# Xóa k chữ số

## Mô tả:

Ta có một số nguyên có n chữ số. Hãy tìm cách xóa đi k chữ số trong số nguyên trên sao cho kết quả thu được là số nhỏ nhất có thể.

## Input:

Một số nguyên dương a, có tối đa 10 chữ số

Một số nguyên k, là số chữ số cần phải xóa trong a (0<=k<=100)

## Output:

Một số nguyên là giá trị nhỏ nhất thu được khi xóa k chữ số trong a. Trong trường hợp không xóa được do k>số chữ số trong a thì xuất “error”.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 2070880  2 | 880 |
| 0034325670  7 | 0 |

# Basic Scheduling. Heuristic algorithm

## Mô tả:

Có n công việc, được phân công cho m máy như nhau thực hiện đồng thời. Hãy viết chương trình dùng kỹ thuật THAM LAM để tìm một phương án phân công sao cho thời gian để các máy hoàn thành hết n công việc là ngắn nhất.

## Input:

Dòng đầu tiên chứa hai số n và m

Dòng thứ hai chứa n số nguyên dương x0, x1, …, xn-1 trong đó xi là thời gian cần thiết để một máy hoàn thành công việc i.

## Output:

Xuất trên một dòng n số nguyên dương y0, y1, …,yn-1, giá trị mỗi số trong đoàn [0, m-1] trong đó giá trị của yi là số thứ tự của máy được phân công để thực hiện công việc i.

LƯU Ý: Vì dùng Tham lam nên chấp nhận sai số, nếu chênh lệch thời gian không lớn (<10%) thì hệ thống sẽ vẫn chấp nhận. GV đưa ví dụ không phải là trường hợp tối ưu để nhấn mạnh điểm này.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 8 3  79 1 80 59 75 51 7 29 | 1 1 0 2 2 1 2 0 |

# Different Machine Scheduling. Heuristic Algorithm

## Mô tả:

Có n công việc, được phân công cho m máy khác nhau thực hiện đồng thời. Hãy viết chương trình dùng kỹ thuật THAM LAM để tìm một phương án phân công sao cho thời gian để các máy hoàn thành hết n công việc là ngắn nhất.

## Input:

Dòng đầu tiên chứa hai số n và m

m dòng tiếp theo, mỗi dòng chứa n số nguyên dương x0, x1, …, xn-1 Trong đó số xi ở dòng thứ j là thời gian cần thiết để máy thứ j hoàn thành công việc i.

## Output:

Xuất trên một dòng n số nguyên dương y0, y1, …,yn-1, giá trị mỗi số trong đoạn [0, m-1] trong đó giá trị của yi là số thứ tự của máy được phân công để thực hiện công việc i.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 8 3  79 1 80 59 75 51 7 29  49 71 44 55 22 61 31 88  59 28 9 19 18 60 54 68 | 1 0 2 2 2 0 0 0 |

# Different Machine Constraint Scheduling. Heuristic Algorithm (Bonus)

## Mô tả:

Có n công việc, được phân công cho m máy khác nhau thực hiện đồng thời. Trong số các công việc trên, có những công việc chỉ có thể được thực hiện trên một số máy nhất định. Hãy viết chương trình dùng thuật giải heuristic tìm một phương án phân công sao cho thời gian để các máy hoàn thành hết n công việc là ngắn nhất.

## Input:

Dòng đầu tiên chứa hai số n và m

m dòng tiếp theo, mỗi dòng chứa n số nguyên dương x0, x1, …, xn-1 Trong đó số xi ở dòng thứ j là thời gian cần thiết để máy thứ j hoàn thành công việc i, nếu xi có giá trị nhỏ hơn 0 điều đó có nghĩa máy thứ j không thể thực hiện công việc i

## Output:

Xuất trên một dòng n số nguyên dương y0, y1, …,yn-1, giá trị mỗi số trong đoạn [0, m-1] trong đó giá trị của yi là số thứ tự của máy được phân công để thực hiện công việc i.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 4 3  684 -1 -1 886  465 686 377 898  565 995 902 -1 | 1 2 1 0 |

# TPS. Check complete weighted graph (Step1)

## Mô tả:

Đơn đồ thị hữu hướng (simple complete dirgraph) đầy đủ là một đồ thị mà trong đó giữa hai đỉnh u, v bất kỳ sẽ được nối với nhau bởi duy nhất một cặp cạnh (một cạnh có hướng từ u sang v và cạnh còn lại từ v sang u).

