



[DSA WEEKLY CONTEST T6 2025]. TEST 7. THAM LAM, QHD PHẦN 1

[Info](#)[Statistics](#)[Rankings](#)[Participation](#)[Submissions](#)[Virtual join](#)[DSA](#)

Bước nhảy [Submit](#)

Time limit: 1.0 / **Memory limit:** 256M**Point:** 100

Một con ếch đang đứng ở chỉ số 0 của mảng **A[]**, giả sử con ếch đang ở chỉ số **i** của mảng **A[]** thì ở mỗi lần di chuyển nó có thể nhảy sang phải với số bước nhảy lớn nhất bằng với **A[i]**, mỗi bước nhảy sẽ giúp con ếch di chuyển sang phải 1 vị trí trong mảng **A[]**. Giả sử **A[i] = 5** thì từ chỉ số **i** con ếch có thể nhảy sang phải tối đa 5 vị trí. Bạn hãy xác định xem con ếch có thể di chuyển tới phần tử cuối cùng trong mảng **A[]** được hay không?

Ví dụ **A[] = [1, 3, 2, 2, 0, 0]** thì con ếch có thể di chuyển qua các chỉ số như sau : 0->1->3->5

Gợi ý : Ở mỗi vị trí **i** xem từ đó nhảy xa nhất là bao nhiêu, nếu tại 1 vị trí bất kỳ mà nhảy được tới **N - 1** thì có thể kết luận ngay.

Đầu vào

Dòng 1 là **N** : số lượng phần tử trong mảng **A[]**

Dòng 2 là các phần tử trong mảng **A[]**

Giới hạn

$$1 \leq N \leq 10^6$$

$$0 \leq A[i] \leq 10^3$$

Đầu ra



Ví dụ :

Input 01

```
6
1 3 2 2 0 0
```

[Copy](#)

Output 01

```
28tech
```

[Copy](#)

Bộ ba tam giác [Submit](#)

Time limit: 1.0 / **Memory limit:** 256M

Point: 100

3 cạnh **a**, **b**, **c** có thể hình thành một tam giác nếu 3 cạnh này **đều là số nguyên dương** và thỏa mãn bất đẳng thức tam giác tức là tổng 2 cạnh luôn lớn hơn cạnh còn lại. Cho một mảng số nguyên **A[]** gồm **N** phần tử, bạn hãy xác định xem trong mảng **A[]** có bao nhiêu bộ số **A[i]**, **A[j]**, **A[k]** có thể tạo thành 3 cạnh của một tam giác.

Ví dụ **A[]** = [1, 6, 7, 9, 2] thì có thể tạo thành các bộ 3 số (6, 7, 9), (2, 6, 7)

Đầu vào

- Dòng 1 là **N** : số lượng phần tử trong mảng **A[]**
- Dòng 2 là các phần tử trong mảng **A[]**

Giới hạn

- $1 \leq N \leq 10^4$



Đầu ra

- In ra số bộ 3 thỏa mãn

Ví dụ :

Input 01

```
5
1 6 7 9 2
```

[Copy](#)

Output 01

```
2
```

[Copy](#)

[Tham Lam]. Bài 43. Bộ ba lớn dần

[Submit](#)

Time limit: 1.0 / **Memory limit:** 256M

Point: 100

Tiếp nối với bài bộ ba tam giác lúc trước thì **28Tech** muốn bạn kiểm tra xem một mảng số nguyên $A[]$ có tồn tại bộ ba $A[i], A[j], A[k]$ với $i < j < k$ và $A[i] < A[j] < A[k]$ hay không ? Nếu có thì hãy in ra **28tech**, ngược lại in ra **29tech**

Gợi ý : Đối với mỗi chỉ số i bạn cần xác định xem đứng trước i có giá trị nào nhỏ hơn nó và có giá trị nào đứng sau i có giá trị lớn hơn nó hay không. Nếu có thì sẽ tồn tại cặp 3 số đề bài yêu cầu. Dùng 2 mảng, 1 mảng để lưu xem mỗi chỉ số i trong mảng có giá trị nhỏ hơn đứng trước không, 1 mảng để lưu xem mỗi giá trị i trong mảng có giá trị lớn hơn đứng sau hay không. Duyệt mọi chỉ số từ 0 tới $N - 1$ và kiểm tra đồng thời giá trị của 2 mảng này là có kết quả.



