

Условие

Гиперпереход, открытый ещё в начале XXI-го века, и сейчас остаётся основным способом перемещения на расстояния до сотен тысяч парсеков. Но совсем недавно физиками открыто новое явление. Оказывается, длительностью альфа-фазы перехода можно легко управлять. Корабль, находящийся в альфа-фазе перехода, накапливает гравитационный потенциал. Чем больше накопленный гравитационный потенциал корабля, тем меньше энергии потребуется ему на прыжок сквозь пространство. Ваша цель — написать программу, которая позволит кораблю за счёт выбора времени начала альфа-фазы и её длительности накопить максимальный гравитационный потенциал. В самой грубой модели грави-интенсивность — это последовательность целых чисел π . Будем считать, что если альфа-фаза началась в момент i и закончилась в момент j , то накопленный в течение альфа-фазы потенциал — это сумма всех чисел, стоящих в последовательности на местах от i до j .

Исходные данные

В первой строке записано целое число N — длина последовательности, отвечающей за грави-интенсивность ($0 \leq N \leq 60000$). Далее идут N строк, в каждой записано целое число p_i ($-30000 \leq p_i \leq 30000$). Результат Выведите максимальный гравитационный потенциал, который может накопить корабль в альфа-фазе прыжка. Считается, что потенциал корабля в начальный момент времени равен нулю.

Примеры

Исходные данные	Результат
10 31 -41 59 26 -53 58 97 -93 -23 84	187
3 -1 -5 -6	0

Успешная попытка

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
10891375	18:03:38 23 фев 2025	Aleksai Smirnov	1296	Clang++ 17 x64	Accepted		0.109	868 КБ

Решение

На языке C++:

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

int kadane(std::vector<int> arr) {
    int current_sum = 0; int best_sum = 0;
    for (int x : arr) {
        current_sum = std::max(x, current_sum + x);
        best_sum = std::max(current_sum, best_sum);
    }
    return best_sum;
}

int main() {
    int x, n; std::cin >> n;
    std::vector<int> arr;
    while (n--) {
        std::cin >> x;
        arr.push_back(x);
    }

    std::cout << kadane(arr) << "\n";
}
```

Про решение

Задача по поиску максимальной суммы подпоследовательности. Использовал Алгоритм Кадана со сложностью $O(n)$.