Cho một đồ thị hãy kiểm tra xem đồ thị đó có đầy đủ hay không

## Input:

Dòng đầu tiên chứa số e, đây là độ dài của input

e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u và i (cách nhau bởi khoảng trắng), thể hiện việc có một cạnh nối từ đỉnh u sang đỉnh i trong đồ thị. Do đây là đơn đồ thị nên nếu như có nhiều dòng cùng chứa hai chuỗi u, i giống nhau vẫn chỉ xem như có một cạnh nối từ u sang i.

## Output:

Xuất ra màn hình chuỗi TRUE nếu đồ thị là đầy đủ. Nếu không xuất ra chuỗi FALSE.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 6  K I  O I  I K  K O  I O  O K | TRUE |

# TPS. Check valid cycle (Step2)

## Mô tả:

Với một cách hiểu đơn giản ta có thể hình dung một đường đi (walk) trên đồ thị là một danh sách có thứ tự các đỉnh trong đó phần tử đầu tiên và phần tử cuối cùng của danh sách là cùng một đỉnh và tồn tại một cạnh nối giữa hai phần tử liên tiếp trong danh sách.

Cho một đơn đồ thị hữu hướng với cạnh có trọng số, và một số đường đi, kiểm tra xem các đường đi đó có đúng là đường đi hợp lý trên đồ thị hay không và tính độ dài của đường đi đó

## Input:

Dòng đầu tiên chứa 02 số e và n, đây là số cạnh của đồ thị và số đường đi cần kiểm tra

e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u, i và số nguyên dương x (cách nhau bởi khoảng trắng), thể hiện việc có một cạnh với độ dài là x nối từ đỉnh u sang đỉnh i trong đồ thị. Tên của các đỉnh chỉ bao gồm các ký tự trong bảng chữ cái.

n dòng tiếp theo, mỗi dòng chứa một dánh sách các chuỗi cách nhau bởi khoảng trắng, danh sách kết thúc khi gặp dấu chấm câu (.) Đây là các đỉnh trên đường đi cần kiểm tra.

## Output:

Ứng với mỗi đường đi, nếu đó đúng là một đường đi trên đồ thị, xuất ra màn hình độ dài của đường đi đó trên một dòng. Nếu đó không phải là đường đi hợp lý, xuất ra chuỗi FALSE.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 8 3  K F 151  F L 199  L K 338  H L 8  K H 491  F H 803  K L 579  H K 299  F L F .  L K H F L .  L K F L . | FALSE  FALSE  688 |

# TPS. Heuristic Algorithm

## Mô tả:

Với một cách hiểu đơn giản ta có thể hình dung một đường đi (walk) trên đồ thị là một danh sách có thứ tự các đỉnh trong đó phần tử đầu tiên và phần tử cuối cùng của danh sách là cùng một đỉnh và tồn tại một cạnh nối giữa hai phần tử liên tiếp trong danh sách.

Có một người đưa thư cần tìm đường đi qua tất cả các đỉnh trên đồ thị (đầy đủ) sao cho mỗi đỉnh chỉ đi duy nhất một lần và trở về vị xuất phát ban đầu. Hãy viết chương trình tìm đường đi đáp ứng yêu cầu đó và có độ dài càng ngắn càng tốt sử dụng thuật giải heuristic

## Input:

Dòng đầu tiên chứa số e và chuỗi s, đây là số cạnh của đồ thị và tên của đỉnh mà từ đó người đưa thư bắt đầu đi.

e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u, i và số nguyên dương x (cách nhau bởi khoảng trắng), thể hiện việc có một cạnh với độ dài là x nối từ đỉnh u sang đỉnh i trong đồ thị. Tên của các đỉnh chỉ bao gồm các ký tự trong bảng chữ cái.

