

Codeforces Round #171

Problema B: *Books*

Prof. Edson Alves – UnB/FGA

Problema

When Valera has got some free time, he goes to the library to read some books. Today he's got t free minutes to read. That's why Valera took n books in the library and for each book he estimated the time he is going to need to read it. Let's number the books by integers from 1 to n . Valera needs a_i minutes to read the i -th book.

Valera decided to choose an arbitrary book with number i and read the books one by one, starting from this book. In other words, he will first read book number i , then book number $i + 1$, then book number $i + 2$ and so on. He continues the process until he either runs out of the free time or finishes reading the n -th book. Valera reads each book up to the end, that is, he doesn't start reading the book if he doesn't have enough free time to finish reading it.

Print the maximum number of books Valera can read.

Input

The first line contains two integers n and t ($1 \leq n \leq 10^5$; $1 \leq t \leq 10^9$) – the number of books and the number of free minutes Valera's got. The second line contains a sequence of n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$), where number a_i shows the number of minutes that the boy needs to read the i -th book.

Output

Print a single integer – the maximum number of books Valera can read.

Exemplo de entradas e saídas

Sample Input

4 5
3 1 2 1

3 3
2 2 3

Sample Output

3

1

Solução com complexidade $O(N)$

- O problema consiste em avaliar, para cada $i \in [1, N]$, o maior número de livros que podem ser lidos começando no i -ésimo livro
- Uma solução de busca completa avaliaria cada i em $O(N)$, tendo portanto complexidade $O(N^2)$
- O uso de dois ponteiros reduz esta complexidade para $O(N)$
- O ponteiro L inicia no primeiro elemento, e deve ser incrementado sequencialmente até o último elemento, representando aqui o i do problema
- Já o ponteiro R também inicia no primeiro elemento, e a cada iteração ele será o maior dentre L e R , pois cada livro será avaliado uma única vez por R

Solução com complexidade $O(N)$

- Deve ser mantida uma variável t , inicialmente igual a T , que mantém o registro do tempo ainda disponível para leitura
- A cada iteração, enquanto R apontar para um elemento do vetor e houver tempo para ler o R -ésimo livro, R deve ser incrementado e t atualizado
- Ao fim de cada iteração o $R - L$ livros do intervalo $[L, R)$ podem ser lidos
- A variável t deve ser atualizada, acrescentado o tempo de leitura do L -ésimo livro, caso tenha sido decrementado de t previamente
- A resposta será o tamanho do maior dentre estes intervalos

Solução AC com complexidade $O(N)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int solve(int N, int T, const vector<int>& xs) {
6     auto ans = 0, L = 0, R = 0, t = T;
7
8     while (L < N) {
9         R = max(L, R);
10
11         while (R < N and xs[R] <= t)
12             t -= xs[R++];
13
14         ans = max(ans, R - L);
15         t = min(T, t + xs[L]);
16         ++L;
17     }
18
19     return ans;
20 }
```

Solução AC com complexidade $O(N)$

```
22 int main()
23 {
24     ios::sync_with_stdio(false);
25
26     int N, T;
27     cin >> N >> T;
28
29     vector<int> xs(N);
30
31     for (int i = 0; i < N; ++i)
32         cin >> xs[i];
33
34     auto ans = solve(N, T, xs);
35
36     cout << ans << '\n';
37
38     return 0;
39 }
```