

AUTUMN CONTEST 2020

Lưu ý: Các thí sinh có thể nộp bằng ngôn ngữ Pascal, C++ hoặc Java tùy theo ngôn ngữ nào thí sinh thấy thuận tiện. Tên các bài là Tenbai.*, trong đó * tương ứng với pas, cpp hay java. Tenbai là tên của bài sẽ được quy định trong cụ thể từng bài.

Bài 4 (AC4.*):

“Và hồn tôi, từ đó, là khúc ca vang trong ngàn, làn đôi môi, rục rờ tình ban đầu. Hòa vào cây vượn vào nắng, và giấc mơ tôi có nàng, trong bài ca, ta bước, thênh thang....”

Đó là những lời ca vàng ngọc trong bài hát mà Ngạn đã sáng tác và dành tặng cho Hà Long. Và chúng ta có thể thấy rằng Hà Long rút cuộc vẫn không rơi vào vòng tay của Ngạn. Dũng đã tán đổ được Hà Long bằng cây guitar điện của mình, vậy một trong những lý do là gì ? Bản nhạc của Ngạn quá êm đềm, nhẹ nhàng và không có nhiều điểm cao trào trong bài hát.



Một đoạn được gọi là “cao trào” của bài hát, là khi với 3 nốt nhạc liên tục với nhau, nếu như nốt ở giữa là nốt cao hơn 2 nốt còn lại, hoặc nốt ở giữa là nốt thấp hơn 2 nốt còn lại. Một bản nhạc của chúng ta chỉ gồm 7 nốt “Do – Re – Mi – Fa – Son – La – Si” và các nốt này được sắp xếp tăng dần theo độ cao. Một bản nhạc có lẽ là hợp với Hà Long khi trong bản nhạc đó có tối thiểu M đoạn cao trào. Hãy giúp Ngạn kiểm tra xem bài hát mà Ngạn sáng tác, có giúp cuốn hút được Hà Long không nhé ?

- **Dữ liệu đầu vào (AC4.INP):**
 - o Đọc từ file AC4.INP

- Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 200$) là số lượng bộ dữ liệu của đề bài.
- T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức như sau:
 - Dòng đầu tiên gồm 2 số nguyên dương N và M , đại diện cho số lượng nốt nhạc trong bản nhạc của Ngạn và số lượng đoạn cao trào tối thiểu cần để làm xiêu lòng Hà Long.
 - Dòng tiếp theo gồm N chuỗi. Các chuỗi ký tự chỉ gồm 1 trong các chuỗi (“Do”, “Re”, “Mi”, “Fa”, “Son”, “La”, “Si”).
- **Dữ liệu đầu ra (AC4.OUT):**
 - Ghi ra file AC4.OUT
 - Với bộ dữ liệu thứ i , ta in ra 1 dòng duy nhất là chuỗi “YES” hoặc “NO” tương ứng là câu trả lời rằng bản nhạc mà Ngạn sáng tác có hợp với Hà Long hay không.
- **Ví dụ:**

AC4.INP	AC4.OUT
5	NO
3 1	YES
Do Re Mi	YES
3 1	NO
Do Mi Re	YES
5 1	
Do Mi Re Fa Mi	
5 2	
Do Re Mi Fa Son	
9 3	
Do Mi Re Son Do Fa Si Re Mi	

- **Giải thích test ví dụ:**
 - Với test ví dụ 1: Bản nhạc của Ngạn có 3 nốt và các nốt này chỉ tăng dần theo cao độ. Do đó không có cao trào nên kết quả là NO.
 - Các test ví dụ còn lại, bạn đọc hãy tự tìm câu trả lời.
- **Bài toán chỉ có duy nhất 1 Dataset:**
 - $1 \leq M \leq N \leq 100$.
- **Giới hạn thời gian và bộ nhớ:**
 - 3s / Dataset.
 - 512Mb / Dataset.

