**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 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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 ≤ n ≤ 106).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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 xâu ký tự S. Hãy tìm ký tự đầu tiên lặp lại nhiều hơn một lần trong xâu. **Input**

* Dòng đầu tiên là xâu ký tự S có độ dài không quá 105 ký tự, chỉ gồm các chữ cái latinh 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 xâu ký tự chứa các số và chữ cái. Tính tổng các chữ số lẻ và ghép với chiều dài xâu. **Input:**

* Một dòng duy nhất chứa xâu ký tự có độ dài không quá 105 ký tự. **Output:**
* In ra tổng các chữ số lẻ và chiều dài xâu.

**Ví dụ:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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 (109).
   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) |

1. **Bài tập mở rộng:**

**Bài 1.1: Tìm số Fibonacci nhỏ nhất Input:**

* + - Dòng đầu tiên chứa số nguyên dương n (1 ≤ n ≤ 105).
    - Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).  **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 ≤ n ≤ 109).  **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 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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 ≤ n ≤ 106).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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 ≤ n ≤ 106).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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 ≤ n ≤ 106).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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 xâu ký tự S có độ dài không quá 105 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** |
| abcabcde | d |  |  |

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

**Input:**

* Dòng đầu tiên là xâu ký tự S có độ dài không quá 105 ký tự.
* Dòng thứ hai là một ký tự cần đếm.  **Output:**
* In ra số lần ký tự xuất hiện trong xâu.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Input** |  | **Output** |
| Aabbcde a | 2 |  |

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

**Input:**

* Dòng đầu tiên là xâu ký tự S có độ dài không quá 105 ký tự.

**Output:**

* "YES" nếu xâu 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 105 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 105 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 105 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** | MAXNU  M\* | 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 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109). **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 ≤ n ≤ 105).
* 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 xâu**

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

* Một dòng duy nhất chứa xâu ký tự (độ dài không quá 105 ký tự). **Output**
* In ra số lượng từ trong xâu. **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 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109). **Output:**
* In ra số lớn nhất có thể tạo được bằng cách sắp xếp các số trong dãy.

**Ví dụ:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input** |  | **Output** |
| 3  3 5 9 21 |  | 95321 |  |

**HẾT**

**HƯỚNG DẪN VÀ CODE THAM KHẢO**

**Bài 1: Số Fibonacci lớn nhất**

1. **Phân tích giải thuật:** 
   1. 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).
   2. 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) |

1. **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 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**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) |

1. **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 ≤ n ≤ 105).
* 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 ≤ n ≤ 105).
* 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 ≤ k ≤ n ≤ 105).
* 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 105 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 105 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 105 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)) |

1. **Bài tập mở rộng:**

**Bài 4.1: Số nhỏ nhất có thể tạo được Input:**

* + Dòng đầu tiên chứa số nguyên dương n (1 ≤ n ≤ 105).
  + Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).  **Output:**
  + In ra số nhỏ nhất có thể tạo được bằng cách sắp xếp các số trong dãy.

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Input** |  | **Output** |
| 3  9 21 3 5 | 12359 |  |

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

**Input:**

* Dòng đầu tiên chứa số nguyên dương n (1 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).
* Dòng thứ ba chứa một số nguyên dương m (mỗi số không vượt quá 109).

**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  3 5 9  21 | 95321 |  |

**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 ≤ n ≤ 105).
* Dòng thứ hai chứa n số nguyên dương (mỗi số không vượt quá 109).

**Output:**

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

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Input** |  | **Output** |
| 4  3 5 5 2 1 | 5553221 |  |