian rảnh trong ngày	FREETIMEOPTIMIZATION*	Bàn phím	Màn hình	5
Bài 3	Theo dõi và phân tích chi phí hàng tháng	MONTHLYEXPENSES*	Bàn phím	Màn hình	5
Bài 4	Dự đoán thời gian giao hàng	DELIVERYPREDICTION*	Bàn phím	Màn hình	6

Bài 1: Tính toán lượng nước tiêu thụ trong hộ gia đình

Một hộ gia đình cần quản lý và theo dõi lượng nước tiêu thụ hàng tháng để tối ưu hóa chi phí. Bạn được yêu cầu tính tổng lượng nước tiêu thụ trong tháng và xác định ngày có lượng nước tiêu thụ cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 31$) là số ngày trong tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng nước tiêu thụ trong từng ngày (đơn vị: lít).

Output

- In ra tổng lượng nước tiêu thụ trong tháng và ngày có lượng nước tiêu thụ cao nhất.

Ví dụ:

Input	Output
7 50 60 80	450 3

70	
60	
50	
80	

Bài 2: Quản lý thời gian rảnh trong ngày

Một người cần quản lý thời gian rảnh trong ngày để thực hiện các hoạt động cá nhân. Bạn được yêu cầu tính tổng thời gian rảnh trong ngày và xác định khoảng thời gian dài nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số khoảng thời gian rảnh trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của khoảng thời gian rảnh (đơn vị: giờ).

Output

- In ra tổng số giờ rảnh trong ngày và khoảng thời gian rảnh dài nhất.

Ví dụ:

Input	Output
3 1 3 5 7 10 12	6 2-4

Bài 3: Theo dõi và phân tích chi phí hàng tháng

Một người cần theo dõi và phân tích chi phí hàng tháng để tối ưu hóa ngân sách. Bạn được yêu cầu tính tổng chi phí hàng tháng và xác định loại chi phí cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10$) là số loại chi phí hàng tháng.
- Mỗi dòng tiếp theo chứa một xâu ký tự là tên chi phí và một số nguyên là

giá trị của chi phí đó.

Output

- In ra tổng chi phí hàng tháng và tên chi phí cao nhất.

Ví dụ:

Input	Output
4 Điện 5000 Nước 2000 Gas 3000 Internet 5000	15000 Điện

Bài 4. Dự đoán thời gian giao hàng

Một công ty giao hàng cần dự đoán thời gian giao hàng cho các đơn hàng trong ngày dựa trên thời gian giao hàng trước đó. Bạn được yêu cầu tính tổng thời gian giao hàng và dự đoán thời gian cho đơn hàng tiếp theo.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số đơn hàng đã giao.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian giao hàng của từng đơn hàng (đơn vị: phút).

Output:

- In ra tổng thời gian giao hàng và dự đoán thời gian cho đơn hàng tiếp theo dựa trên trung bình thời gian giao hàng của các đơn trước.

Ví dụ:

Input	Output
5 20 25 30 15	110 22

20

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Tính toán lượng nước tiêu thụ trong hộ gia đình

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng lượng nước tiêu thụ của tất cả các ngày.
2. Tìm ngày có lượng nước tiêu thụ cao nhất.

2. Code tham khảo:

```
n = int(input())
total_water = 0
max_water = 0
max_day = 0

for i in range(n):
    water = int(input())
    total_water += water
    if water > max_water:
        max_water = water
        max_day = i + 1

print(total_water, max_day)
```

3. Bài tập mở rộng:

Bài 1.1: Tính lượng nước tiêu thụ trung bình mỗi ngày trong tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 31$) là số ngày trong tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng nước tiêu thụ trong từng ngày (đơn vị: lít).

Output:

- In ra lượng nước tiêu thụ trung bình mỗi ngày trong tháng.

Ví dụ:

Input	Output
7	64.29
50	

60	
80	
70	
60	
50	
80	

Bài 1.2: Xác định ngày có lượng nước tiêu thụ ít nhất và in ra số lượng nước tiêu thụ trong ngày đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 31$) là số ngày trong tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng nước tiêu thụ trong từng ngày (đơn vị: lít).

Output:

- In ra số lượng nước tiêu thụ của ngày có lượng nước tiêu thụ ít nhất.

