

Công việc

Cho n công việc, một số công việc chỉ được thực hiện khi một số công việc khác đã được thực hiện xong. Khi thực hiện, một công việc phải làm trong một ngày, mỗi ngày không được làm quá m công việc.

Yêu cầu: Tính số ngày ít nhất để thực hiện xong n công việc

Input

- Dòng đầu chứa 2 số n, m ;
- n dòng sau, dòng thứ i có dạng:
 - o Số đầu tiên của dòng là k_i - số công việc phải thực hiện trước ngày công việc i được thực hiện;
 - o tiếp theo là k_i số là chỉ số các công việc phải được thực hiện trước công việc i .

Output

- Gồm một số duy nhất là số ngày ít nhất để thực hiện hết n công việc, nếu không có phương án nào thực hiện được hết n công việc thì ghi -1 .

Input	Output
3 2 2 2 3 0 0	2

Subtask 1: $n \leq m \leq 100$;

Subtask 2: $m \leq n \leq 15$.

Phân chia hàng cứu trợ (Distribution.*)

Một tổ chức từ thiện có n kiện hàng cứu trợ, các kiện có giá trị tương ứng là a_1, a_2, \dots, a_n . Có ba vùng bị ảnh hưởng bởi thiên tai, tổ chức đang lên kế hoạch phân chia các kiện hàng thành ba phần (không được chia nhỏ các kiện hàng), mỗi phần chuyển tới một vùng. Gọi $A \geq B \geq C$ là tổng giá trị hàng cứu trợ của ba vùng tương ứng, tổ chức mong muốn chênh lệch ($A - C$) là nhỏ nhất.

Dữ liệu:

- Dòng đầu chứa số nguyên n ;
- Dòng thứ hai chứa n số nguyên dương a_1, a_2, \dots, a_n ($a_i \leq 10^9$).

Kết quả:

- Ghi một dòng chứa một số là giá trị ($A - C$) nhỏ nhất tìm được.

Ràng buộc:

- Subtask 1 (90%): $n \leq 20$;
- Subtask 2 (10%): $n \leq 25$.

Ví dụ:

Dữ liệu vào	Kết quả ra	Giải thích
5 2 3 4 5 6	1	Một phương án phân chia $A=2+5$, $B=3+4$, $C=6$, để chênh lệch $A-C$ nhỏ nhất bằng 1.

oliu23split

Cho một đồ thị có dạng cây gồm n đỉnh, đỉnh thứ i ($1 \leq i \leq n$) có trọng số w_i . Với một số nguyên k , Oliu muốn chia cây thành k vùng liên thông để phần có tổng trọng số nhỏ nhất là lớn nhất.

Input

- Dòng đầu chứa hai số nguyên n, k ($1 \leq k \leq n \leq 10^6$);
- Dòng thứ hai chứa n số nguyên dương w_1, w_2, \dots, w_n ($w_i \leq 10^6$);
- Tiếp theo là $n - 1$ dòng, mỗi dòng chứa hai số u, v mô tả một cạnh của cây.

Output

- Gồm một dòng chứa một số là tổng trọng số nhỏ nhất là lớn nhất.

Input	Output
4 2 1 2 3 4 1 2 2 3 3 4	4

Subtask 1: $n \leq 10$;

Subtask 2: $n \leq 10^3$;

Subtask 3: $n \leq 10^5$;

Subtask 4: $n \leq 10^6$;

Chọn điểm

Các nhà khoa học đang nghiên cứu để xây dựng trạm vũ trụ quốc tế theo tiêu chuẩn mới. Ban đầu, họ sẽ lắp đặt nó ở trung tâm hàng không vũ trụ trên mặt đất sau đó dùng tàu vũ trụ để đưa trạm lên quỹ đạo. Họ biểu diễn trung tâm bằng hệ toạ độ đề-các hai chiều, trên đó đánh dấu sẵn 29 điểm (được đánh số từ 0 đến 28) có toạ độ là các số nguyên không âm, và là các vị trí thuận lợi cho việc vận chuyển vật liệu. Tiếp theo, họ cần chọn ra 8 điểm trong số đó để lắp đặt 8 mô-đun của trạm. Để thuận tiện cho việc đưa trạm vũ trụ vào quỹ đạo sau này, 8 điểm được chọn phải có trọng tâm là toạ độ nguyên. Nghĩa là, trung bình cộng hoành độ các điểm này là một số nguyên và trung bình cộng tung độ các điểm này cũng là một số nguyên.

