

موزاييك

تخطط سلمى لرسم لوحة فسيفسائية طينية على الحائط. اللوحة مكونة من شبكة أبعادها $N \times N$ تحوي N^2 قطعة مربعة أبعاد كل منها 1×1 وهي غير ملونة بالبداية. يتم ترقيم الأسطر في اللوحة من 0 إلى $N - 1$ من الأعلى إلى الأسفل، ويتم ترقيم الأعمدة من 0 إلى $N - 1$ من اليسار إلى اليمين. يتم ترميز القطعة في السطر رقم i والعمود رقم j بـ (i, j) . يجب تلوين كل قطعة إما باللون الأبيض (ونرمز له باللون 0) أو اللون الأسود (ونرمز له بالرمز 1).

لتلوين اللوحة، تختار سلمى في البداية مصفوفتين X و Y بطول N لكل منهما، تتكون كل واحدة منهما من القيم 0 و 1، بحيث $X[0] = Y[0]$. تلوّن سلمى القطع في أول سطر من الأعلى (أي السطر رقم 0) بحسب المصفوفة X حيث يكون لون القطعة $(0, j)$ هو $X[j]$ ($0 \leq j < N$). تقوم أيضاً بتلوين القطع في أول عمود على اليسار (أي العمود رقم 0) وفقاً للمصفوفة Y ، أي أنه يكون لون القطعة $(i, 0)$ هو $Y[i]$ ($0 \leq i < N$).

من بعدها تقوم سلمى بتكرار الخطوات التالية حتى تلوّن كل القطع:

- تبحث سلمى عن أي قطعة غير ملونة (i, j) بحيث أن القطعة المجاورة لها بالأعلى (أي القطعة $(i - 1, j)$) والقطعة المجاورة لها من اليسار (أي القطعة $(i, j - 1)$) كلاهما ملونتان.
- ومن ثم، تقوم بتلوين القطعة (i, j) بالأسود إذا كان كلا هاتين القطعتين المجاورتين لونهما أبيض؛ وإلا، فإنها تلوّن القطعة (i, j) باللون الأبيض.

يمكن إثبات أن التلوين النهائي للقطع لا يعتمد على الترتيب الذي قامت سلمى بتلوينها به.

لدى ياسمين فضول قوي للتعرف على ألوان القطع في اللوحة الفسيفسائية. ستقوم ياسمين بسؤال سلمى Q سؤالاً، مرقمة من 0 إلى $Q - 1$. في السؤال k ($0 \leq k < Q$)، تحدد ياسمين مستطيلاً جزئياً من اللوحة عن طريق:

- أول سطر في الأعلى $T[k]$ وأول سطر في الأسفل $B[k]$ ($0 \leq T[k] \leq B[k] < N$).
- أول عمود في اليسار $L[k]$ وأول عمود في اليمين $R[k]$ ($0 \leq L[k] \leq R[k] < N$).

الإجابة على هذا السؤال تكون عدد القطع السوداء في هذا المستطيل الجزئي. بشكل دقيق، يجب على سلمى أن تحسب عدد القطع (i, j) الموجودة، بحيث $T[k] \leq i \leq B[k]$ ، $L[k] \leq j \leq R[k]$ ، ولون القطعة (i, j) هو أسود.

اكتب برنامجاً للإجابة على أسئلة ياسمين.

تفاصيل البرمجة

يجب عليك برمجة التابع التالي.

```
std::vector<long long> mosaic(
    std::vector<int> X, std::vector<int> Y,
    std::vector<int> T, std::vector<int> B,
    std::vector<int> L, std::vector<int> R)
```

- Y, X : مصفوفتان بطول N يصفان لون القطع في أول سطر بالأعلى وأول عمود باليسار على التوالي.
- R, L, B, T : مصفوفات بطول Q تصف اسئلة باسمين.
- يجب على التابع أن يعيد مصفوفة C طولها Q ، حيث أن $C[k]$ هو جواب السؤال رقم k ($0 \leq k < Q$).
- سيتم استدعاء هذا التابع مرة واحدة بالنسبة لكل حالة اختبار.

القيود

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 200\,000$
- $X[i] \in \{0, 1\}$ و $Y[i] \in \{0, 1\}$ من أجل كل i حيث أن $0 \leq i < N$
- $X[0] = Y[0]$
- $0 \leq T[k] \leq B[k] < N$ و $0 \leq L[k] \leq R[k] < N$ من أجل كل k حيث أن $0 \leq k < Q$

المسائل الجزئية

المسألة الجزئية	العلامة	القيود الإضافية
1	5	$N \leq 2; Q \leq 10$
2	7	$N \leq 200; Q \leq 200$
3	7	$T[k] = B[k] = 0$ ($0 \leq k < Q$) من أجل كل k حيث أن
4	10	$N \leq 5000$
5	8	$X[i] = Y[i] = 0$ ($0 \leq i < N$) من أجل كل i حيث أن
6	22	$L[k] = R[k]$ و $T[k] = B[k]$ ($0 \leq k < Q$) من أجل كل k حيث أن
7	19	$T[k] = B[k]$ ($0 \leq k < Q$) من أجل كل k حيث أن
8	22	لا يوجد قيود إضافية.

مثال

ليكن لدينا الاستدعاء التالي.

```
mosaic([1, 0, 1, 0], [1, 1, 0, 1], [0, 2], [3, 3], [0, 0], [3, 2])
```

هذا المثال موضح بالصورة في الأسفل. الصورة اليسارية تعرض ألواناً للقطع في اللوحة الفسيفسائية. صورتان في المنتصف واليمين تعرضان المستطيلات الجزئية التي سألت عنها باسمين في السؤال الأول والثاني على التوالي.

	0	1	2	3
0	1	0	1	0
1	1	0	0	1
2	0	1	0	0
3	1	0	1	0

	0	1	2	3
0	1	0	1	0
1	1	0	0	1
2	0	1	0	0
3	1	0	1	0

	0	1	2	3
0	1	0	1	0
1	1	0	0	1
2	0	1	0	0
3	1	0	1	0

الجواب على هذه الأسئلة (أي اعداد الواحدات في المستطيلات المظلمة) هي 7 و 3, على التوالي. لذلك يجب على التابع أن يعيد [7, 3].

Sample Grader

:Input format

```
N
X[0] X[1] ... X[N-1]
Y[0] Y[1] ... Y[N-1]
Q
T[0] B[0] L[0] R[0]
T[1] B[1] L[1] R[1]
...
T[Q-1] B[Q-1] L[Q-1] R[Q-1]
```

:Output format

```
C[0]
C[1]
...
C[S-1]
```

.Here, S is the length of the array C returned by mosaic