Ví dụ:

Input	Output
7 50 60 80 70 60 50 80	50

Bài 1.3: So sánh lượng nước tiêu thụ giữa hai ngày liên tiếp và xác định ngày có sự tăng trưởng lớn nhất về lượng nước tiêu thụ

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 31$) là số ngày trong tháng.
- Mỗi dòng tiếp theo chứa một số nguyên là lượng nước tiêu thụ trong từng ngày (đơn vị: lít).

Output:

- In ra ngày có sự tăng trưởng lượng nước tiêu thụ lớn nhất và mức độ tăng trưởng.

Ví dụ:

Input	Output
7	2-3 20
50	
60	
80	
70	
60	
50	
80	

Bài 2: Quản lý thời gian rảnh trong ngày

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng thời gian rảnh của tất cả các khoảng thời gian.
2. Tìm khoảng thời gian rảnh dài nhất.

2. Code tham khảo

```
n = int(input())
total_free_time = 0
max_free_time = 0
max_free_range = ""
```

```
for _ in range(n):  
    start, end = map(int, input().split())  
    free_time = end - start  
    total_free_time += free_time  
    if free_time > max_free_time:  
        max_free_time = free_time  
        max_free_range = f"{start}-{end}"  
  
print(total_free_time, max_free_range)
```

3. Bài tập mở rộng:

Bài 2.1: Tính thời gian rảnh trung bình trong ngày

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số khoảng thời gian rảnh trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của khoảng thời gian rảnh (đơn vị: giờ).

Output:

- In ra thời gian rảnh trung bình trong ngày.

Ví dụ:

Input	Output
3 1 3 5 7 10 12	2

Bài 2.2: Xác định khoảng thời gian rảnh ngắn nhất và in ra số giờ của khoảng thời gian đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số khoảng thời gian rảnh trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của khoảng thời gian rảnh (đơn vị: giờ).

Output:

- In ra số giờ của khoảng thời gian rảnh ngắn nhất.

Ví dụ:

Input	Output
3	2
1 3	
5 7	
10 12	

Bài 2.3: So sánh thời gian rảnh giữa hai khoảng thời gian liên tiếp và xác định khoảng nào dài hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số khoảng thời gian rảnh trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của khoảng thời gian rảnh (đơn vị: giờ).

Output:

- In ra khoảng thời gian rảnh dài hơn giữa hai khoảng liên tiếp và mức chênh lệch.

Ví dụ:

Input	Output
3	2-3 0
1 3	
5 7	
10 12	

Bài 3: Theo dõi và phân tích chi phí hàng tháng

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng chi phí hàng tháng.
2. Tìm loại chi phí có giá trị cao nhất.

2. Code tham khảo

```
n = int(input())
total_expenses = 0
max_expense = 0
max_expense_name = ""

for _ in range(n):
    data = input().split()
    name = data[0]
    value = int(data[1])
    total_expenses += value
    if value > max_expense:
        max_expense = value
        max_expense_name = name

print(total_expenses, max_expense_name)
```

3. Bài tập mở rộng:

Bài 3.1: Tính chi phí trung bình hàng tháng cho tất cả các loại chi phí

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10$) là số loại chi phí hàng tháng.
- Mỗi dòng tiếp theo chứa một xâu ký tự là tên chi phí và một số nguyên là giá trị của chi phí đó.

Output:

- In ra chi phí trung bình hàng tháng.

Ví dụ:

Input	Output
4 Điện 5000 Nước 2000	3750

Gas 3000	
Internet 5000	

Bài 3.2: Xác định loại chi phí thấp nhất và in ra giá trị của loại chi phí đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10$) là số loại chi phí hàng tháng.
- Mỗi dòng tiếp theo chứa một xâu ký tự là tên chi phí và một số nguyên là giá trị của chi phí đó.

Output:

- In ra giá trị của loại chi phí thấp nhất.

Ví dụ:

Input	Output
4 Điện 5000 Nước 2000 Gas 3000 Internet 5000	5000

Bài 3.3: So sánh giá trị giữa hai loại chi phí liên tiếp và xác định loại nào cao hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10$) là số loại chi phí hàng tháng.
- Mỗi dòng tiếp theo chứa một xâu ký tự là tên chi phí và một số nguyên là giá trị của chi phí đó.