Do sơ suất nên phuơng án chọn 8 điểm đã bị mất. Là một kỹ sư tài năng, Bob được giao nhiệm vụ thiết kế và vận hành một máy tính đơn giản để tìm 8 điểm mà trọng tâm của chúng có toạ độ nguyên. Máy có 64 ô nhớ được đánh số từ 0 đến 63, mỗi ô nhớ đều lưu trữ được các số nguyên không âm có 30 bit, giá trị ô nhớ thứ i được kí hiệu là A_i . Các phép toán mà máy có thể thực thi là:

- **add(i, j, k):** Tính $A_i + A_j$ và gán kết quả vào A_k , nếu kết quả là một số nguyên có nhiều hơn 30 bit thì sẽ chỉ được giữ lại 30 bit cuối.
- **sub(i, j, k):** Tính $A_i - A_j$ và gán kết quả vào A_k , nếu kết quả là một số âm thì sẽ được đặt bằng 0.
- **half(i, j):** Tính phần nguyên của $A_i / 2$ và gán kết quả vào A_j .

Lưu ý, các chỉ số i, j, k có thể bằng nhau và $0 \leq i, j, k \leq 63$.

Ngoài ra, máy có một đèn tín hiệu với rất nhiều dây dẫn đầu vào, mỗi dây có thể được cắm vào một ô nhớ. Đèn sẽ sáng khi trong số các ô nhớ được cắm vào, tồn tại hai ô có giá trị bằng nhau.

Hiện tại, toạ độ của 29 điểm đã được nạp vào máy, hoành độ của điểm thứ i được nạp vào ô nhớ $2i$ và tung độ của điểm thứ i được nạp vào ô nhớ $2i + 1$. Các ô nhớ còn lại đều có giá trị bằng 0.

Yêu cầu

Hãy giúp Bob đưa ra các lệnh tính toán, thay đổi các dây dẫn của đèn và quan sát trạng thái đèn để tìm ra được chỉ số của 8 điểm mà ban đầu trọng tâm của chúng có toạ độ nguyên, sao cho số lần quan sát đèn là càng ít càng tốt và số lệnh tính toán sử dụng là không quá 10^5 . Nếu có nhiều cách chọn ra 8 điểm, chỉ cần đưa ra một cách bất kỳ trong số đó.

Tương tác

Dãy là bài toán tương tác không thích ứng. Hệ thống cung cấp thư viện **cpointlib.h** chứa các hàm sau:

- **void add(int i, int j, int k):** Thực hiện phép toán add.
- **void sub(int i, int j, int k):** Thực hiện phép toán sub.
- **void half(int i, int j):** Thực hiện phép toán half.
- **bool ask(string S):** Cắm dây dẫn vào các ô nhớ và quan sát đèn. S là xâu nhị phân độ dài 64, kí tự thứ i là '1' nếu ô nhớ i được cắm vào, ngược lại là '0'.
Hàm trả về true nếu đèn sáng, false nếu không sáng.

Chi tiết cài đặt

Thí sinh cần cài đặt hàm sau trong file **cpoint.cpp**:

```
#include "cpointlib.h"
vector<int> solve() {
    // Code của thí sinh
}
```

Hàm solve phải trả về một vector chứa 8 chỉ số của các điểm được chọn.

Chấm điểm

Gọi Q là số lần quan sát đèn trong phương án của thí sinh. Thí sinh sẽ nhận được:

- 0 điểm nếu $Q > 276$;
- $(0.2 \times 0.95^{(Q-174)}) \times 100\%$ số điểm nếu $174 \leq Q \leq 276$;
- $(0.2 + 0.2 \times 0.95^{(Q-94)}) \times 100\%$ số điểm nếu $94 \leq Q < 174$;
- $(0.4 + 0.5 \times 0.8^{(Q-76)}) \times 100\%$ số điểm nếu $76 \leq Q < 94$;
- $(0.9 + 0.1 \times 0.8^{(Q-66)}) \times 100\%$ số điểm nếu $66 < Q < 76$;
- 100% số điểm nếu $Q \leq 66$.

Ràng buộc

Các hoành độ và tung độ của 29 điểm đều là số nguyên không âm và nhỏ hơn 2^{30} .

Subtask

- **Subtask 1 (20 điểm):** Có ít nhất 8 điểm trong 29 điểm mà cả tung độ và hoành độ đều chia hết cho 8.
- **Subtask 2 (80 điểm):** Không có ràng buộc nào thêm.

Pacman

Trò chơi pacman trên đa đồ thị vô hướng, trọng số, có khuyên như sau: Người chơi phải tìm một hành trình di chuyển pacman, xuất phát tại đỉnh 1, đi qua hết tất cả các cạnh, mỗi cạnh qua ít nhất một lần để thu thập hết các viên ngọc nằm trên mỗi cạnh sau đó quay về vị trí xuất phát 1. Người chơi sẽ đánh giá cao nếu tổng độ trọng trên các cạnh mà pacman phải đi là nhỏ nhất.