Dòng 1 là **N** : số lượng phần tử trong mảng **A[]**

Dòng 2 là các phần tử trong mảng **A[]**

Giới hạn

$$1 \leq N \leq 10^6$$

$$0 \leq A[i] \leq 10^9$$

Đầu ra

In ra **28tech** nếu có thể tìm được bộ 3 lớn dần, **29tech** nếu không thể.

Ví dụ :

Input 01

```
5
1 0 0 3 4
```

[Copy](#)

Output 01

```
28tech
```

[Copy](#)

[Tham Lam]. Bài 44. Mua bitcoin

[Submit](#)

Time limit: 1.0 / **Memory limit:** 256M

Point: 100

28Tech đang có một chiến lược mua **bitcoin**, may mắn là anh ta biết trước giá của bitcoin trong N ngày liên tiếp luôn rồi nhưng không may là anh ta không biết ngày nào mua ngày nào bán để kiếm được lợi nhuận lớn nhất sau N ngày. Ở mỗi ngày anh ta có thể mua, bán hoặc không mua bán gì cả. Tuy



anh ta chỉ có thể bán vào những ngày sau ngày mua.

Bạn hãy giúp 28Tech tìm ra lợi nhuận lớn nhất có thể nhé.

Ví dụ giá bitcoin trong N ngày sẽ là [1, 3, 6, 3, 2, 4, 5, 7, 6, 1] thì anh ta sẽ mua ở ngày 1, bán ở ngày 3 lợi nhuận thu được là 5, tiếp tục mua bitcoin ở ngày thứ 5 và bán ở ngày thứ 8 lợi nhuận thu được là 5. Vậy tổng lợi nhuận lớn nhất thu được là 10. Ngày thứ 9, 10 thấy giá giảm nên sẽ không mua.

Đầu vào

Dòng 1 là **N** : số ngày

Dòng 2 là **N** số tương ứng với giá bitcoin trong **N** ngày

Giới hạn

$$1 \leq N \leq 10^6$$

$$0 \leq A[i] \leq 10^9$$

Đầu ra

In ra lợi nhuận lớn nhất mà 28Tech có thể kiếm được.

Ví dụ :

Input 01

```
6
1 6 7 9 2 3
```

[Copy](#)

Output 01

```
9
```

[Copy](#)



Time limit: 1.0 / **Memory limit:** 256M

Point: 100

Hiện tại **28Tech** đang có **N** đô la trong ngân hàng, anh ta sẽ cho bạn luôn số tiền này, thật sự rất hào phóng. Tuy nhiên bạn có thể tối đa số tiền bạn lấy về bằng cách hoán đổi 2 chữ số của N đúng 1 lần duy nhất, vậy thì bạn sẽ nhận được số tiền tối đa là bao nhiêu.

Ví dụ 28Tech có 9125 đô la trong ngân hàng thì bạn sẽ hoán đổi số 1 và 5 để có thể lấy về 9521 đô la

Đầu vào

Dòng 1 là **N**

Giới hạn

N là số nguyên không âm có không quá 1000 chữ số

Đầu ra

In ra số tiền tối đa mà bạn có thể lấy được từ 28Tech

Ví dụ :

Input 01

9125

Copy

Output 01

9521

Copy



I

Time limit: 1.0 / **Memory limit:** 256M**Point:** 100

COMBINATOR1

Ta định nghĩa tổ hợp chập k của n là:

$$C_n^k = \frac{n!}{k! \times (n - k)!}$$

Gồm q truy vấn, mỗi truy vấn gồm hai số nguyên dương k và n .

Với mỗi truy vấn cần tính $C_n^k \text{ MOD } 10^9 + 7$

- Input:
 - Dòng đầu tiên là số nguyên dương q ($q \leq 10^5$).
 - q dòng tiếp theo mỗi dòng gồm hai số nguyên dương k, n ($1 \leq k \leq n \leq 2 \cdot 10^3$)
- Output:
 - Gồm q dòng mỗi dòng là đáp án tìm được chia dư cho $10^9 + 7$.
- Sample Input:


```
3
2 5
3 7
4 9
```
- Sample Output:


```
10
35
126
```

[HSG_DP25]. Bài 42. Matrix Exponentiation - Fibonacci

Submit

Time limit: 1.0 / **Memory limit:** 256M



Matrix Exponentiation - Fibonacci

Giới hạn thời gian: 0.25s Giới hạn bộ nhớ: 256M

Hãy tìm phần dư của phép chia lấy dư số **Fibonacci** thứ n cho $10^9 + 7$.