Bài 5 (AC5.*):

Bạn được cho một số nguyên dương K và một bộ gồm N khóa. Khóa thứ i mang giá trị là key_i . Ta có giá trị ban đầu là 1. Trong mỗi bước biến đổi, ta có thể chọn bất kỳ một khóa nào trong bộ N khóa trên và nhân giá trị hiện tại cho khóa đó. Bạn hãy trả về số bước biến đổi ít nhất để biến đổi giá trị ban đầu thành số K , hoặc trả ra Impossible nếu như không tồn tại cách biến đổi nào có thể thực hiện. Biết rằng mỗi khóa có thể sử dụng nhiều lần.

- **Dữ liệu đầu vào (AC5.INP):**
 - o Đọc từ file AC5.INP.
 - o Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 40$) là số lượng bộ dữ liệu cần xử lý
 - o T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức như sau:
 - Dòng đầu tiên chứa 2 số nguyên dương N và K .
 - Dòng thứ hai gồm N số nguyên dương, số nguyên dương thứ i mang giá trị key_i .
- **Dữ liệu đầu ra (AC5.OUT):**
 - o Ghi vào file AC5.OUT.
 - o Gồm T dòng, dòng thứ i là số lượng bước tối thiểu để biến từ số 1 thành số K thông qua các bước biến đổi, hoặc một chuỗi “Impossible” nếu như không tồn tại cách biến đổi từ số 1 thành số K .

- **Ví dụ:**

AC5.INP	AC5.OUT
4	2
3 12	0
2 3 4	Impossible
1 1	Impossible
100	
5 9	
2 4 6 8 10	
8 10	
1 1 1 1 1 1 1 1	

- **Giải thích test ví dụ:**
 - o Ví dụ 1: Ta cần tối thiểu 2 lần biến đổi. Lần thứ 1, sử dụng khóa số 2 để biến đổi, từ 1 trở thành 3. Lần thứ 2, sử dụng khóa số 3 để biến đổi, từ 3 trở thành 12. Chúng ta còn một cách khác để có thể biến đổi 1 thành 12, đó là sử dụng 2 lần khóa số 1 và sử dụng 1 lần khóa số 2. Tuy nhiên cách này mất 3 bước biến đổi.
 - o Ví dụ 2: Giá trị K là 1 rồi nên không cần một phép biến đổi nào cả.
 - o Ví dụ 3: Giá trị K là một số lẻ, nhưng các khóa chỉ bao gồm số chẵn, do đó không thể biến đổi từ giá trị 1 sang 9 được.
 - o Ví dụ 4: Vì các khóa chỉ gồm số 1, do đó không thể nào thay đổi giá trị ban đầu thành 1 số khác được.
- **Bài toán sẽ được chia thành 2 Dataset:**
 - o Small Dataset: $1 \leq N \leq 10, 1 \leq K \leq 1000$
 - o Large Dataset: $1 \leq N \leq 100, 1 \leq K \leq 100000$.
 - o Trong mọi Dataset, ta luôn có $1 \leq key_i \leq 100$.
- **Giới hạn dữ liệu:**
 - o 5s / Dataset.
 - o 512Mb / Dataset.

Bài 6 (AC6.*):

Bạn có một nền đất cần lát gạch có kích thước $1 \times N$ (gồm N viên gạch hình vuông kích thước 1×1). Trên đó có những ô còn nguyên vẹn và những ô đã hỏng. Ta có M ô gạch bị hỏng và các ô gạch bị hỏng được ký hiệu là '#', các ô gạch nguyên vẹn được ký hiệu là '.'. Ta có 3 loại gạch men như sau:

- Loại 1: Gạch men kích thước 1×1 , chỉ được sử dụng để lát ô còn nguyên vẹn (‘.’) và độ thẩm mỹ khi lát 1 viên gạch loại 1 là G_1 . Ngoài ra, ta chỉ có tối đa K viên gạch loại 1.
- Loại 2: Gạch men kích thước 1×2 , chỉ được sử dụng để lát 2 ô còn nguyên vẹn và liên tiếp với nhau (“..”) và độ thẩm mỹ khi lát 1 viên gạch loại 2 là G_2 . Các viên gạch loại 2 không giới hạn số lượng.
- Loại 3: Gạch men kích thước 1×3 , chỉ được sử dụng để lát 3 ô gạch liên tục, trong đó ô chính giữa là ô gạch bị hỏng, và 2 viên bên cạnh là 2 ô nguyên vẹn (“.#.”). Các viên gạch loại 3 không giới hạn số lượng. Độ thẩm mỹ khi lát 1 viên gạch loại 3 là G_3 .