Input

- Dòng đầu chứa hai số nguyên dương n, m ($n \leq 20; m \leq ???$);
- Tiếp theo là m dòng, mỗi dòng chứa ba số nguyên dương i, j, c_{ij} ($1 \leq i, j \leq n; c_{ij} \leq 10^6$).

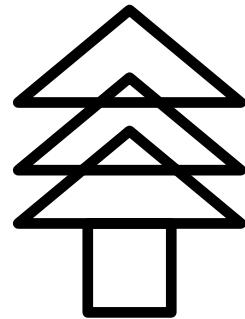
Output

- Gồm một dòng chứa một số nguyên là tổng trọng số nhỏ nhất của hành trình tìm được. Nếu không tồn tại phương án ghi -1.

Dữ liệu vào	Kết quả ra
2 2 1 1 3 1 2 1	5

DRGAME

Trong lý thuyết đồ thị, một đường đi trong đồ thị $G=(V,E)$ được gọi là đường đi Euler nếu nó đi qua tất cả các cạnh của đồ thị, mỗi cạnh đúng một lần. Đường đi Euler có đỉnh cuối cùng trùng với đỉnh xuất phát gọi là chu trình Euler. Khái niệm chu trình Euler xuất phát từ bài toán bảy cây cầu do Euler giải quyết vào khoảng năm 1737. Đường đi Euler có thể tìm thấy trong các bài toán vui vẽ một nét (vẽ một hình nào đó mà không nhắc bút khỏi mặt giấy, không tô lại cạnh nào hai lần). Ví dụ hình bên có thể vẽ bằng một nét.



Xét trò chơi DRGAME liên quan đến vấn đề vẽ một nét như sau: Mỗi lượt chơi, người chơi sẽ được cho một hình vẽ và phải tìm cách vẽ lại hình đó bằng ít lần nhắc bút khỏi mặt giấy nhất và không có cạnh nào được tô lại hai lần.

Yêu cầu: Cho một hình vẽ được mô tả bằng một đồ thị đơn, vô hướng gồm n đỉnh, m cạnh. Hãy tìm cách vẽ lại hình đó bằng ít lần nhắc bút khỏi mặt giấy nhất và không có cạnh nào được tô lại hai lần.

Input

- Dòng đầu gồm hai số nguyên n, m ($n \leq 1000$);
- m dòng tiếp theo mô tả m cạnh của đồ thị, mỗi dòng chứa hai số nguyên i, j mô tả cạnh nối từ đỉnh i đến đỉnh j .

Output

- Dòng đầu chứa số một số nguyên k là số lần nhắc bút khỏi mặt giấy;
- Tiếp theo là k dòng, mỗi dòng theo khuôn dạng:
 - Số đầu tiên là số lượng đỉnh s đi qua trong một lần vẽ;
 - Tiếp theo là s số mô tả các đỉnh lần lượt đi qua.

Dữ liệu vào	Kết quả ra
4 6	2
1 2	6 1 2 3 4 1 3
2 3	2 2 4
3 4	
1 4	
1 3	
2 4	

Chuỗi vòng hạt

Oliu có n đoạn chuỗi hạt, mỗi đoạn chuỗi gồm một số hạt, các hạt có màu sắc khác nhau và được mô tả bằng các ký tự từ ‘0’ đến ‘6’ (xanh, đỏ, tím, vàng, nâu, đen, trắng). Oliu muốn ghép các đoạn chuỗi hạt này thành một chuỗi vòng tròn theo quy tắc: đoạn chuỗi y muốn ghép theo sau đoạn chuỗi x thì màu hạt cuối cùng của x phải giống với màu hạt đầu tiên của chuỗi y . Tuy nhiên, với một đoạn chuỗi hạt khi ghép vào chuỗi vòng tròn có thể lấy theo chiều xuôi hoặc chiều ngược. Sau khi suy nghĩ, Oliu nhận thấy n đoạn chuỗi có thể không ghép hết để thành một vòng tròn được. Do đó, Oliu muốn tìm cách ghép để được chuỗi vòng tròn gồm nhiều đoạn chuỗi nhất.

Yêu cầu: Cho danh sách gồm n đoạn chuỗi, tìm cách chọn nhiều đoạn chuỗi nhất trong n đoạn chuỗi để ghép được thành một vòng tròn mà vòng tròn phải có ít nhất ba đoạn.

Input

- Dòng đầu chứa số nguyên dương n ;
- Dòng thứ hai đến dòng thứ $n + 1$ mô tả n đoạn chuỗi, mỗi dòng là một xâu số có độ dài không vượt quá 10.

Output

- Gồm một dòng chứa một số là số lượng nhiều nhất chọn được để ghép được thành một vòng tròn mà vòng tròn phải có ít nhất ba đoạn.

