|  |  |
| --- | --- |
| **Người đóng góp** | Hà Minh Ngọc, Nguyễn Mạnh Phúc |
| **Tên đề bài** | Minimize Good Delivery |
| **Nguồn** | Click here to enter source\_if\_any. |
| **Giới hạn thời gian**  **(Thời gian chạy từng test)** | 1s |
| **Lưu ý** | * Đề bài dạng case test (Từng test case riêng biệt). * Giới hạn thời gian cho từng bộ test. * Tạo folder dạng [<Tên tác giả>][<Tên đề bài cho mỗi bài>] * Trong folder mỗi bài gồm 3 thành phân: File mô tả (File này), source code, file Input và Output cho từng test. * Riêng file input và output đặt tên lần lượt như sau: * [num].in * [num].out   ( VD: 0.in và 0.out  1.in và 1.in) |

|  |
| --- |
| **ĐỀ BÀI** |
| **Lưu ý:** Giới hạn dữ liệu input cụ thể. |
| "giaohangtocdo.vn" is a start-up company. Every day, the company delivers many kinds of products for n retail stores, called 1, 2...n. All stores are located on a long street: store i at position pi on that street.  For simplicity, we assume that all kinds of products have the same size, same weight, and store i needs a fixed number ni products each day.  The company uses many trucks to deliver products. Each truck can bring maximum k products. A truck can also deliver products for some stores in a trip. The company plans to build a grand-warehouse where all trucks pick up products and start a trip to the stores. They want to choose a location on the street to build the grand-warehouse to optimize the total distance of all trips in a day.  **Input**  + The first line contains integer numbers n (1 < n < 10^5) and k (0 < n < 10^9)  + There are n lines after that, where line i contains data for store i, that are pi (0 < pi < 10^9) and ni (0 < ni < 10^9) respectively  **Output**  Print the total distance of all trips (one-way delivery) in a day  **Example**  **Input**  6 5  1 7  2 2  3 6  8 9  10 11  15 13  **Output**  44 |

|  |
| --- |
| **ĐÁP ÁN (HƯỚNG DẪN GIẢI) <nếu có>** |
| **Lưu ý:** Đáp án có thể dạng code C/C++ hoặc Java hoặc hướng dẫn cách giải. |
| Nhận xét:   1. Vị trí Warehouse trong trường hợp tổng quát sẽ trung với vị trí của Stores. 2. Ta có thể tính được quãng đường từ Stores đến Warehouse bằng giải thuật tham lam. Xe ưu tiên phân phối hàng cho Store xa nhất trước, nếu xe còn chỗ trống thì ưu tiên phân phố hàng cho stores kế cận. Thời gian chạy O(n) 3. Ta kiểm tra Warehouse ở vị trí trùng với Store cuối cùng (hoặc đầu tiên), tính được được **tổng quãng đường** và **tổng số hàng hóa** của các Stores đứng trước Warehouse, và tương tứng của các Stores đứng sau Warehouse. Thời gian chạy O(n) 4. Ta dịch chuyển Warehouse đi một vị trí, ta tính lại được 4 giá trị trên trong thời gian O(0). 5. Ta tìm được giá trị nhỏ nhất trong thời gian O(n) |

|  |
| --- |
| **GHI CHÚ** |
| Click here to enter note. |