Output:

- In ra loại chi phí cao hơn giữa hai loại chi phí liên tiếp và mức chênh lệch.

Ví dụ:

Input	Output
-------	--------

4 Điện 5000 Nước 2000	Gas-Internet 2000
Gas 3000 Internet 5000	

Bài 4: Dự đoán thời gian giao hàng

1. Phân tích giải thuật

1. Sử dụng vòng lặp để tính tổng thời gian giao hàng.
2. Tính trung bình thời gian giao hàng và sử dụng giá trị này để dự đoán thời gian cho đơn hàng tiếp theo.

2. Code tham khảo

```
n = int(input())
total_time = 0

for i in range(n):
    time = int(input())
    total_time += time

average_time = total_time // n
print(total_time, average_time)
```

3. Bài tập mở rộng:

Bài 4.1: Dự đoán thời gian giao hàng nếu thời gian giao hàng tăng đều 5% cho mỗi đơn hàng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số đơn hàng đã giao.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian giao hàng của từng đơn hàng (đơn vị: phút).

Output:

- In ra thời gian dự đoán cho đơn hàng tiếp theo nếu thời gian giao hàng tăng đều 5% cho mỗi đơn hàng.

Ví dụ:

Input	Output
5	23
20	
25	
30	
15	
20	

Bài 4.2: So sánh thời gian giao hàng giữa hai đơn hàng liên tiếp và xác định đơn hàng nào mất nhiều thời gian hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số đơn hàng đã giao.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian giao hàng của từng đơn hàng (đơn vị: phút).

Output:

- In ra đơn hàng nào mất nhiều thời gian giao hơn và mức chênh lệch thời gian.

Ví dụ:

Input	Output
5	2-3 5
20	
25	
30	
15	
20	

Bài 4.3: Tính thời gian giao hàng trung bình cho tất cả các đơn hàng trong ngày

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số đơn hàng đã giao.
- Mỗi dòng tiếp theo chứa một số nguyên là thời gian giao hàng của từng đơn hàng (đơn vị: phút).

Output:

- In ra thời gian giao hàng trung bình cho tất cả các đơn hàng trong ngày.

Ví dụ:

Input	Output
5 20 25 30 15 20	22

ĐỀ SỐ 29

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Phân tích dữ liệu nhân khẩu học	DEMOGRAPHICANALYSIS*	Bàn phím	Màn hình	4
Bài 2	Quản lý thời gian học tập và nghỉ ngơi	STUDYTIMEMANAGEMENT*	Bàn phím	Màn hình	5
Bài 3	Tính toán lợi nhuận từ đầu tư	INVESTMENTPROFIT*	Bàn phím	Màn hình	5
Bài 4	Dự đoán chi phí bảo trì hàng tháng	MAINTENANCEPREDICTION*	Bàn phím	Màn hình	6

Bài 1: Phân tích dữ liệu nhân khẩu học

Một công ty cần phân tích dữ liệu nhân khẩu học để xác định độ tuổi trung bình và độ tuổi phổ biến nhất trong một nhóm dân số. Bạn được yêu cầu tính độ tuổi trung bình và tìm độ tuổi có nhiều người nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số người trong nhóm.
- Mỗi dòng tiếp theo chứa một số nguyên là độ tuổi của từng người.

Output

- In ra độ tuổi trung bình của nhóm dân số và độ tuổi phổ biến nhất.

Ví dụ:

Input	Output
5 25 30 22 30	28.2 30

34	
----	--

Bài 2: Quản lý thời gian học tập và nghỉ ngơi

Một học sinh cần quản lý thời gian học tập và nghỉ ngơi trong ngày để đảm bảo sức khỏe và hiệu suất học tập. Bạn được yêu cầu tính tổng thời gian học tập và nghỉ ngơi, và xác định khoảng thời gian học tập dài nhất trong ngày.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số khoảng thời gian học tập trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của mỗi khoảng thời gian học tập (đơn vị: giờ).

Output

- In ra tổng số giờ học tập trong ngày và khoảng thời gian học tập dài nhất.