## Output:

Xuất ra danh sách các đỉnh trên đường đi cách nhau bởi khoảng trắng.

LƯU Ý: Những bài dùng thuật giải heuristic thì có thể có nhiều cách làm (do chọn heuristic khác nhau) nên mấy đứa có thể làm cách khác với cách cô gợi ý trên lớp (thâm chí có thể cô ko cài đặt giống cách giải như hướng dẫn trên lớp luôn). Do đó, Cô ko so khớp đúng đáp án trong VD, chỉ cần SV cho kết quả gần đúng là ok rồi.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 20 L  L P 780  T P 90  P L 76  M T 300  L H 218  L T 994  P H 851  M P 648  H T 591  H M 246  T M 194  T H 186  H L 328  P M 723  P T 65  H P 807  L M 706  T L 75  M L 133  M H 382 | L H M T P L |

# Graph coloring. Degree of vertices (Step1)

## Mô tả:

Cho một đơn đồ thị vô hướng, cho biết mỗi đỉnh trong đồ thị có bậc bao nhiêu.

## Input:

Dòng đầu tiên chứa 02 số v, e, đây lần lượt là số đỉnh, số cạnh của đồ thị

Dòng tiếp theo chứa v chuỗi, mỗi chuỗi bao một hoặc nhiều ký tự hoa trong bảng chữ cái Tiếng Anh và cách nhau bởi khoảng trắng. Đây là danh sách tên các đỉnh của đồ thị.

e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u và i (cách nhau bởi khoảng trắng), thể hiện việc có một cạnh nối giữa đỉnh u và đỉnh i trong đồ thị.

## Output:

Xuất ra trên một dòng bậc của mỗi đỉnh, thứ tự các đỉnh ứng với thứ tự chúng xuất hiện trong dòng thứ 2 của input.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 7 18  J R Q F L P I  L I  L F  F R  F J  L J  J R  J P  F Q  P L  F I  R P  I Q  Q L  F P  J I  I R  P Q  P I | 5 4 4 6 5 6 6 |

# Graph coloring. Color selection (Step2)

## Mô tả:

Cho một đồ thị đang được tô màu dang dở, hãy thực hiện việc chọn màu để tiếp tục tô cho đồ thị. Màu được sử dụng là hệ màu RGB 24-bit với tối đa hơn 16 triệu màu, các màu được đánh số từ 0 (màu đen #000000) đến 16777215 (màu trắng #FFFFFF).

Quy định là khi tô thì đỉnh không được trùng màu với các đỉnh lân cận của nó. Khi chọn màu chỉ được sử dụng các màu đã tô trước đó, trừ khi những đỉnh lân cận với đỉnh cần tô đã chiếm hết những màu đã sử dụng trước thì mới được chọn màu mới. Khi tô màu dù là các màu đã có hay chọn màu mới, luôn chọn màu có thứ tự nhỏ nhất có thể chọn được.

## Input:

Dòng đầu tiên chứa 03 số v, e, n đây lần lượt là số đỉnh, số cạnh của đồ thị và số lượng đỉnh cần xem xét.

Dòng tiếp theo chứa v chuỗi, mỗi chuỗi bao một hoặc nhiều ký tự hoa trong bảng chữ cái Tiếng Anh và cách nhau bởi khoảng trắng. Đây là danh sách tên các đỉnh của đồ thị.

e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u và i (cách nhau bởi khoảng trắng), thể hiện việc có một cạnh nối giữa đỉnh u và đỉnh i trong đồ thị.

Dòng áp chót trong input chứa v số nguyên, đây tương ứng là màu của các đỉnh đã tô. Số nguyên đầu tiên là màu của đỉnh đầu tiên, đỉnh chưa tô màu sẽ có số tương ứng là -1

Dòng cuối cùng trong input chứa n chuỗi, đây là các đỉnh mà người chơi đang muốn xem xét để chọn màu.