Input	Output
4 0 12 23 13	3

Subtask 1: $n \leq 20$

Subtask 2: $n \leq ???$

oliucake

Một chiếc bánh hình chữ nhật kích thước $m \times n$ ô. Các hàng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n . Ô nằm trên hàng i ($1 \leq i \leq m$), cột j ($1 \leq j \leq n$) là một mẫu bánh loại socola $a(i, j)$. Biết rằng mỗi loại socola xuất hiện đúng 4 lần.

Oliu muốn cắt chiếc bánh thành các miếng bánh, mỗi miếng bánh có đúng hai mẫu bánh là hai ô kề cạnh. Sau khi cắt thành các miếng bánh, Oliu đã tìm ra một cách chia các miếng bánh thành hai phần mà ở mỗi phần mỗi loại socola xuất hiện đúng hai lần. Bài toán không hề đơn giản, Oliu đó bạn tìm được cách chia thỏa mãn.

Input

- Dòng đầu chứa hai số nguyên m, n ($m, n \leq 1000$);
- Dòng thứ i trong m dòng tiếp theo chứa n số nguyên dương $a(i, 1), a(i, 2), \dots, a(i, n)$.

Output

- Gồm $\frac{m \times n}{4}$ dòng mô tả phần thứ nhất, mỗi dòng chứa bốn số x, y, u, v là hai ô kề nhau mô tả miếng bánh gồm hai mẫu bánh.

Input	Output
2 4 1 2 1 2 2 2 1 1	1 1 2 1 1 4 2 4

d5mst

Xét đồ thị gồm $n + s$ đỉnh, các đỉnh được đánh số từ 1 đến $n + s$. Với đỉnh $n + k$ ($1 \leq k \leq s$) có cạnh nối tới mỗi đỉnh $1 \leq l_k, l_k + 1, \dots, r_k \leq n$ với trọng số c_k .

Yêu cầu: Tìm cây khung nhỏ nhất.

Input

- Dòng đầu chứa hai số nguyên n, s ($n, s \leq 2e5$);
- Dòng thứ k ($1 \leq k \leq s$) trong s dòng sau chứa ba số nguyên dương l_k, r_k, c_k ($1 \leq l_k \leq r_k \leq n; c_k \leq 10^9$).

Output

- Gồm một dòng chứa một số là trọng số cây khung nhỏ nhất tìm được. Trong trường hợp không tồn tại cây khung thì đưa ra -1 .

Input	Output
3 2 1 2 3 2 3 1	8

Subtask 1 (25%): $n \times s \leq 1e7$;

Subtask 2 (75%): Không có ràng buộc nào thêm.

gstree.cpp

Xét đồ thị gồm n đỉnh, các đỉnh được đánh số từ 1 đến n và có nhãn là một số nguyên dương a_i . Gọi $GCD(x, y)$ là ước số chung lớn nhất của hai số x, y . Cạnh giữa hai đỉnh i, j có trọng số $123456 - GCD(a_i, b_j)$.

Yêu cầu: Tìm cây khung nhỏ nhất của đồ thị.

Input

- Dòng đầu chứa số nguyên n ;
- Dòng thứ hai chứa n số nguyên dương a_1, a_2, \dots, a_n ($a_i \leq 10^5$).

Output

- Gồm một số là trọng số của cây khung nhỏ nhất tìm được.

Dữ liệu vào	Kết quả ra
3 10 20 30	246892

Subtask 1: $n \leq 500$;

Subtask 2: $n \leq 50000$;

mst

Cho một đồ thị vô hướng có trọng số G . Biết rằng, ban đầu tất cả các cạnh đều có trọng số bằng 0, có q thao tác thay đổi, mỗi thao tác mô tả bằng 5 số: $i_{min}, i_{max}, j_{min}, j_{max}, w$, nghĩa là các cạnh (i, j) mà $i_{min} \leq i \leq i_{max} < j_{min} \leq j \leq j_{max}$ được thay đổi một lượng w .

Yêu cầu: Tìm trọng số của cây khung nhỏ nhất của G .

Input

- Dòng đầu chứa hai số nguyên n, q ;
- Tiếp theo là q dòng, mỗi dòng chứa 5 số $i_{min}, i_{max}, j_{min}, j_{max}, w$ ($1 \leq i_{min} \leq i_{max} < j_{min} \leq j_{max} \leq n; |w| \leq 10^6$).

Output

- Gồm một dòng chứa một số là trọng số cây khung nhỏ nhất của G .