Bạn hãy tìm cách lát nền gạch sao cho độ thẩm mỹ là lớn nhất. Ta không được phép lát những viên gạch đặt chồng lên nhau, cũng không được phép chia nhỏ các viên gạch của các loại thành những phần nhỏ hơn. Biết rằng không bắt buộc phải lát toàn bộ N ô gạch. Với mỗi ô gạch bị hỏng sau khi lát, độ thẩm mỹ sẽ giảm mất một lượng là D . Độ thẩm mỹ sẽ không thay đổi với các ô gạch còn nguyên vẹn nhưng không được lát.

- **Dữ liệu đầu vào (AC6.INP):**

- Dòng đầu tiên gồm một số nguyên dương T ($1 \leq T \leq 40$), là số lượng bộ dữ liệu.
- T bộ dữ liệu tiếp theo, mỗi bộ dữ liệu được tổ chức dưới dạng như sau:
 - Dòng 1 gồm 6 số nguyên dương cách nhau bởi các dấu cách, lần lượt là $N, M, K, D, G_1, G_2, G_3$.
 - Dòng 2 gồm 1 chuỗi ký tự, chỉ gồm các ký tự ‘.’ và ‘#’, đại diện cho trạng thái của nền gạch, trong đó số lượng ký tự ‘#’ là M .

- **Dữ liệu đầu ra (AC6.OUT):**

- Ta in ra đáp án với T bộ dữ liệu tương ứng. Với dòng thứ i , ta cần trả ra một số nguyên là độ thẩm mỹ lớn nhất để lát nền gạch.

- **Ví dụ:**

AC6.INP	AC6.OUT
5	12
5 1 2 4 5 6 1	100
..#..	100
1 0 4 5 100 3 2	-200
.	77
2 0 1 100 1 100 2	
..	
2 2 1 100 1 100 2	
##	
20 3 10 30 5 3 9	
..#.....#...#.....	

- **Giải thích test ví dụ:**

- Với test ví dụ 1:
Ta sử dụng 1 viên gạch loại 2 để lát 2 ô trống số 1 và 2. Sau đó dùng 2 viên gạch loại 1 để lát 2 ô trống số 4 và 5. Vì ô số 3 là ô gạch bị hỏng và không được lát nên tổng độ đẹp là $6 + 2 * 5 - 4 = 12$.

- Với test ví dụ 2:
Chỉ có duy nhất một ô gạch còn nguyên vẹn nên ta sử dụng 1 viên gạch loại 1 để lát nên độ thẩm mỹ là 100.
- Với test ví dụ 3:
Vì số lượng viên gạch loại 1 chỉ có tối đa là 1 viên nên không thể lát được 2 ô gạch còn trống. Do đó ta sử dụng 1 viên gạch loại 2 và độ đẹp là 100.
- Với test ví dụ 4:
Nền gạch gồm 2 ô hỏng, do đó không thể lát được viên gạch nào vào. Do đó độ thẩm mỹ của nền gạch là -200 vì với mỗi ô gạch hỏng, độ thẩm mỹ giảm mất 100 đơn vị.
- Với test ví dụ 5:
Dùng để kiểm tra cách làm của bạn có đúng hay không.
- **Bài toán sẽ được chia làm 2 Dataset:**
 - Small Dataset: $M = 0$.
 - Large Dataset: $0 \leq M \leq 50$.
 - Trong mọi Dataset, ta có: $1 \leq K \leq N \leq 100000$, $0 \leq D$, $G_1, G_2, G_3 \leq 100$.
- **Giới hạn dữ liệu:**
 - 5s / Dataset.
 - 512Mb / Dataset.

-----Chúc các bạn làm bài tốt <3 <3 <3 -----