**Books**

**Link submit:** <http://codeforces.com/problemset/problem/279/B>

**Solution:**

|  |  |
| --- | --- |
| C++ | <https://ideone.com/gN1X1Y> |
| Java | <https://ideone.com/bj8X6T> |
| Python | <https://ideone.com/dAJpR1> |

**Tóm tắt đề:**

Valera đang ở thư viện, cậu có t phút để đọc các quyển sách mà mình muốn. Trên kệ sách của thư viện hiện tại có n quyển sách. Nếu Valera chọn đọc bắt đầu từ quyển sách thứ i thì sau đó, cậu chỉ có thể đọc tiếp các quyển sách kề sau i, tức quyển thứ i + 1, i + 2, … trở về sau.

Nhiệm vụ của bạn là hãy giúp Valera tính số lượng sách tối đa mà cậu có thể đọc trong khoảng thời gian t. Lưu ý rằng nếu thời gian cần đọc một quyển quá thời gian đọc sách còn lại, Valera sẽ không đọc quyển sách ấy.

**Input:**

Dòng đầu tiên chứa hai số nguyên *n* (1 ≤ n ≤ 105) và *t* (1 ≤ t ≤ 109). Trong đó, n là số quyển sách hiện có và t là thời gian tối đa Valera có thể ở thư viện.

Dòng thứ hai gồm n số nguyên *ai* (1 ≤ ai ≤ 104) là thời gian để đọc xong quyển sách thứ i.

**Output:**

In ra một số nguyên duy nhất là số quyển sách tối đa Valera có thể đọc trong thời gian cho phép.

**Ví dụ:**

|  |  |
| --- | --- |
| 4 5 3 1 2 1 | 3 |

|  |  |
| --- | --- |
| 3 3 2 2 3 | 1 |

**Giải thích ví dụ:**

***Ví dụ 1:*** Valera có thời gian 5 phút ở thư viện. Cậu sẽ đọc được tối đa là 3 quyển sách nếu bắt đầu đọc từ quyển sách thứ 2 trở về sau, tức [1, 2, 1].

***Ví dụ 2:*** Valera có thời gian 3 phút ở thư viện. Cậu chỉ có thể đọc được tối đa là 1 quyển dù bắt đầu đọc tại vị trí nào trên kệ sách.

**Hướng dẫn giải:**

Nhận thấy rằng nếu ta có thể duyệt qua từng đoạn liên tiếp có tổng thời gian còn nằm trong t thời gian cho phép thì việc còn lại trở nên vô cùng đơn giản: chọn ra đoạn có chiều dài lớn nhất, tương ứng với số lượng sách đọc được là nhiều nhất.

Mỗi đoạn thỏa tính chất như trên đều có thể được biểu diễn dưới dạng một đoạn dài nhất có tổng thời không lớn hơn t tính từ quyển sách thứ i trở về trước.

Lúc này, ta có thể áp dụng kỹ thuật Two Pointers với hai biến chạy i và j như sau:

* Bước 1: Đưa toàn bộ thời gian cần để đọc các quyển sách hiện có vào mảng.
* Bước 2: Duyệt qua lần lượt từng quyển sách thứ i trong mảng, kiểm tra xem thời gian còn lại có đủ để ta đọc hết quyển sách thứ i này không:
  + Nếu không, đồng nghĩa với việc đoạn ta đang chọn đọc bắt đầu từ quyển sách thứ j đã tới giới hạn (đã là đoạn lớn nhất có tổng thời gian không quá t tính từ quyển i - 1 trở về trước). Ta di chuyển đến đoạn tiếp theo, tức đoạn bắt đầu tại vị trí (j + 1) và xét tiếp cho đến khi nào ta nhận được một vị trí bắt đầu mới mà tại đó ta vẫn có thể đọc được quyển sách thứ i đang xét.
  + Ngược lại, nghĩa là ta còn có thể đọc thêm quyển sách thứ i này vào đoạn hiện tại. Ta tiến hành cập nhật lại các thông số tương ứng về thời gian đọc sách còn lại và số lượng sách đã đọc được trong đoạn.
  + So sánh số sách hiện tại đọc được với số sách tối đa hiện có và cập nhật nếu cần.
* Bước 3: In ra kết quả.

**Độ phức tạp:** **O(n)** với n là số lượng quyển sách.