

1. Ô tô bay [FLYCAR.*]

Hãng xe ô tô BT đang thử nghiệm một loại ô tô bay. Mỗi khi gặp một chướng ngại vật có độ cao h , ô tô có thể đi qua chướng ngại vật này bằng cách "nâng" độ cao của mình cách mặt đất một khoảng $l \geq h$. Tất nhiên "nâng" độ cao càng lớn thì nhiên liệu sử dụng càng nhiều. Do đó BT định nghĩa "độ lãng phí" khi nâng ô tô lên chiều cao l để đi qua chướng ngại vật chiều cao h là $l - h$. Trong ngày thử nghiệm loại ô tô mới này, BT cho ô tô đi qua n chướng ngại vật theo thứ tự có chiều cao là h_1, h_2, \dots, h_n . Tất nhiên khi đi qua chướng ngại vật nào ô tô phải duy trì chiều cao tối thiểu bằng chướng ngại vật đó. Do đang là phiên bản thử nghiệm nên trong suốt quá trình đi qua n chướng ngại vật ô tô chỉ có thể thay đổi độ cao không quá k lần.

Yêu cầu: Viết chương trình lên lịch thay đổi độ cao của ô tô sao cho tổng "độ lãng phí" khi đi qua n chướng ngại vật là nhỏ nhất? Ô tô có thể khởi hành với độ cao ban đầu bất kỳ và việc xuất phát đưa ô tô lên độ cao ban đầu này không được tính vào k lần thay đổi độ cao.

Input:

- Dòng đầu tiên chứa hai số nguyên dương n, k ($1 \leq k < n \leq 400$)
- Dòng thứ hai chứa n số nguyên h_1, h_2, \dots, h_n ($0 \leq h_i \leq 10^9$) là độ cao của các chướng ngại vật lần lượt xuất hiện trên hành trình.

Output: In ra một số nguyên là "tổng độ lãng phí" nhỏ nhất khi thay đổi độ cao của ô tô một cách hợp lý.

Example:

Input	Output
6 2 7 9 8 2 3 2	3

Giải thích: Ô tô xuất phát với độ cao 7. Sau khi vượt qua chướng ngại vật thứ nhất nó tăng độ cao lên 9, giữ nguyên độ cao này cho đến khi vượt qua chướng ngại vật thứ ba thì giảm xuống độ cao 3 bay cho đến khi vượt qua chướng ngại vật thứ sáu. Tổng "độ lãng phí" là:

$$(7-7)+(9-9)+(9-8)+(3-2)+(3-3)+(3-2)=3$$

2. Tô màu [FLOODFILL.*]

Trên một đường thẳng cho n hình vuông xếp thành một hàng đánh số $1, 2, \dots, n$ từ trái qua phải. Hình vuông thứ i có màu là c_i .

Ta nói rằng một dãy hình vuông từ vị trí i đến vị trí j là một mảng màu nếu như $c_i = c_j$ và $c_i = c_k$ với mọi $i < k < j$. Nói cách khác tất cả các hình vuông từ vị trí i đến vị trí j là cùng một màu. Mỗi thao tác bạn có thể tô lại màu cho các hình vuông trong một mảng màu thành màu mới.

Hỏi rằng số thao tác ít nhất cần thực hiện để đưa tất cả n hình vuông về cùng một màu là bao nhiêu

Input:

- Dòng đầu tiên ghi số nguyên dương n ($n \leq 5000$) - số hình vuông
- Dòng thứ hai ghi n số nguyên dương c_1, c_2, \dots, c_n ($1 \leq c_i \leq 5000$) - màu ban đầu của các hình vuông.

Output: In ra một số nguyên là số thao tác nhỏ nhất cần thực hiện

Example:

Input	Output
4 5 2 2 1	2
8 4 5 2 2 1 3 5 5	4

Giải thích: Trong ví dụ thứ nhất biến đổi:

$[5, 2, 2, 1] \rightarrow [5, 5, 5, 1] \rightarrow [1, 1, 1, 1]$

3. Rút gọn đoạn [CUTSEQ.*]

Cho một dãy gồm N chữ số thuộc đoạn $0..9$ ($N \leq 200$). Ở mỗi bước, ta có thể lấy ra từ dãy này một đoạn liên tiếp các chữ số giống nhau và nhận được một số tiền bằng bình phương độ dài của đoạn được lấy ra. Nếu sau khi lấy, dãy đã cho bị tách làm 2 dãy con, 2 dãy con này lập tức được sát nhập lại thành 1 (giữ nguyên thứ tự).

Hãy tính số lượng tiền lớn nhất có thể thu được.

Input:

Dòng đầu ghi số N . Dòng thứ hai ghi N chữ số thể hiện dãy.

Output:

Ghi ra số lượng tiền lớn nhất có thể thu được

Example:

Input	Output
6 100011	18

4. Chia kẹo [CANDY.*]

Hùng và Dũng là hai anh em, Hùng là anh và Dũng là em. Nhân ngày Quốc tế Thiếu nhi, mẹ mua n gói kẹo cho hai anh em. Vấn đề hóc búa là chia các gói kẹo này. Để kiểm tra độ thông minh của hai con, mẹ đưa ra yêu cầu chia theo qui tắc: Hai anh em chọn một số gói kẹo sao cho có thể chia thành hai phần với số lượng kẹo trong mỗi phần bằng nhau và tổng số kẹo trong các gói không được chia là nhỏ nhất có thể. Số kẹo trong các gói không được chia mẹ sẽ dành cho Dũng (em - bé hơn!!!).

Hỏi rằng tổng số kẹo mà Dũng nhận được là bao nhiêu?

Input:

- Dòng đầu tiên ghi số nguyên dương n ($n \leq 500$) số gói kẹo mà mẹ mua
- n dòng tiếp theo, dòng thứ i ghi c_i là số kẹo có trong gói thứ i . Tổng số kẹo trong n gói kẹo không vượt quá 100000.

Output: Số kẹo mà Dũng nhận được.

Example:

Input	Output
5 2 3 5 8 13	18

Subtasks:

- Subtask 1: $n \leq 13$ [50%]
- Subtask 2: $n \leq 50$, tổng số kẹo ≤ 1000 [70%]