## Output:

Ứng với mỗi đỉnh cần xem xét thực hiện yêu cầu sau:

Nếu đỉnh đã có màu, kiểm tra màu đó có trùng với màu của các đỉnh lân cận hay không. Nếu không trùng xuất ra chuỗi TRUE trên một dòng

Nếu đỉnh chưa có màu hoặc đã có nhưng màu đó trùng với màu của một hoặc nhiều đỉnh lân cận thì chọn màu cho đỉnh theo đúng quy tắc và xuất ra màu đó trên một dòng.

Yêu cầu xem xét được thực hiện độc lập cho mỗi đỉnh. Kết quả xem xét các đỉnh trước hoàn toàn không ảnh hưởng gì đến các đỉnh sau.

GIẢI THÍCH THÊM THEO YÊU CẦU CỦA SV:

- Khi chọn màu chỉ được sử dụng các màu đã tô trước đó (có thể không bắt đầu từ 0). Khi tô màu dù là các màu đã có hay chọn màu mới, luôn chọn màu có thứ tự nhỏ nhất có thể chọn được.

VD: màu đã sử dụng = {2,3,4}. Khi thử tô lại màu đã sử dụng thì thử bắt đầu từ màu 2. Nếu thử hết các màu trên mà ko được thì phải tô màu mới, nhưng màu mới phải là màu có thứ tử nhỏ nhất có thể chọn được tức là màu 0.

- Theo VD 1, xét các đỉnh độc lập nghĩa là: sau khi tô màu cho đỉnh H xong thì bài toán quay trở lại trạng thái như ban đầu, các màu được tô (0 -1 0 -1 -1)

- Ứng với mỗi đỉnh cần xem xét thực hiện yêu cầu sau:

+ Nếu đỉnh đã có màu và tô đúng thì xuất TRUE.

+ Nếu đỉnh chưa có màu thì tô nó.

+Nếu đỉnh đã tô nhưng tô sai thì tô lại "Theo ví dụ 2: RU kề nhau nhưng tô cùng màu 1 là sai.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 5 6 2  I K L G H  I H  I K  I G  K L  H K  L I  0 -1 0 -1 -1  H G | 1  1 |

# Graph coloring. Heuristic algorithm

## Mô tả:

Cho một đơn đồ thị vô hướng. Hãy tô màu cho đồ thị bằng PHƯƠNG PHÁP THAM LAM sao cho các đỉnh lân cận nhau không trùng màu với nhau và số màu phải dùng là nhỏ nhất có thể. Màu được sử dụng là hệ màu RGB 24-bit với tối đa hơn 16 triệu màu, các màu được đánh số từ 0 (màu đen #000000) đến 16777215 (màu trắng #FFFFFF).

## Input:

Dòng đầu tiên chứa 02 số v, e, đây lần lượt là số đỉnh, số cạnh của đồ thị

Dòng tiếp theo chứa v chuỗi, mỗi chuỗi bao một hoặc nhiều ký tự hoa trong bảng chữ cái Tiếng Anh và cách nhau bởi khoảng trắng. Đây là danh sách tên các đỉnh của đồ thị.

e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u và i (cách nhau bởi khoảng trắng), thể hiện việc có một cạnh nối giữa đỉnh u và đỉnh i trong đồ thị.

## Output:

Xuất ra trên một dòng dãy v số nguyên, tương ứng là màu cho các đỉnh. Số đầu tiên tương ứng là màu cho đỉnh đầu tiên trong danh sách chuỗi các đỉnh ở dòng thứ 2 của input.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 7 18  J R Q F L P I  L I  L F  F R  F J  L J  J R  J P  F Q  P L  F I  R P  I Q  Q L  F P  J I  I R  P Q  P I | 0 1 0 2 1 3 4 |

# Basic Scheduling. Exact algorithm (Bonus)

## Mô tả:

Có n công việc, được phân công cho m máy như nhau thực hiện đồng thời. Hãy viết chương trình tìm một phương án phân công sao cho thời gian để các máy hoàn thành hết n công việc là ngắn nhất. (Thời gian bắt buộc phải ngắn nhất, không chấp nhận sai số)

## Input:

Dòng đầu tiên chứa hai số n và m

Dòng thứ hai chứa n số nguyên dương x0, x1, …, xn-1 trong đó xi là thời gian cần thiết để một máy hoàn thành công việc i.