Ví dụ:

Input	Output
3 2 5 7 9 10 13	7 2-5

Bài 3: Tính toán lợi nhuận từ đầu tư

Một nhà đầu tư cần tính toán lợi nhuận từ các khoản đầu tư khác nhau để đánh giá hiệu quả. Bạn được yêu cầu tính tổng lợi nhuận và xác định khoản đầu tư có lợi nhuận cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số khoản đầu tư.
- Mỗi dòng tiếp theo chứa một số nguyên là lợi nhuận từ mỗi khoản đầu tư (đơn vị: nghìn đồng).

Output

- In ra tổng lợi nhuận và khoản đầu tư có lợi nhuận cao nhất.

Ví dụ:

Input	Output
4	150 50
40	
50	
30	
30	

Bài 4. Dự đoán chi phí bảo trì hàng tháng

Một công ty cần dự đoán chi phí bảo trì hàng tháng dựa trên dữ liệu chi phí của các tháng trước đó. Bạn được yêu cầu tính tổng chi phí bảo trì và dự đoán chi phí cho tháng tiếp theo dựa trên trung bình chi phí của các tháng trước.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là chi phí bảo trì của từng tháng (đơn vị: nghìn đồng).

Output:

- In ra tổng chi phí bảo trì và dự đoán chi phí cho tháng tiếp theo dựa trên trung bình chi phí của các tháng trước đó.

Ví dụ:

Input	Output
5	600 120
100	
120	
150	
100	
130	

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Phân tích dữ liệu nhân khẩu học

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng độ tuổi và tìm độ tuổi phổ biến nhất.
2. Tính độ tuổi trung bình bằng cách lấy tổng độ tuổi chia cho số người.

2. Code tham khảo:

```
n = int(input())
ages = []
age_count = {}

for _ in range(n):
    age = int(input())
    ages.append(age)
    if age in age_count:
        age_count[age] += 1
    else:
        age_count[age] = 1

average_age = sum(ages) / n
most_common_age = max(age_count, key=age_count.get)

print(round(average_age, 1), most_common_age)
```

3. Bài tập mở rộng:

Bài 1.1: Xác định độ tuổi trẻ nhất và cao tuổi nhất trong nhóm

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số người trong nhóm.
- Mỗi dòng tiếp theo chứa một số nguyên là độ tuổi của từng người.

Output:

- In ra độ tuổi trẻ nhất và cao tuổi nhất trong nhóm.

Ví dụ:

Input	Output
5	22 34
25	
30	

22	
30	
34	

Bài 1.2: Tính tỷ lệ phần trăm người dưới 25 tuổi trong nhóm

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 10^5$) là số người trong nhóm.
- Mỗi dòng tiếp theo chứa một số nguyên là độ tuổi của từng người.

Output:

- In ra tỷ lệ phần trăm người dưới 25 tuổi trong nhóm.

Ví dụ:

Input	Output
5	20.0
25	
30	
22	
30	
34	

Bài 1.3: So sánh độ tuổi trung bình giữa hai nhóm dân số và xác định nhóm nào có độ tuổi trung bình cao hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n và m ($1 \leq n, m \leq 10^5$) là số người trong hai nhóm dân số.
- Mỗi dòng tiếp theo chứa một số nguyên là độ tuổi của từng người.

Output:

- In ra nhóm nào có độ tuổi trung bình cao hơn và giá trị độ tuổi trung bình của nhóm đó.

Ví dụ:

Input	Output
3 2 25 35 33 22 28	Group 1 31.0

Bài 2: Quản lý thời gian học tập và nghỉ ngơi

1. Phân tích giải thuật:

- Sử dụng vòng lặp để tính tổng thời gian học tập của tất cả các khoảng thời gian.
- Tìm khoảng thời gian học tập dài nhất.

2. Code tham khảo

```
n = int(input())
total_study_time = 0
max_study_time = 0
max_study_range = ""

for _ in range(n):
    start, end = map(int, input().split())
    study_time = end - start
    total_study_time += study_time
    if study_time > max_study_time:
        max_study_time = study_time
        max_study_range = f"{start}-{end}"

print(total_study_time, max_study_range)
```

3. Bài tập mở rộng:

Bài 2.1: Tính thời gian học tập trung bình trong ngày

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số khoảng thời gian học tập trong ngày.

- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của mỗi khoảng thời gian học tập (đơn vị: giờ).

Output:

- In ra thời gian học tập trung bình trong ngày.

Ví dụ:

Input	Output
3 2 5 7 9 10 13	2.33

Bài 2.2: Xác định khoảng thời gian học tập ngắn nhất và in ra số giờ của khoảng thời gian đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số khoảng thời gian học tập trong ngày.

- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của mỗi khoảng thời gian học tập (đơn vị: giờ).

Output:

- In ra số giờ của khoảng thời gian học tập ngắn nhất.

Ví dụ:

Input	Output
3 2 5 7 9 10 13	2

Bài 2.3: So sánh thời gian học tập giữa hai khoảng thời gian liên tiếp và xác định khoảng nào dài hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 24$) là số khoảng thời gian học tập trong ngày.
- Mỗi dòng tiếp theo chứa hai số nguyên là thời gian bắt đầu và thời gian kết thúc của mỗi khoảng thời gian học tập (đơn vị: giờ).

Output:

- In ra khoảng thời gian học tập dài hơn giữa hai khoảng liên tiếp và mức chênh lệch.

Ví dụ:

Input	Output
3 2 5 7 9 10 13	2-3 1

Bài 3: Tính toán lợi nhuận từ đầu tư

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng lợi nhuận của tất cả các khoản đầu tư.
2. Tìm khoản đầu tư có lợi nhuận cao nhất.

2. Code tham khảo

```
n = int(input())
total_profit = 0
max_profit = 0

for _ in range(n):
    profit = int(input())
    total_profit += profit
    if profit > max_profit:
        max_profit = profit
```



```
print(total_profit, max_profit)
```

3. Bài tập mở rộng:

Bài 3.1: Tính lợi nhuận trung bình từ các khoản đầu tư

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số khoản đầu tư.
- Mỗi dòng tiếp theo chứa một số nguyên là lợi nhuận từ mỗi khoản đầu tư (đơn vị: nghìn đồng).

Output:

- In ra lợi nhuận trung bình từ các khoản đầu tư.

Ví dụ:

Input	Output
4	37.5
40	
50	
30	
30	

Bài 3.2: Xác định khoản đầu tư có lợi nhuận thấp nhất và in ra giá trị lợi nhuận của khoản đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số khoản đầu tư.
- Mỗi dòng tiếp theo chứa một số nguyên là lợi nhuận từ mỗi khoản đầu tư (đơn vị: nghìn đồng).

Output:

- In ra giá trị lợi nhuận của khoản đầu tư có lợi nhuận thấp nhất.

Ví dụ:

Input	Output
4	30
40	
50	
30	
30	

Bài 3.3: So sánh lợi nhuận giữa hai khoản đầu tư liên tiếp và xác định khoản nào có lợi nhuận cao hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số khoản đầu tư.
- Mỗi dòng tiếp theo chứa một số nguyên là lợi nhuận từ mỗi khoản đầu tư (đơn vị: nghìn đồng).

Output:

- In ra khoản đầu tư nào có lợi nhuận cao hơn giữa hai khoản liên tiếp và mức chênh lệch.

Ví dụ:

Input	Output
4	1-2 10
40	
50	
30	
30	

Bài 4: Dự đoán chi phí bảo trì hàng tháng

1. Phân tích giải thuật

1. Sử dụng vòng lặp để tính tổng chi phí bảo trì.

2. Tính trung bình chi phí bảo trì và sử dụng giá trị này để dự đoán chi phí cho tháng tiếp theo.

2. Code tham khảo

```
n = int(input())
total_cost = 0

for i in range(n):
    cost = int(input())
    total_cost += cost

average_cost = total_cost // n
print(total_cost, average_cost)
```

3. Bài tập mở rộng:

Bài 4.1: Dự đoán chi phí bảo trì nếu chi phí tăng đều 10% mỗi tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là chi phí bảo trì của từng tháng (đơn vị: nghìn đồng).

Output:

- In ra chi phí dự đoán cho tháng tiếp theo nếu chi phí tăng đều 10% mỗi tháng.

