

# Bài 1. Sliding Window

## ◆ Bài toán:

Cho dãy số  $A[]$  gồm có  $N$  phần tử và số nguyên dương  $K$ .

Hãy tìm dãy con liên tiếp có  **$K$  phần tử** sao cho dãy con đó có **tổng lớn nhất**.

Nếu có nhiều dãy con có cùng tổng lớn nhất thì in ra dãy con **xuất hiện đầu tiên**.

## ◆ Input:

- Dòng đầu tiên: hai số nguyên  $N$  và  $K$  — số lượng phần tử của mảng và độ dài dãy con cần xét.
- Dòng thứ hai:  $N$  số nguyên — các phần tử trong mảng  $A$ .

## ◆ Output:

- Dòng đầu tiên: Tổng lớn nhất của dãy con gồm  $K$  phần tử liên tiếp.
- Dòng thứ hai: Dãy con đầu tiên có tổng lớn nhất, gồm  $K$  số nguyên cách nhau bởi dấu cách.

## ◆ Ràng buộc:

- $1 \leq K \leq N \leq 10^7$
- $1 \leq A[i] \leq 10^9$

## ◆ Ý tưởng giải:

Sử dụng **Sliding Window (cửa sổ trượt)**:

- Tính tổng của  $K$  phần tử đầu tiên.
- Duyệt từ vị trí  $K$  đến  $N - 1$ , mỗi bước:
  - Trừ phần tử ở đầu cửa sổ, cộng phần tử tiếp theo ở cuối cửa sổ vào tổng hiện tại.
  - Nếu tổng mới lớn hơn tổng lớn nhất đang có, cập nhật lại tổng lớn nhất và chỉ số bắt đầu của cửa sổ.
- Cuối cùng, in ra tổng lớn nhất và dãy con bắt đầu từ chỉ số đã lưu.

## ◆ Độ phức tạp:

- Thời gian:**  $O(N)$
- Bộ nhớ:**  $O(1)$  (nếu không tính mảng đầu vào)

## ◆ Sample Input 0:

```
59 4
72 39 5 74 29 83 63 67 69 10 50 86 32 66 96 100 47 32 41 96 9 7 58
97 87 26 21 23 100 52 31 76 43 74 17 72 104 40 99 8 16 27 38 78 69
46 16 58 73 52 49 67 62 42 64 28 46 34 83
```

## ◆ Sample Output 0:

315

72 104 40 99