Perfect Pair

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 64M

Ở ngôi làng Cổ Tích, cậu bé Bin được giao nhiệm vụ ghép các **đôi găng tay ma thuật**. Trên mỗi chiếc găng tay có ghi một số nguyên dương. Hai chiếc găng tay chỉ có thể ghép thành một **cặp hoàn hảo** nếu tổng hai số ghi trên chúng đúng bằng con số bí ẩn **K**.

Bin có một xấp găng tay gồm **N** chiếc. Cậu muốn biết có bao nhiêu cách chọn ra **hai chiếc găng tay khác nhau** (phân biệt vị trí) tạo thành một cặp hoàn hảo. Bạn hãy giúp Bin đếm số cặp nhé!

Yêu cầu

Cho dãy gồm **N** số nguyên dương $a_1, a_2, \ldots, a_N$. Hãy đếm số cặp chỉ số (i, j) với i < j sao cho $a_i + a_j = K$.

Dữ liệu vào

Đọc từ file **PAIR.INP**:

- Dòng thứ nhất chứa hai số nguyên **N** và **K** ($1 \leq N \leq 10^5; 1 \leq K \leq 2 \times 10^9$).
- Dòng thứ hai chứa **N** số nguyên dương $a_1, a_2, \ldots, a_N$ ($1 \leq a_i \leq 10^9$), mỗi số cách nhau bởi dấu cách.

Dữ liệu ra

Ghi ra file **PAIR.OUT**:

- Một số nguyên duy nhất là số cặp (i, j) thỏa mãn.

Ví dụ

Ví dụ 1:

Input:

```
5 9
4 1 3 6 5
```

Output:

```
2
```

Giải thích: Các cặp (i, j) có tổng bằng 9 là $(4, 5)$ và $(3, 6)$.

Ví dụ 2:

Input:

```
4 7
2 2 3 4
```

Output:

```
1
```

Subtask

- **Subtask 1 (40% số điểm):** $1 \leq N \leq 10^3$.
- **Subtask 2 (60% số điểm):** $1 \leq N \leq 10^5$.

Planting Trees

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 64M

Vào mùa thu hoạch, bác Tư dự định trồng **M** cây giống trên một khu vườn thẳng dài. Trên khu vườn có **N** vị trí có thể trồng, vị trí thứ i nằm ở tọa độ x_i. Hai cây trồng cạnh nhau phải cách nhau ít nhất một khoảng cách nhất định để không che bóng lẫn nhau.

Bác Tư muốn **khoảng cách nhỏ nhất** giữa hai cây liên tiếp trong số các cây được trồng là **lớn nhất có thể** để khu vườn thoáng đãng. Hãy giúp bác Tư tìm khoảng cách đó.

Yêu cầu

Cho **N** vị trí x_1**, x_2, ...,** x_N trên trục số (các tọa độ đôi một khác nhau, chưa sắp xếp). Cần chọn đúng **M** vị trí ($M \leq N$) để trồng cây sao cho **khoảng cách nhỏ nhất** giữa hai vị trí được chọn liên tiếp (theo thứ tự tọa độ) là lớn nhất có thể.

Dữ liệu vào

Đọc từ file **TREE.INP**:

- Dòng thứ nhất chứa hai số nguyên **N** và **M** ($2 \leq M \leq N \leq 10^5$).
- Dòng thứ hai chứa **N** số nguyên x_1**, x_2, ..., x_N** ($-10^9 \leq x_i \leq 10^9$, các giá trị đôi một khác nhau), mỗi số cách nhau bởi dấu cách.

Dữ liệu ra

Ghi ra file **TREE.OUT**:

- Một số nguyên duy nhất là khoảng cách nhỏ nhất lớn nhất có thể tìm được.

Ví dụ

Ví dụ 1:

Input:

```
5 3
1 2 8 4 9
```

Output:

```
3
```

Giải thích: Chọn các vị trí 1, 4, 8 (hoặc 1, 4, 9) với khoảng cách nhỏ nhất là 3.

Ví dụ 2:

Input:

```
6 4
0 1 5 8 10 15
```

Output:

Giải thích: Chọn 0, 5, 8, 15 (hoặc 1, 5, 8, 15) có khoảng cách nhỏ nhất là 3.

