

Sherlock Holmes i włamanie stulecia

Pusty skarbiec Narodowego Banku Bajtociji robił niezwykle przygnębiające wrażenie. Sherlock Holmes dłuższą chwilę badał podłogę i ściany.

– Ślady dowodzą, że było tu dwóch rabusiów. Jeśli posłużą się panowie węchem, z pewnością poczują lekki zapach oranżady, co wskazuje na Szalonego Anatola... a tak skomplikowany zamek mógł otworzyć tylko Nienormalny Bajtazar.

– Ale w jaki sposób dwóch przestępcołów o tak silnych charakterach zdoałoby się dogadać i podzielić zdobycz? – wyraził wątpliwość główny inspektor bajtockiej policji.

– Elementarne. Ze stosu złota ściągali na przemian pewną liczbę sztabek. Przypuszczam, że miały różną wagę, więc umówili się, że za jednym ruchem każdy może ściągnąć tyle ile uniesie, czyli co najwyżej M kilogramów złota. Sądzę, że rozpoczęła Anatol.

– I co teraz? – z nadzieją w głosie zapytał inspektor.

– Mam dla panów dobrą wiadomość. – oznajmił Holmes. – Na trawniku przed bankiem leży pusta butelka po oranżadzie, a to znaczy, że Anatol będzie potrzebował nowego zapasu, i to szybko. Jeśli się panowie pospieszą, powinni go zastać w najbliższej “Biedronce”. Watsonie, czy potrafisz powiedzieć, ile złota będzie miał przy sobie?

Zakładając, że Anatol i Bajtazar istotnie postąpili tak, jak mówi Holmes, i wiedząc, jakie sztabki leżały kolejno na stosie oblicz, ile złota wziął Anatol. Obaj przestępcy grali optymalnie, starając się zabrać jak najwięcej.

Wejście i wyjście

Ze względu na duży rozmiar wejścia, jest ono podane w formie binarnej. Do jego obsługi zalecamy użyć następujący kod:

```
#include <iostream>
#include <vector>

int solve(int n, int M, std::vector<int> weights) {
    // Tutaj uzupełnij kod swojego rozwiązania.
    // Wartość n to liczba sztabek, a wartość M to maksymalny udźwig osoby.
    // Ciężary sztabek wylistowane są w kolejności od góry do dołu do stosu.
    // Funkcja powinna obliczyć ile złota wziął Anatol.
}

int main() {
    int t, n, M;
    fread(&t, sizeof(int), 1, stdin);
    while (t--) {
        fread(&n, sizeof(int), 1, stdin);
        fread(&M, sizeof(int), 1, stdin);
        std::vector<int> input(n);
        fread(input.data(), sizeof(int)*n, 1, stdin);
        printf("%d\n", solve(n, M, std::move(input)));
    }
}
```

Ograniczenia

- $1 \leq n \leq 10^6$
- $1 \leq M \leq 10^9$
- $1 \leq a_i \leq \min(M, 1000)$

Przykład

Wywołanie `solve(5, 3, {1,2,1,2,1})` powinno zwrócić 4. Plik wejściowy dla przykładu można pobrać tutaj.