# ĐỀ SỐ 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Bài** | **Tên bài** | **File** | **Điểm** |
| **Bài 1** | **Số nguyên tố lớn nhất** | Ex\_1\_1 | 0 |
| **Bài 2** | **Sắp xếp số tăng dần trong chuỗi** | Ex\_1\_2 | 0 |
| **Bài 3** | **Đếm cặp có tổng bằng 0** | Ex\_1\_3 | 0 |
| **Bài 4** | **Thống kê số lượng từ trong xâu** | Ex\_1\_4 | 0 |

## **Bài 1: Số nguyên tố lớn nhất**

Cho dãy số nguyên (a1, a2, ... , an), 1 ≤n ≤ 10000; với mọi i sao cho ai ≤ 108. 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 a1, a2, ... , an.

**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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  2 4 5 7 9 7 | 7  4 6 |

## **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)

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| a3b2c1 | a1b2c3 |

## **Bài 3: Đếm cặp có tổng bằng 0**

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

**Input:**

* Dòng đầu tiên chứa một số nguyên dương N (1 ≤ N ≤ 2\*105)
* Dòng thứ hai chứa dãy số A gồm N số nguyên cách nhau bởi một ký tự
* khoảng trống. (|Ai| ≤ 109)

**Output:**

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

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  1 -1 2 -2 3 -3 | 3 |

## **Bài 4: Thống kê số lượng từ trong xâu**

Cho một xâu 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 xâu ký tự.
* Chiều dài của xâu 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 xâu. Ví dụ:
* 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

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Hello, this is a test. This test is simple! | 0 từ có độ dài 1, 0 từ có độ dài 2, 2 từ có độ dài 3, 3 từ có độ dài 4, 3 từ có độ dài 5  test is |

## **Hướng dẫn giải**

### **Bài 1: Số nguyên tố lớn nhất**

**1. Phân tích:**

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:**

|  |
| --- |
| 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())) max\_prime = -1 indices = []  for i in range(n):  if is\_prime(arr[i]):  if arr[i] > max\_prime:  max\_prime = arr[i]  indices = [i + 1]  elif arr[i] == max\_prime:  indices.append(i + 1)  print(max\_prime) print(' '.join(map(str, indices))) |

### **Bài 2: Sắp xếp số tăng dần trong chuỗi**

**1. Phân tích:**

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()  # Tách số và phần không phải số numbers = list(map(int, re.findall(r'\d+', s))) parts = re.split(r'\d+', s)  # Sắp xếp các số numbers.sort()  # Ghép lại chuỗi result = '' for i in range(len(numbers)):  result += parts[i] + str(numbers[i]) if len(parts) > len(numbers):  result += parts[-1]  print(result) |

### **Bài 3: Đếm cặp có tổng bằng 0**

**1. Phân tích:**

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:

3. + 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.

4. + 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.

5. In ra giá trị của biến đém, 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()) arr = list(map(int, input().split())) count = 0 freq = {} for num in arr:  opposite = -num  if opposite in freq:  count += freq[opposite]  if num in freq:  freq[num] += 1  else:  freq[num] = 1 print(count) |

### **Bài 4: Thống kê số lượng từ trong xâu**

**1. Phân tích:**

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 Counter  s = input() words = re.findall(r'\b\w+\b', s.lower())  length\_count = [0] \* 21 for word in words:  if len(word) <= 20:  length\_count[len(word)] += 1  word\_freq = Counter(words) max\_freq = max(word\_freq.values()) most\_common\_words = sorted([w for w, f in word\_freq.items() if f == max\_freq])  output\_lengths = [] for i in range(1, 6):  output\_lengths.append(f'{length\_count[i]} từ có độ dài {i}')  print(', '.join(output\_lengths)) print(' '.join(most\_common\_words)) |

## **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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:

Giải thích: Số nguyên tố đặc biệt lớn nhất nhỏ hơn hoặc bằng 30 là 23, vì 22+33 = 4 + 9 =13 là một số nguyên tố.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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 ≤ N ≤ 2\*105).
* Dòng thứ hai chứa dãy số A gồm N số nguyên cách nhau bởi một ký
* tự khoảng trống. (|Ai| ≤ 109)

**Output:**

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

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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 ≤ N ≤ 2\*105).
* Dòng thứ hai chứa dãy số A gồm N số nguyên cách nhau bởi một ký
* tự khoảng trống. (|Ai| ≤ 109)

**Output:**

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

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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 ≤ N ≤ 2\*105).
* Dòng thứ hai chứa dãy số A gồm N số nguyên cách nhau bởi một ký
* tự khoảng trống. (|Ai| ≤ 109)

**Output:**

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

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

# ĐỀ SỐ 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Bài** | **Tên bài** | **File** | **Điểm** |
| **Bài 1** | **Số Fibonacci lớn nhất** | Ex\_2\_1 | 0 |
| **Bài 2** | **Phép nhân lớn nhất** | Ex\_2\_2 | 0 |
| **Bài 3** | **Tìm ký tự lặp** | Ex\_2\_3 | 0 |
| **Bài 4** | **Tổng chữ số lẻ và chiều dài** | Ex\_2\_4 | 0 |

## **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7  1 4 13 21 22 8 5 | 21 |

## **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  2 3 3 5 7 | 35 |

## **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| abcbade | b |

## **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| a1b3c5d7 | 16 8 |

## **Hướng dẫn giải**

### **Bài 1: Số Fibonacci lớn nhất**

**1. Phân tích:**

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 generate\_fib(max\_val):  fib = [0, 1]  while fib[-1] <= max\_val:  fib.append(fib[-1] + fib[-2])  return set(fib)  n = int(input()) arr = list(map(int, input().split())) fib\_set = generate\_fib(10\*\*9) max\_fib = -1 for num in arr:  if num in fib\_set and num > max\_fib:  max\_fib = num print(max\_fib) |

### **Bài 2: Phép nhân lớn nhất**

**1. Phân tích:**

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 = list(map(int, input().split())) arr.sort(reverse=True) max\_product = -1 for i in range(n):  for j in range(i+1, n):  if arr[i] != arr[j]:  max\_product = arr[i] \* arr[j]  break  if max\_product != -1:  break print(max\_product) |

### **Bài 3: Tìm ký tự lặp**

**1. Phân tích:**

1. Duyệt qua từng ký tự trong xâu.

2. Sử dụng một tập hợp để theo dõi các ký tự đã gặp.

**2. Code tham khảo:**

|  |
| --- |
| s = input() seen = set() result = 'No character repeats' for c in s:  if c in seen:  result = c  break  seen.add(c) print(result) |

### **Bài 4: Tổng chữ số lẻ và chiều dài**

**1. Phân tích:**

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() total\_odd = 0 for ch in s:  if ch.isdigit():  if int(ch) % 2 == 1:  total\_odd += int(ch) print(total\_odd, len(s)) |

## **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |

### **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.

**Example:**

|  |  |
| --- | --- |
| **Input** | **Output** |
|  |  |