Input	Output
3 2 1 1 3 3 3 1 2 3 3 -2	-2

Subtask 1: $n \leq 1000; q \leq 10^5$;

Subtask 2: $n \leq 10^5; q \leq 10^5$;

Khớp xâu hoán vị

Alice rất yêu thích các xâu kí tự. Với ba số N, ℓ, r ($1 \leq \ell \leq r \leq N \leq 13$), cô đã tạo ra tập Ω gồm tất cả các xâu độ dài N là hoán vị của N kí tự Latin in thường đầu tiên trong bảng chữ cái, sao cho vị trí của kí tự 'a' trên xâu thuộc đoạn $[\ell, r]$. Bob muốn tạo ra một xâu U chứa tất cả các xâu trong tập Ω và có độ dài càng nhỏ càng tốt. Ở đây, xâu X được gọi là chứa được xâu Y nếu có thể xoá đi một số kí tự của X (hoặc không xoá kí tự nào) và giữ nguyên thứ tự các kí tự còn lại để thu được Y .

Tuy nhiên, Alice đã không cho Bob biết N, ℓ , và r . Thay vào đó, cô sẽ trả lời một số câu hỏi của anh. Mỗi lần hỏi, Bob sẽ đưa cho Alice một xâu S có độ dài không quá 200, chỉ chứa các kí tự Latin in thường và hỏi xem xâu S này chứa được bao nhiêu xâu trong tập Ω . Alice sẽ tính toán và trả lời Bob số dư khi chia kết quả tính được cho 379. Cuối cùng, Bob sẽ trả lời Alice xâu U mà anh tìm được. Bob thể hiện sự thông minh bằng cách sử dụng ít câu hỏi và tìm ra xâu U có độ dài nhỏ nhất có thể.

Yêu cầu: Hãy lập trình giúp Bob tương tác với Alice để đưa ra được xâu U thỏa mãn sao cho giá trị $Q + |U|$ càng nhỏ càng tốt, với Q là số câu hỏi mà Bob đã hỏi và $|U|$ là độ dài của xâu U .

Tương tác

Đây là bài toán tương tác không thích ứng.

Hệ thống cung cấp thư viện **smatchlib.h** chứa các hàm sau:

```
int ask(string S);
```

- **Mục đích:** Hỏi Alice về số xâu trong Ω mà xâu S chứa được (kết quả được chia lấy dư cho 379).
- **Tham số:** S : Xâu cần hỏi (độ dài ≤ 200 , chỉ chứa kí tự Latin in thường).
- **Giá trị trả về:** Số nguyên là số xâu khớp chia dư cho 379.

Thí sinh cần cài đặt hàm:

```
string solve();
```

- **Mục đích:** Trả về xâu U tìm được.
 - **Lưu ý:**
 - ✓ Được phép gọi hàm `ask` để hỏi Alice.
 - ✓ Hàm có thể được gọi nhiều lần (ứng với nhiều test case).
-

Chấm điểm

Gọi:

- $P=Q+|U|$ (với Q là số lần hỏi, $|U|$ là độ dài xâu U).
- J : Giá trị $Q+|U|$ của phương án tối ưu do giám khảo đưa ra.

Thí sinh được điểm như sau:

- 0 điểm, nếu $Q > 10$;
- $(0.4 \times 0.9^{P-J-N-1}) \times 100\%$ số điểm, nếu $N < P - J$;
- $(0.4 + 0.4 \times 0.8^{P-J-4}) \times 100\%$ số điểm, nếu $4 \leq P - J \leq N$;
- $(0.8 + 0.2 \times 0.8^{P-J}) \times 100\%$ số điểm, nếu $0 < P - J < 4$;
- 100% số điểm nếu $P \leq J$.

Subtask	Điểm	Giới hạn N
1	10	$N=4$
2	10	$N \leq 5$
3	10	$N \leq 6$
4	10	$N \leq 7$
5	10	$N \leq 8$
6	10	$N \leq 9$
7	10	$N \leq 10$
8	30	Không giới hạn

D2SET

Trong toán học, **tập hợp** (tiếng Anh: Set) có thể hiểu tổng quát là một sự tụ tập của một số hữu hạn hay vô hạn các đối tượng nào đó. Các đối tượng này được gọi là các phần tử của tập hợp. *Tập hợp* là một khái niệm nền tảng (*fundamental*) và quan trọng của toán học hiện đại. Ngành toán học nghiên cứu về tập hợp là lý thuyết tập hợp.

Trong lý thuyết tập hợp, người ta xem tập hợp là một khái niệm nguyên thủy, không định nghĩa. Nó tồn tại theo các tiên đề được xây dựng một cách chặt chẽ. Khái niệm tập hợp là nền tảng để xây dựng các khái niệm khác như số, hình, hàm số... trong toán học.

Cho một dãy gồm n số nguyên dương đôi một khác nhau a_1, a_2, \dots, a_n , một tập được gọi là tập *D2SET* nếu thỏa mãn tính chất sau: là tập con có lực lượng lớn nhất trong các tập con của tập $\{a_1, a_2, \dots, a_n\}$ và nếu x thuộc tập thì $2x$ sẽ không thuộc tập.