Ví dụ:

Input	Output
5	132
100	
120	
150	
100	
130	

Bài 4.2: So sánh chi phí bảo trì giữa hai tháng liên tiếp và xác định tháng nào có chi phí cao hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là chi phí bảo trì của từng tháng (đơn vị: nghìn đồng).

Output:

- In ra tháng nào có chi phí bảo trì cao hơn giữa hai tháng liên tiếp và mức chênh lệch chi phí.

Ví dụ:

Input	Output
5	2-3 30
100	
120	
150	
100	
130	

Bài 4.3: Tính chi phí bảo trì trung bình hàng tháng cho tất cả các tháng đã qua

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là chi phí bảo trì của từng tháng (đơn vị: nghìn đồng).

Output:

- In ra chi phí bảo trì trung bình hàng tháng.

Ví dụ:

Input	Output
5	120

Trung tâm đào tạo và phát triển năng lực AI – LEARN, K122/23 Phan Thanh, Đà Nẵng.
ĐT: 0906.579.469-0914.753.469.

100	
120	
150	
100	
130	

ĐỀ SỐ 30

TỔNG QUAN ĐỀ THI

Bài	Tên bài	File bài làm	Đầu vào	Kết quả	Điểm
Bài 1	Phân tích doanh số cửa hàng	SHOPSALSA ANALYSIS*	Bàn phím	Màn hình	4
Bài 2	Quản lý và tối ưu hóa lịch trình tập luyện	WORKOUTSCHEDULE*	Bàn phím	Màn hình	5
Bài 3	Tính toán chi phí vận chuyển hàng hóa	SHIPPINGCOST*	Bàn phím	Màn hình	5
Bài 4	Dự đoán doanh thu hàng năm từ mô hình kinh doanh	BUSINESSFORECAST*	Bàn phím	Màn hình	6

Bài 1: Phân tích doanh số cửa hàng

Một cửa hàng cần phân tích doanh số hàng tháng để đưa ra quyết định kinh doanh. Bạn được yêu cầu tính tổng doanh số của cửa hàng trong tháng và xác định mặt hàng bán chạy nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 50$) là số mặt hàng được bán trong tháng.
- Mỗi dòng tiếp theo chứa tên mặt hàng và số lượng bán ra của mặt hàng đó.

Output

- In ra tổng số lượng bán ra trong tháng và tên mặt hàng bán chạy nhất.

Ví dụ:

Input	Output
4 Táo 500 Chuối 300 Sữa 400 Bánh mì 300	1500 Táo

Bài 2: Quản lý và tối ưu hóa lịch trình tập luyện

Một người cần quản lý thời gian tập luyện trong tuần để đảm bảo hiệu quả sức khỏe. Bạn được yêu cầu tính tổng thời gian tập luyện và xác định ngày có thời gian tập luyện dài nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày tập luyện trong tuần.
- Mỗi dòng tiếp theo chứa số giờ tập luyện của từng ngày.

Output

- In ra tổng số giờ tập luyện trong tuần và ngày có thời gian tập luyện dài nhất.

Ví dụ:

Input	Output
5	15 3
2	
3	
4	
3	
3	

Bài 3: Tính toán chi phí vận chuyển hàng hóa

Một công ty cần tính toán chi phí vận chuyển hàng hóa dựa trên trọng lượng của các gói hàng. Bạn được yêu cầu tính tổng chi phí vận chuyển và xác định gói hàng có chi phí cao nhất.

Input

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số gói hàng cần vận chuyển.
- Mỗi dòng tiếp theo chứa trọng lượng của gói hàng và chi phí vận chuyển tương ứng (đơn vị: kg và nghìn đồng).

Output

- In ra tổng chi phí vận chuyển và chi phí cao nhất trong số các gói hàng.

Ví dụ:

Input	Output
3	750 300
10 200	
15 300	
20 250	

Bài 4. Dự đoán doanh thu hàng năm từ mô hình kinh doanh

Một doanh nghiệp cần dự đoán doanh thu hàng năm dựa trên mô hình kinh doanh hiện tại và dữ liệu từ các tháng trước đó. Bạn được yêu cầu tính tổng doanh thu của năm và dự đoán doanh thu cho năm tiếp theo dựa trên trung bình doanh thu của các tháng trước.