## Output:

Xuất trên một dòng n số nguyên dương y0, y1, …,yn-1, giá trị mỗi số trong đoàn [0, m-1] trong đó giá trị của yi là số thứ tự của máy được phân công để thực hiện công việc i.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 8 3  722 985 380 129 966 310 56 374 | 1 2 1 1 0 2 1 0 |

# Graph coloring. Exact algorithm (Bonus)

## Mô tả:

Cho một đơn đồ thị vô hướng. Hãy tô màu cho đồ thị sao cho các đỉnh lân cận nhau không trùng màu với nhau và số màu phải dùng là tối thiểu. Màu được sử dụng là hệ màu RGB 24-bit với tối đa hơn 16 triệu màu, các màu được đánh số từ 0 (màu đen #000000) đến 16777215 (màu trắng #FFFFFF).

## Input:

Dòng đầu tiên chứa 02 số v, e, đây lần lượt là số đỉnh, số cạnh của đồ thị

Dòng tiếp theo chứa v chuỗi, mỗi chuỗi bao một hoặc nhiều ký tự hoa trong bảng chữ cái Tiếng Anh và cách nhau bởi khoảng trắng. Đây là danh sách tên các đỉnh của đồ thị.

e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u và i (cách nhau bởi khoảng trắng), thể hiện việc có một cạnh nối giữa đỉnh u và đỉnh i trong đồ thị.

## Output:

Xuất ra trên một dòng dãy v số nguyên, tương ứng là màu cho các đỉnh. Số đầu tiên tương ứng là màu cho đỉnh đầu tiên trong danh sách chuỗi các đỉnh ở dòng thứ 2 của input.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 7 18  J R Q F L P I  L I  L F  F R  F J  L J  J R  J P  F Q  P L  F I  R P  I Q  Q L  F P  J I  I R  P Q  P I | 0 1 0 2 1 3 4 |

# TPS. Exact Algrorithm (Bonus)

## Mô tả:

Với một cách hiểu đơn giản ta có thể hình dung một đường đi (walk) trên đồ thị là một danh sách có thứ tự các đỉnh trong đó phần tử đầu tiên và phần tử cuối cùng của danh sách là cùng một đỉnh và tồn tại một cạnh nối giữa hai phần tử liên tiếp trong danh sách.

Có một người đưa thư cần tìm đường đi qua tất cả các đỉnh trên đồ thị sao cho mỗi đỉnh chỉ đi duy nhất một lần và trở về vị xuất phát ban đầu. Hãy viết chương trình tìm đường đi đáp ứng yêu cầu đó và có độ dài tối thiểu.

## Input:

Dòng đầu tiên chứa 2 số e và chuỗi s, đây là số cạnh của đồ thị và tên của đỉnh mà từ đó người đưa thư bắt đầu đi.

e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u, i và số nguyên dương x (cách nhau bởi khoảng trắng), thể hiện việc có một cạnh với độ dài là x nối từ đỉnh u sang đỉnh i trong đồ thị. Tên của các đỉnh chỉ bao gồm các ký tự trong bảng chữ cái.

## Output:

Xuất ra danh sách các đỉnh trên đường đi cách nhau bởi khoảng trắng.

## Ví dụ:

|  |  |
| --- | --- |
| Input | Output |
| 20 L  L P 780  T P 90  P L 76  M T 300  L H 218  L T 994  P H 851  M P 648  H T 591  H M 246  T M 194  T H 186  H L 328  P M 723  P T 65  H P 807  L M 706  T L 75  M L 133  M H 382 | L H M T P L |