Số **Fibonacci** thứ n (F_n) được xác định bởi dãy truy hồi sau:

$$F_0 = 0, F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad (\forall n \geq 2)$$

Input

Gồm 1 dòng chứa số nguyên n ($0 \leq n \leq 10^{18}$).

Output

Phần dư của phép chia F_n cho $10^9 + 7$.

Sample 1

Input

3

Output

2

Sample 2

Input

6

Output

8

[DP Basic]. Bài 22. Tổng chuỗi ký tự

Submit

Time limit: 1.0 / **Memory limit:** 256M

Point: 100



cho bạn lấy tính tổng các số có trong string và trả về kết quả.

Ví dụ **S** = "123" bạn cần phải tính tổng các số 1, 2, 3, 12, 123, 23.

Vì kết quả sẽ rất lớn nên bạn cần chia dư kết quả cho $1e9 + 7$.

Đầu vào

- Dòng duy nhất chứa xâu **S**

Giới hạn

- $1 \leq \text{len}(S) \leq 10^6$

Đầu ra

- In ra đáp án của bài toán

Ví dụ :

Input 01

123

Copy

Output 01

164

Copy

Input 02

282828

Copy

Output 02

Copy



[DP Basic]. Bài 23. Mua stock Submit

Time limit: 1.0 / **Memory limit:** 256M

Point: 100

Bạn được cung cấp giá của cổ phiếu **Tesla** tại **N** ngày liên tiếp, mỗi ngày bạn có thể mua, bán 1 cổ phiếu **Tesla** hoặc không thực hiện giao dịch nào cả.

Nhiệm vụ của bạn là hãy tối ưu giao dịch của mình để thu được khoản lợi nhuận lớn nhất.

Ví dụ : Cổ phiếu trong 3 ngày liên tiếp là 1, 2, 50 thì bạn có thể thu được lợi nhuận tối đa là 97 bằng cách mua cổ phiếu ở ngày 1, 2 và bán vào ngày thứ 3. Trong khi nếu cổ phiếu 3 ngày liên tiếp là 50 2 1 thì bạn lại không thu được lợi nhuận nào cả, vì không thể bán với giá cao hơn giá mua.

Đầu vào

- Dòng đầu là số bộ test **T**
- Mỗi test case gồm 2 dòng : Dòng 1 là **N** tương ứng số ngày giao dịch, dòng 2 là **N** số tương ứng với giá cổ phiếu trong **N** ngày.

Giới hạn

- $1 \leq T \leq 100$
- $1 \leq N \leq 50000$
- Giá cổ phiếu là số nguyên dương không vượt quá 10^5 .

Đầu ra

- In ra đáp án của từng test trên mỗi dòng



Input 01

```
1
3
1 2 50
```

[Copy](#)

Output 01

```
97
```

[Copy](#)

Input 02

```
1
3
5 3 1
```

[Copy](#)

Output 02

```
0
```

[Copy](#)

[HSG_DP25]. Bài 31. Cắm hoa

[Submit](#)

Time limit: 1.0 / **Memory limit:** 256M

Point: 100



cắm bó hoa thứ i vào lọ thứ j là $v(i, j)$. Để tránh đạt không bị thua cuộc trước người ở nhanh tính khác, hãy tìm 1 cách cắm sao cho tổng giá trị thẩm mỹ là lớn nhất. Chú ý rằng mỗi bó hoa chỉ được cắm vào 1 lọ và mỗi lọ cũng chỉ cắm được 1 bó hoa.

Dữ liệu: vào từ file FLOWER.INP gồm

- Dòng đầu tiên gồm 2 số nguyên K và N ($1 \leq K \leq 100, K \leq N \leq 100$)
- Tiếp theo gồm K dòng, mỗi dòng gồm N số nguyên thể hiện giá trị thẩm mỹ (các giá trị nằm trong khoảng từ -50 đến 50)

Kết quả: ghi ra file FLOWER.OUT một dòng duy nhất là tổng số giá trị thẩm mỹ cao nhất.

Ví dụ:

| FLOWER.INP | FLOWER.OUT |
|-----------------|------------|
| 3 5 | 53 |
| 7 23 -5 -24 16 | |
| 5 21 -4 10 23 | |
| -21 5 -4 -20 20 | |

Được phát triển bởi **28TECH.COM.VN**