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh thu của từng tháng (đơn vị: triệu đồng).

Output:

- In ra tổng doanh thu của năm và dự đoán doanh thu cho năm tiếp theo dựa trên trung bình doanh thu của các tháng trước đó.

Ví dụ:

Input	Output
6	1800 300
250	
300	
320	
270	
330	
330	

HẾT

HƯỚNG DẪN VÀ CODE THAM KHẢO

Bài 1: Phân tích doanh số cửa hàng

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng số lượng bán ra của tất cả các mặt hàng.
2. Tìm mặt hàng bán chạy nhất.

2. Code tham khảo:

```
n = int(input())
total_sales = 0
max_sales = 0
best_selling_item = ""

for _ in range(n):
    item, quantity = input().split()
    quantity = int(quantity)
    total_sales += quantity
    if quantity > max_sales:
        max_sales = quantity
        best_selling_item = item

print(total_sales, best_selling_item)
```

3. Bài tập mở rộng:

Bài 1.1: Tính doanh số trung bình của mỗi mặt hàng trong tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 50$) là số mặt hàng được bán trong tháng.
- Mỗi dòng tiếp theo chứa tên mặt hàng và số lượng bán ra của mặt hàng đó.

Output:

- In ra doanh số trung bình của mỗi mặt hàng trong tháng.

Ví dụ:

Input	Output
4	375
Táo 500	
Chuối 300	

Sữa 400	
Bánh mì 300	

Bài 1.2: Xác định mặt hàng bán chậm nhất và in ra số lượng bán của mặt hàng đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 50$) là số mặt hàng được bán trong tháng.
- Mỗi dòng tiếp theo chứa tên mặt hàng và số lượng bán ra của mặt hàng đó.

Output:

- In ra số lượng bán của mặt hàng bán chậm nhất.

Ví dụ:

Input	Output
4 Táo 500 Chuối 300 Sữa 400 Bánh mì 300	Chuối 300

Bài 1.3: So sánh doanh số giữa hai mặt hàng liên tiếp và xác định mặt hàng nào bán chạy hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 50$) là số mặt hàng được bán trong tháng.
- Mỗi dòng tiếp theo chứa tên mặt hàng và số lượng bán ra của mặt hàng đó.

Output:

- In ra mặt hàng nào bán chạy hơn giữa hai mặt hàng liên tiếp và mức chênh lệch.

Ví dụ:

Input	Output
4 Táo 500 Chuối 300 Sữa 400 Bánh mì 300	Táo-Sữa 100

Bài 2: Quản lý và tối ưu hóa lịch trình tập luyện

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng thời gian tập luyện của tất cả các ngày.
2. Tìm ngày có thời gian tập luyện dài nhất.

2. Code tham khảo

```
n = int(input())
total_workout = 0
max_workout = 0
max_day = 0

for i in range(n):
    hours = int(input())
    total_workout += hours
    if hours > max_workout:
        max_workout = hours
        max_day = i + 1

print(total_workout, max_day)
```

3. Bài tập mở rộng:

Bài 2.1: Tính thời gian tập luyện trung bình mỗi ngày trong tuần

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày tập luyện trong tuần.

- Mỗi dòng tiếp theo chứa số giờ tập luyện của từng ngày.

Output:

- In ra thời gian tập luyện trung bình mỗi ngày trong tuần.

Ví dụ:

Input	Output
5	3
2	
3	
4	
3	
3	

Bài 2.2: Xác định ngày có thời gian tập luyện ngắn nhất và in ra số giờ của ngày đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày tập luyện trong tuần.

- Mỗi dòng tiếp theo chứa số giờ tập luyện của từng ngày.

Output:

- In ra số giờ của ngày có thời gian tập luyện ngắn nhất.

Ví dụ:

Input	Output
5	2
2	
3	
4	
3	
3	

3

Bài 2.3: So sánh thời gian tập luyện giữa hai ngày liên tiếp và xác định ngày nào có thời gian dài hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 7$) là số ngày tập luyện trong tuần.
- Mỗi dòng tiếp theo chứa số giờ tập luyện của từng ngày.

Output:

- In ra ngày nào có thời gian tập luyện dài hơn giữa hai ngày liên tiếp và mức chênh lệch.