Yêu cầu: Cho a_1, a_2, \dots, a_n , hãy tìm lực lượng của tập *D2SET* và số cách khác nhau để chọn tập *D2SET*.

Input

Dòng đầu gồm 2 số nguyên n và k ($k \leq 10^9$);

Dòng thứ hai gồm n số nguyên dương a_1, a_2, \dots, a_n ($a_i \leq 10^9$).

Output

Gồm một dòng chứa 2 số s, d , trong đó s là lực lượng của tập *D2SET*, d là số cách khác nhau để chọn tập *D2SET* chia dư cho k .

Input	Output
2 100 1 2	1 2

Subtask 1 (30%): $n \leq 20$;

Subtask 2 (40%): $n \leq 10^6$;

Subtask 3 (30%): $n \leq 10^9$; $a_i = i$ (khi đó file dữ liệu vào chỉ gồm một dòng chứa 2 số nguyên n, k).

apzip

Cho xâu s độ dài n , cần xác định vector r có độ dài n , trong đó, giá trị $r[i]$ ($0 \leq i < n$) là số lượng số 1 nhiều nhất có thể lấy được khi chọn một xâu con gồm $(i+1)$ kí tự liên tiếp bất kì trên xâu s . Vector r được chấp nhận nếu mọi $r[i]$ chênh lệch với kết quả đúng không quá 5%. Cụ thể, gọi $a[i]$ là số lượng số 1 nhiều nhất có thể lấy được khi chọn một xâu con gồm $(i+1)$ kí tự liên tiếp bất kì trên xâu s thì $\frac{|a[i]-r[i]|}{a[i]} \leq 5\%$.

Input

- Dòng đầu chứa số n ($n \leq 10^5$);
- Dòng thứ hai gồm một xâu độ dài n .

Output

- Gồm n số là giá trị của vector r .

Dữ liệu vào	Kết quả ra
5 10101	1 1 2 2 3

Nén dữ liệu

Tham gia cuộc thi “Thiết kế thuật toán nén dữ liệu”, các thí sinh được Ban tổ chức yêu cầu xây dựng hàm nén dữ liệu và trả lời truy vấn như sau:

- Hàm nén dữ liệu: Hàm `string zip(string x)` nhận dữ liệu được truyền vào trong x là một xâu nhị phân độ dài n , hàm sẽ nén dữ liệu thành xâu y cũng là một xâu nhị phân, xâu y là xâu mà hàm `zip` trả về;
- Hàm trả lời truy vấn: Hàm `vector<int> answer(int n, string y)` sẽ nhận dữ liệu n là độ dài xâu x và xâu y là xâu nén của xâu x do hàm `zip` trả về. Hàm `answer` trả về một vector r có độ dài n là kết quả của n truy vấn. Giá trị $r[i]$ là kết quả cho truy vấn thứ i ($0 \leq i < n$) là truy vấn tính số lượng số 1 nhiều nhất có thể lấy được khi chọn một xâu con gồm $(i + 1)$ kí tự liên tiếp bất kì trên xâu x . Trả lời của hàm `answer` được chấp nhận nếu mọi $r[i]$ chênh lệch với kết quả đúng không quá 5%. Cụ thể, gọi $s[i]$ là số lượng số 1 nhiều nhất có thể lấy được khi chọn một xâu con gồm $(i + 1)$ kí tự liên tiếp bất kì trên xâu x thì $\frac{|s[i] - r[i]|}{s[i]} \leq 5\%$.

Yêu cầu: Lê là một thí sinh tham gia cuộc thi trên. Hãy giúp Lê thực hiện được yêu cầu của Ban tổ chức sao cho độ dài xâu y càng nhỏ càng tốt.

Tương tác:

- Thí sinh cần cài đặt hàm `string zip(string x)` và hàm `vector<int> answer(int n, string y)` trong file `apzip.cpp`. Trong file `apzip.cpp` thí sinh cần phải khai báo thư viện bằng dòng lệnh `#include "apziplib.h"` ở đầu file. Ngoài ra, thí sinh được phép khai báo thêm thư viện, xây dựng các hàm, sử dụng biến toàn cục khác nếu cần. File `apzip.cpp` sẽ được biên dịch cùng với thư viện `apziplib.h` (xem thêm các file mẫu trong mục đính kèm trên hệ thống để hiểu hơn về cách tương tác với hệ thống, các file này chỉ để thí sinh hiểu cách thức tương tác, không phải dùng để chấm bài).
- Lưu ý rằng, hàm `zip` và `answer` sẽ được gọi ở các tiến trình độc lập. Vì vậy, nếu hàm `zip` có sử dụng và lưu trữ dữ liệu vào các biến toàn cục, các dữ liệu này sẽ không tồn tại khi hàm `answer` được thực thi.