Subtask

- **Subtask 1 (40% số điểm):** $2 \leq N \leq 10^3$.
- **Subtask 2 (60% số điểm):** $2 \leq N \leq 10^5$.

Magic Staircase

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 64M

Có một chiếc cầu thang thần kỳ với **N** bậc. Mỗi lần bước, người ta có thể đi lên **một bậc** hoặc **hai bậc**. Câu đố đặt ra là: có bao nhiêu cách khác nhau để đi từ dưới đất lên tới bậc thứ **N**?

Bin muốn thử tất cả các cách đi khác nhau để tìm ra cách "đẹp nhất", nhưng số cách là quá lớn. Vì vậy Bin chỉ cần biết **số cách** (chia lấy dư cho $10^9 + 7$) để lên đến bậc thứ **N**.

Yêu cầu

Cho số nguyên dương **N**. Hãy tính số cách đi từ bậc 0 lên bậc N, biết mỗi bước đi được 1 hoặc 2 bậc. In kết quả chia lấy dư cho $10^9 + 7$.

Dữ liệu vào

Đọc từ file **STAIR.INP**:

- Gồm một dòng duy nhất chứa số nguyên **N** ($1 \leq N \leq 10^6$).

Dữ liệu ra

Ghi ra file **STAIR.OUT**:

- Một số nguyên duy nhất là số cách lên tới bậc N, lấy modulo $10^9 + 7$.

Ví dụ

Ví dụ 1:

Input:

1

Output:

1

Giải thích: Chỉ có 1 cách: bước 1 bậc.

Ví dụ 2:

Input:

2

Output:

2

Giải thích: Cách 1: 1 + 1; cách 2: bước 2 bậc một lần.

Ví dụ 3:

Input:

3

Output:

3

Giải thích: 1+1+1; 1+2; 2+1.

Subtask

- **Subtask 1 (40% số điểm):** $1 \leq N \leq 30$.
- **Subtask 2 (60% số điểm):** $1 \leq N \leq 10^6$.

Treasure Subarray

Giới hạn thời gian: 1.0s **Giới hạn bộ nhớ:** 64M

Thám tử An đang truy tìm một **kho báu bị nguyền rủa**. Trên bản đồ kho báu có một dãy gồm **N** ô vuông, mỗi ô ghi một số nguyên (có thể âm). Truyền thuyết kể rằng: tổng giá trị của một **đoạn ô liên tiếp** đúng bằng con số **K** thì phía dưới đoạn ô đó có chứa một phần của kho báu.

An muốn biết có bao nhiêu đoạn ô liên tiếp (khác nhau về vị trí bắt đầu hoặc kết thúc) có tổng đúng bằng **K**, để lập kế hoạch đào bới toàn diện.

Yêu cầu

Cho dãy **N** số nguyên $a_1, a_2, \ldots, a_N$. Hãy đếm số cặp (l, r) với $1 \le l \le r \le N$ sao cho tổng $a_l + a_{l+1} + \cdots + a_r = K$.

Dữ liệu vào

Đọc từ file **TREASURE.INP**:

- Dòng thứ nhất chứa hai số nguyên **N** và **K** ($1 \le N \le 2 \times 10^5$; $-10^9 \le K \le 10^9$).
- Dòng thứ hai chứa **N** số nguyên $a_1, a_2, \ldots, a_N$ ($-10^9 \le a_i \le 10^9$), mỗi số cách nhau bởi dấu cách.

Dữ liệu ra

Ghi ra file **TREASURE.OUT**:

- Một số nguyên duy nhất là số đoạn con có tổng bằng K.

Ví dụ

Ví dụ 1:

Input:

```
5 2
1 1 1 1 1
```

Output:

```
4
```

Giải thích: Các đoạn [1,2], [2,3], [3,4], [4,5].

Ví dụ 2:

Input:

```
4 0
1 -1 1 -1
```

Output:

```
4
```

Giải thích: Các đoạn [1,2], [2,3], [3,4], [1,4].

Subtask

- **Subtask 1 (40% số điểm):** $1 \leq N \leq 10^3$, các số đều không âm.
- **Subtask 2 (60% số điểm):** $1 \leq N \leq 2 \times 10^5$, có thể có số âm.