Ví dụ:

Input	Output
5	3-4 1
2	
3	
4	
3	
3	

Bài 3: Tính toán chi phí vận chuyển hàng hóa

1. Phân tích giải thuật:

1. Sử dụng vòng lặp để tính tổng chi phí vận chuyển của tất cả các gói hàng.
2. Tìm gói hàng có chi phí vận chuyển cao nhất.

2. Code tham khảo

```
n = int(input())
total_cost = 0
max_cost = 0
```

```
for _ in range(n):  
    weight, cost = map(int, input().split())  
    total_cost += cost  
  
    if cost > max_cost:  
        max_cost = cost  
  
print(total_cost, max_cost)
```

3. Bài tập mở rộng:

Bài 3.1: Tính chi phí vận chuyển trung bình của mỗi gói hàng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số gói hàng cần vận chuyển.
- Mỗi dòng tiếp theo chứa trọng lượng của gói hàng và chi phí vận chuyển tương ứng (đơn vị: kg và nghìn đồng).

Output:

- In ra chi phí vận chuyển trung bình của mỗi gói hàng.

Ví dụ:

Input	Output
3 10 200 15 300 20 250	250

Bài 3.2: Xác định gói hàng có chi phí vận chuyển thấp nhất và in ra chi phí của gói đó

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số gói hàng cần vận chuyển.
- Mỗi dòng tiếp theo chứa trọng lượng của gói hàng và chi phí vận chuyển tương ứng (đơn vị: kg và nghìn đồng).

Output:

- In ra chi phí vận chuyển của gói hàng có chi phí thấp nhất.

Ví dụ:

Input	Output
3	200
10 200	
15 300	
20 250	

Bài 3.3: So sánh chi phí vận chuyển giữa hai gói hàng liên tiếp và xác định gói nào có chi phí cao hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 100$) là số gói hàng cần vận chuyển.
- Mỗi dòng tiếp theo chứa trọng lượng của gói hàng và chi phí vận chuyển tương ứng (đơn vị: kg và nghìn đồng).

Output:

- In ra gói hàng nào có chi phí vận chuyển cao hơn giữa hai gói liên tiếp và mức chênh lệch.

Ví dụ:

Input	Output
3	1-2 100
10 200	
15 300	
20 250	

Bài 4: Dự đoán doanh thu hàng năm từ mô hình kinh doanh

1. Phân tích giải thuật

1. Sử dụng vòng lặp để tính tổng doanh thu của tất cả các tháng.

2. Tính trung bình doanh thu của các tháng trước và sử dụng giá trị này để dự đoán doanh thu cho năm tiếp theo.

2. Code tham khảo

```
n = int(input())
total_revenue = 0

for i in range(n):
    revenue = int(input())
    total_revenue += revenue

average_revenue = total_revenue // n
print(total_revenue, average_revenue)
```

3. Bài tập mở rộng:

Bài 4.1: Dự đoán doanh thu nếu doanh thu tăng đều 10% mỗi tháng

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh thu của từng tháng (đơn vị: triệu đồng).

Output:

- In ra doanh thu dự đoán cho năm tiếp theo nếu doanh thu tăng đều 10% mỗi tháng.

Ví dụ:

Input	Output
6	330
250	
300	
320	
270	
330	

330	
-----	--

Bài 4.2: So sánh doanh thu giữa hai tháng liên tiếp và xác định tháng nào có doanh thu cao hơn

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh thu của từng tháng (đơn vị: triệu đồng).

Output:

- In ra tháng nào có doanh thu cao hơn giữa hai tháng liên tiếp và mức chênh lệch doanh thu.

Ví dụ:

Input	Output
6	4-5 60
250	
300	
320	
270	
330	
330	

Bài 4.3: Tính doanh thu trung bình hàng tháng cho tất cả các tháng đã qua

Input:

- Dòng đầu tiên chứa số nguyên dương n ($1 \leq n \leq 12$) là số tháng có dữ liệu.
- Mỗi dòng tiếp theo chứa một số nguyên là doanh thu của từng tháng (đơn vị: triệu đồng).

Output:

- In ra doanh thu trung bình hàng tháng.

Ví dụ:

Input	Output
6 250	300
300 320 270 330 330	