

## BÀI THI TUYỂN DỤNG

- Đây là bài thi cá nhân.
- Bài thi gồm 2 phần:
  - Phần A: Thuật toán,
  - Phần B: Toán học và nghiên cứu định lượng.
- Ngôn ngữ lập trình được yêu cầu là Python.
- Trong phần thi “Thuật toán”, kết quả chỉ được chấp nhận nếu chính xác và không vượt quá thời gian giới hạn. Ngoài ra, bạn chỉ được sử dụng thư viện math (`import math`). Các thư viện từ Python Standard Library, numpy, scipy, pandas,... không được phép sử dụng trong phần thi này.
- Trong phần thi “Toán học và nghiên cứu định lượng”, bạn cần trình bày rõ bài làm, chứng minh, viết báo cáo giải thích về mô hình (đính kèm code chi tiết và các hình ảnh liên quan).

### A THUẬT TOÁN

Đề thi thuật toán được tham khảo từ cuộc thi Code tour 2022

**Câu 1.** Ví điện tử ZaloPay chính thức trở thành một trong những phương thức thanh toán chính trên nền tảng TikTok. Theo đó, người dùng có thể nạp TikTok xu hoặc thanh toán trên TikTok Shop bằng ví ZaloPay vô cùng thuận tiện và dễ dàng.

Để đảm bảo an toàn trong các giao dịch, các giao dịch của người dùng sẽ được mã hoá bởi một mật khẩu. Mật khẩu được tạo ra mỗi giây bằng một ma trận  $N \times N$  ( $1 \leq N \leq 10^{18}$ ) như hình dưới:

1	2	3	...	$N$
$N + 1$	$N + 2$	$N + 3$	...	$2N$
$2N + 1$	$2N + 2$	$2N + 3$	...	$3N$
...	...	...	...	...
$(N - 1)N + 1$	$(N - 1)N + 2$	$(N - 1)N + 3$	...	$N^2$

Chọn ra  $N$  ô vuông từ ma trận trên sao cho không có hai ô được chọn nào trong cùng dòng hoặc cột. Tìm tổng lớn nhất có thể của  $N$  ô vuông được chọn. Do đáp án có thể rất lớn nên chúng ta sẽ chia lấy dư cho 2027

**Dữ liệu nhập:**

Chỉ gồm một dòng duy nhất chứa số  $N$  với  $1 \leq N \leq 10^{18}$

**Dữ liệu xuất:**

Một dòng đáp án duy nhất chính là tổng lớn nhất của  $N$  ô được chọn từ ma trận trên chia lấy dư cho 2027.

**Giới hạn thời gian:** 4000ms

Test case mẫu:

Input :

566

Output :

1429

Input :

5

Output :

65

**Gợi ý code nhận input từ bàn phím:**

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

```
# Here is your code ...
```

**Câu 2.** Nhân ngày Quốc Tế Thiếu Nhi, trường của Khải và Vũ tổ chức các trò chơi cho các học sinh. Ở trò chơi đầu tiên, Vũ là quản trò và Khải là người chơi. Trò chơi đầu tiên sẽ liên quan tới các viên bi. Luật chơi cơ bản như sau:

- Có tổng cộng  $n$  lượt.
- Ở mỗi lượt Khải có 2 lựa chọn:
  - **Thêm bi** vào một chiếc hộp (ban đầu rỗng) với điều kiện lần thêm đầu tiên là 1 viên bi. Mỗi lần thêm tiếp theo mỗi lần sẽ phải thêm nhiều hơn  $c$  bi so với lần thêm trước. Ví dụ: lần thêm đầu tiên là 1 bi, lần thứ hai là  $1 + c$  bi, lần thứ ba là  $1 + c + c$  bi,...
  - **Lấy đi** 1 bi từ hộp
- Lượt chơi đầu luôn là thêm bi vào hộp.

Số bi Khải lấy từ hộp sẽ là phần thưởng của Khải. Tuy nhiên Khải phải để lại đúng  $k$  bi trong hộp cho Vũ vì nếu không Vũ sẽ rất buồn.

Hãy tìm số bi Khải lấy được sau  $n$  lượt chơi.

**Dữ liệu nhập:**

Dòng duy nhất chứa 3 số nguyên  $n, k, c$  lần lượt là tổng số lượt chơi, số bi còn lại trong hộp và số bi tăng lên sau mỗi lần thêm. ( $1 \leq n < 10^9, 0 \leq k \leq 10^9, 1 \leq c \leq 100$ )

**Dữ liệu xuất:**

Số bi mà Khải nhận được như phần thưởng của mình

Chú ý: Đề đảm bảo kết quả là duy nhất và luôn tồn tại kết quả.