Chấm điểm:

- Có tất cả 5 subtask. Với mỗi subtask, gọi Q là yêu cầu mức độ nén, với mỗi test trong subtask, cách tính % điểm như sau:
 - ✓ Thí sinh sẽ bị 0% điểm nếu xảy ra một trong các trường hợp: chạy sinh lỗi; tương tác sai quy cách; chạy quá thời gian; trả lời của hàm `answer` không được chấp nhận.
 - ✓ Ngược lại, gọi TS là độ dài xâu y mà thí sinh nén từ xâu x , khi đó % điểm của test là:
$$\begin{cases} 100\% & \text{điểm nếu } TS \leq Q, \\ 100\% \times \frac{3 \times Q - TS}{2 \times Q} & \text{điểm nếu } Q < TS \leq 3 \times Q, \\ 0\% & \text{điểm nếu } TS > 3 \times Q. \end{cases}$$

- Với mỗi subtask, gọi e là % điểm nhỏ nhất mà thí sinh đạt được trong các test thuộc subtask, $score$ là điểm của subtask, khi đó, điểm cho subtask được tính bằng $e \times score$.

Subtask 1 (23 điểm): $n = 10^4$ và $Q = \left\lfloor \frac{81 \times n}{100} \right\rfloor$;

Subtask 2 (19 điểm): $n = 10^4$ và $Q = \left\lfloor \frac{27 \times n}{100} \right\rfloor$;

Subtask 3 (13 điểm): $n = 10^5$ và $Q = \left\lfloor \frac{9 \times n}{100} \right\rfloor$;

Subtask 4 (24 điểm): $n = 10^5$ và $Q = \left\lfloor \frac{3 \times n}{100} \right\rfloor$;

Subtask 5 (21 điểm): $n = 10^5$ và $Q = \left\lfloor \frac{n}{100} \right\rfloor$.

Quà tặng

Tết thiếp năm nay, ba bạn An, Bình, Cường lên kế hoạch làm các món quà tặng các bạn thiếp ba miền Bắc, Trung, Nam. Các món quà được làm từ hai loại nguyên liệu: nguyên liệu loại A và nguyên liệu loại B. Nếu một món quà dùng x nguyên liệu loại A và y nguyên liệu loại B (x và y là các số nguyên dương), thì món quà đó có giá trị là $2^{(x-1)}3^{(y-1)}$. Ba bạn dự định làm các món quà với tổng giá trị là n , giả sử a, b, c tương ứng là tổng giá trị các món quà cho ba miền Bắc, Trung, Nam thì a, b, c thỏa mãn các điều kiện sau:

- 1) $a + b + c = n; 0 < a < b < c;$
- 2) Với mỗi miền, các món quà tạo ra không thể so sánh được với nhau, có nghĩa là không tồn tại hai món quà có giá trị $2^{(x-1)}3^{(y-1)}$ và $2^{(u-1)}3^{(v-1)}$ mà đồng thời $0 < x \leq u$ và $0 < y \leq v$;
- 3) $S(a) + S(b) + S(c)$ là lớn nhất, trong đó ký hiệu $S(p)$ là tổng các chữ số của p .

Yêu cầu: Cho số nguyên dương n , hãy tìm cách tạo ra các món quà thỏa mãn điều kiện đề bài.

Dữ liệu: Vào từ thiết bị vào chuẩn: Dòng đầu tiên ghi số nguyên dương K là số lượng bộ dữ liệu. Tiếp đến là K dòng, mỗi dòng tương ứng với một bộ dữ liệu chứa một số nguyên dương n .

Kết quả: Ghi ra thiết bị ra chuẩn gồm K dòng tương ứng với K bộ dữ liệu trong dữ liệu vào, mỗi dòng có dạng: Ba số nguyên dương đầu tiên của dòng là m_1, m_2, m_3 lần lượt là số lượng món quà được tạo ra của ba miền Bắc, Trung, Nam. m_1 số nguyên dương tiếp theo mô tả giá trị các món quà cho miền Bắc; m_2 số nguyên dương tiếp theo mô tả giá trị các món quà cho miền Trung; m_3 số nguyên dương cuối cùng mô tả giá trị các món quà cho miền Nam. Nếu bộ dữ liệu không tồn tại cách tạo ra các món quà thỏa mãn điều kiện đề bài thì trên dòng tương ứng chỉ ghi một số -1.

Subtask 1: Giả thiết là $n \leq 10^3$.

Subtask 2: Giả thiết là $n \leq 10^5$.

Subtask 3: Giả thiết là $n \leq 10^{15}$.

Ví dụ:

Dữ liệu	Kết quả
2	1 1 1 2 3 4
9	1 2 1 6 3 4 8
21	

d5strp2

Xét tất cả các xâu nhị phân độ dài k , xâu thứ k ($1 \leq k \leq 2^k$) theo thứ tự từ điển là t_k có chi phí là $\text{cost}(t_k)$.

Với một xâu s , chi phí xâu s được tính bằng:

$$\text{cost}(s) = \sum_{i=1}^{n-k+1} \text{cost}(s_{i,\dots,i+k-1})$$

Yêu cầu: Hãy dựng xâu nhị phân s độ dài n có tổng chi phí nhỏ nhất.

Input

- Dòng đầu chứa hai số nguyên n, k ($n \leq 2048; k \leq 10$);
- Dòng thứ hai chứa 2^k số nguyên tương ứng là chi phí các xâu theo thứ tự từ điển.

Output

- Xâu s độ dài n có tổng chi phí nhỏ nhất.

Input	Output
3 1 1 2	000

d5light

Một giàn đèn được mô tả bằng một đồ thị dạng cây gồm n đỉnh, trong đó đỉnh thứ i ($1 \leq i \leq n$) có một đèn màu c_i . Khi chọn đỉnh r làm gốc, hai đỉnh u, v được gọi là hai con tương đương nếu số lượng đèn của từng loại màu trong hai cây con giống nhau.

Yêu cầu: Cho đồ thị mô tả giàn đèn và q truy vấn, truy vấn thứ t ($1 \leq k \leq q$) cần đếm số cặp đỉnh mà hai cây con tương ứng là tương đương khi lấy đỉnh r_k làm gốc.

Input

Dòng đầu chứa số nguyên T là số bộ dữ liệu, tiếp theo là T nhóm dòng, mỗi nhóm có dạng:

- Dòng đầu chứa số nguyên n ($n \leq 2 \times 10^5$);
- Dòng thứ hai chứa n số nguyên dương c_1, c_2, \dots, c_n ($c_i \leq 10^9$);
- $n - 1$ dòng tiếp theo, mỗi dòng chứa hai số nguyên mô tả đồ thị dạng cây;
- Dòng tiếp theo chứa số nguyên q ($q \leq 2 \times 10^5$);
- Dòng tiếp theo chứa q số nguyên r_1, r_2, \dots, r_q .

Tổng các giá trị n, q trong T bộ dữ liệu không vượt quá 2×10^5 .

Output

Mỗi dòng là số lượng cặp đỉnh đếm được chia dư cho $(10^9 + 7)$ tương ứng với các truy vấn.

Input	Output
1 5 3 1 2 2 1 1 2 1 3 2 4 3 5 2 1 2	1 0

Subtask 1: $n, q \leq 500$;

Subtask 2: $c_i \leq 3$;

Subtask 3: $c_i \leq 20$;

Subtask 4: Không có ràng buộc nào thêm.

Quà Noel

Món quà đặc biệt mà Ông già Tuyết làm tặng cho các bạn nhỏ chăm ngoan năm nay là một chiếc đồng hồ và một hộp quà bí mật. Bản mặt của chiếc đồng hồ có dạng hình chữ nhật, trên đó có một hình tròn là phần hiển thị giờ. Viền xung quanh hình tròn được chia thành n đoạn, được đánh thứ tự từ 1 đến n , mỗi đoạn được khắc một kí tự 0 hoặc 1. Để mở được hộp quà bí mật cần có mật khẩu, mật khẩu là một chuỗi nhị phân s độ dài m . Ông già tuyết muốn khắc viền xung quanh hình tròn trên bản mặt đồng hồ để xâu s xuất hiện trên viền. Xâu s được gọi là xuất hiện trên viền nếu tồn tại một vị trí p ($1 \leq p \leq n$) và lấy liên tiếp m kí tự theo chiều kim đồng hồ thì nhận được xâu s . Là người yêu thích Tin học, Ông già Tuyết muốn đếm xem có bao nhiêu cách khắc thỏa mãn, hai cách được gọi là khác nhau nếu tồn tại vị trí q ($1 \leq q \leq n$) mà hai kí tự khắc trên đó là khác nhau.

Yêu cầu: Cho n và xâu s . Hãy đếm số cách khắc thỏa mãn.

Input

- Dòng đầu chứa số nguyên n ;
- Dòng thứ hai chứa một xâu kí tự s chỉ gồm kí tự 0 hoặc 1 (độ dài xâu s không vượt quá n).

Output

- Gồm một dòng, chứa một số là số cách khắc thỏa mãn.

Input	Output
3	4
11	

Ràng buộc:

- Có 30% số test có $n \leq 20$;
- Có 30% số test khác có $n \leq 60$, độ dài xâu s không vượt quá 10;
- Có 40% số test còn lại có $n \leq 60$.