**Giới hạn thời gian:** 4000ms

**Test case mẫu:**

```
Input :
9 11 1
Output :
4
```

**Gợi ý code nhận input từ bàn phím:**

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

# Here is your code ...
```

**Câu 3.** Trong lúc đang dọn nhà, Harry vô tình tìm thấy  $N$  hộp bi ma thuật. Các hộp được đánh số thứ tự từ 1 đến  $N$ , và hộp thứ  $i$  sẽ có  $A_i$  viên bi. Khi xếp các hộp cạnh nhau, Harry thấy rằng một đoạn con liên tiếp các hộp bi đều có một mức độ đẹp riêng và cứ thế thì đoạn  $[l, r]$  sẽ có mức độ đẹp được tính như sau:  $F(l, r) = \sum_{i=l}^r |A_{i+1} - A_i| \cdot (r - l)$  Harry muốn chọn ra một đoạn  $[l, r], (l < r)$  trong các hộp bi của mình để trang trí, và mức độ đẹp đó không nhỏ hơn  $X$ . Hãy giúp Harry tìm xem có bao nhiêu đoạn như vậy.

**Dữ liệu nhập:**

Dòng đầu tiên chứa 2 số nguyên dương  $N$  và  $X$  ( $2 \leq N \leq 10^5, 1 \leq X \leq 10^{15}$ )

Dòng tiếp theo chứa  $N$  số nguyên dương  $A_i$  ( $1 \leq A_i \leq 10^7$ )

**Dữ liệu xuất:**

Một số duy nhất chứa số lượng đoạn  $[l, r]$  thoả mãn

**Giới hạn thời gian:** 4000ms

**Test case mẫu:**

```
Input :
6 5
1 5 2 3 1 6
```

Output:

11

### Gợi ý code nhận input từ bàn phím:

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

a = [int(i) for i in input().split()]

# Here is your code ...
```

**B TOÁN HỌC VÀ NGHIÊN CỨU ĐỊNH LƯỢNG****Câu 1.**

Cho  $m_t = \sum_{k=0}^p c_k t^k$ ,  $t = 0, \pm 1, \pm 2, \dots$ ,  $p \geq 1, c_p \neq 0$ .

Ký hiệu:  $\nabla P_t = P_t - P_{t-1}$  và  $\nabla^n P_t = \nabla(\nabla^{n-1} P_t)$ , với  $n \in \mathbb{N}, n \geq 2$ ,  $P_t$  là một đa thức.

Chứng minh rằng:

a)  $\nabla m_t$  là một đa thức bậc  $p - 1$ .

b) Chứng minh  $\nabla^{p+1} m_t = 0$

**Câu 2.**

Xét phương trình  $x^n - x - n = 0$ .

a) Chứng minh rằng phương trình đã cho có nghiệm duy nhất  $u_n \in (1, 2]$ .

b) Chứng minh  $(u_n)_{n \in \mathbb{N}}$  là dãy giảm.

c) Tính giới hạn  $l = \lim_{n \rightarrow \infty} u_n$ .

**Câu 3.**

Đọc bài báo theo link sau:

<https://www.quantstart.com/articles/ARIMA-GARCH-Trading-Strategy-on-the-SP500-Stock-Market-Index-Using-R>

Trong file data đính kèm:

Dùng mô hình ARIMA và ARIMA-GARCH dự đoán giá của tất cả cổ phiếu trong tương lai và chỉ ra mô hình nào ứng với cổ phiếu nào cho ra kết quả tốt nhất (in-sample; out-sample tự chọn và sử dụng các kiểm định giả thiết phù hợp).

Giải thích lý do tại sao mô hình đã chọn là mô hình phù hợp.

**Câu 4.**

Dùng mô hình Markowitz để tìm nghiệm tối ưu của bài toán: Danh mục tối ưu theo mô hình Markowitz (Markowitz' optimal portfolio).

Tìm giá trị lớn nhất của

$$\omega^T \mu - \frac{\delta}{2} \omega^T \Sigma \omega$$

với

- $\omega$  là véc-tơ trọng số của danh mục (the vector of portfolio weights)
- $\mu$  là véc-tơ kỳ vọng của lợi nhuận (the vector of expected excess returns)

- $\Sigma$  là ma trận covariance (the covariance matrix)
- $\delta$  là hệ số lo ngại rủi ro, hệ số này cho biết sự cân bằng giữa rủi ro và lợi nhuận (the risk aversion coefficient).

trong 2 trường hợp

- a) Được phép bán khống (Short selling is permitted)
- b) Không được phép bán khống (Short selling is not permitted)

**Lưu ý:** Danh mục đầu tư gồm 10 cổ phiếu trong thư mục dữ liệu đính kèm.