**Cấu trúc dữ liệu và giải thuật**

**THI HỌC KỲ - CA 2**

**XA03 - ĐỨNG ĐẦU**

**Độ khó:**1

**Mô tả đề bài:**

Có N người xếp hàng với N cái tên phân biệt. Người ta muốn ưu tiên một người duy nhất và người đó sẽ luôn luôn ở đầu hàng. Hãy liệt kê tất cả các cách xếp hàng thoả mãn theo thứ tự từ điển.

**Input**

Dòng đầu ghi số N (1 < N < 10) là số người xếp hàng

Dòng 2 ghi N cái tên khác nhau từng đôi một, mỗi cái tên là một dãy ký tự độ dài không quá 30 và không có khoảng trống.

Dòng 3 ghi tên người luôn đứng đầu hàng. Dữ liệu đảm bảo tên của người đứng đầu luôn có trong danh sách ở dòng 2.

**Output**

Ghi ra danh sách tất cả các cách xếp hàng theo thứ tự từ điển tăng dần.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  DONG TAY NAM BAC  NAM | NAM BAC DONG TAY  NAM BAC TAY DONG  NAM DONG BAC TAY  NAM DONG TAY BAC  NAM TAY BAC DONG  NAM TAY DONG BAC |

**XA04 - ĐỨNG CUỐI**

**Độ khó:**1

**Mô tả đề bài:**

Có N người xếp hàng với N cái tên phân biệt. Người ta muốn xếp một người duy nhất luôn đứng ở cuối hàng. Hãy liệt kê tất cả các cách xếp hàng thoả mãn theo thứ tự từ điển.

**Input**

Dòng đầu ghi số N (1 < N < 10) là số người xếp hàng

Dòng 2 ghi N cái tên khác nhau từng đôi một, mỗi cái tên là một dãy ký tự độ dài không quá 30 và không có khoảng trống.

Dòng 3 ghi tên người luôn đứng cuối hàng. Dữ liệu đảm bảo tên của người đứng cuối luôn có trong danh sách ở dòng 2.

**Output**

Ghi ra danh sách tất cả các cách xếp hàng theo thứ tự từ điển tăng dần.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  DONG TAY NAM BAC  NAM | BAC DONG TAY NAM  BAC TAY DONG NAM  DONG BAC TAY NAM  DONG TAY BAC NAM  TAY BAC DONG NAM  TAY DONG BAC NAM |

 #include <bits/stdc++.h>

using namespace std;

int n,x[1005] = {};

bitset<10005> check;

vector<string> v(105);

void init(int m){

x[n] = m;

check[m] = 1;

int cnt = 1;

for(int i = 1; i <= n; i++){

if(!check[i]){

x[cnt] = i;

cnt++;

}

}

}

void Try(int i){

for(int j = 1; j <= n; j++){

if(!check[j] && j != x[n]){

check[j] = 1;

x[i] = j;

if(i == n - 1){

for(int i = 1; i <= n; i++) cout << v[x[i]] << ' ';

cout << endl;

}

else Try(i + 1);

check[j] = 0;

}

}

}

int main(){

ios\_base::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

int m;

cin >> n;

string s;

for(int i = 1; i <= n ;i++) cin >> v[i];

cin >> s;

for(int i = 1; i <= n; i++)

{

if(s == v[i]){

m = i;

break;

}

}

init(m);

Try(1);

}

**XB01 - VỊ TRÍ XUẤT PHÁT**

**Độ khó:**2

**Mô tả đề bài:**

Cuộc đua xe công thức 1 đã chuẩn bị sẵn N vị trí xuất phát là các giá trị nguyên. Có M tay đua tham gia. Người ta muốn sắp xếp các xe đua vào các vị trí sao cho khoảng cách nhỏ nhất giữa hai chiếc xe là lớn nhất.

Hãy tính giá trị lớn nhất này.

**Input**

Dòng đầu tiên ghi hai số N, M (2 ≤ M ≤ N ≤ 105).

N dòng tiếp theo, mỗi dòng ghi một vị trí xuất phát. Các giá trị là các số nguyên không âm và không quá 109.

**Output**

Ghi ra kết quả của bài toán.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 3  1  2  8  4  9 | 3 |

*Giải thích ví dụ: các xe đua được xếp ở các vị trí 1 - 4 - 8*

**XB04 - DÃY CON KHÔNG LIÊN TIẾP BẬC K**

**Độ khó:**1

**Mô tả đề bài:**

Cho dãy số A[] có N phần tử là các số nguyên dương.

Với mỗi số nguyên K, một **dãy con không liên tiếp bậc K** được định nghĩa là dãy con của dãy A[] trong đó không có K số liên tiếp nào trong dãy A[] được chọn (tức là chỉ được phép chọn tối đa K-1 số liên tiếp).

Hãy tính tổng lớn nhất có thể của dãy con không liên tiếp bậc K.

**Input**

* Dòng đầu tiên ghi hai số N và K (1 ≤ N ≤ 10000; 1 ≤ K ≤ 1000)
* Dòng tiếp theo ghi N số của dãy A[], các giá trị đều nguyên dương và không quá 1000

**Output**

Ghi ra kết quả tính được trên một dòng.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 2  10 10 20 20 10 | 40 |

**XB06 - XÂU CON KHÔNG GIẢM**

**Độ khó:**1

**Mô tả đề bài:**

Cho một xâu ký tự S chỉ bao gồm các chữ cái in hoa. Xâu con được định nghĩa là một tập con khác rỗng các ký tự lấy từ xâu S giữ nguyên thứ tự ban đầu.

Xâu con không giảm C của S là xâu con của S thỏa mãn với mọi cặp vị trí i < j trong xâu thì C[i] ≤ C[j].

Hãy tính độ dài của xâu con không giảm dài nhất trong xâu S.

**Input**

Chỉ có một dòng ghi xâu S. Độ dài không quá 200.

**Output**

Ghi ra kết quả tính được trên một dòng

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| VIETNAM | 2 |
| ABCC | 4 |

**XC01 - HẬU TỐ**

**Độ khó:**2

**Mô tả đề bài:**

Biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ AB+CD-\* (tương ứng với biểu thức trung tố (A+B)\*(C-D).

Hãy viết chương trình tính toán giá trị của biểu thức hậu tố.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi số nguyên N (N ≤ 1000), là số lượng toán tử và toán hạng.

Dòng tiếp theo là biểu thức hậu tố với N toán tử và toán hạng. Các toán hạng và toán tử cách nhau bởi một dấu cách.

Các toán hạng có thể có nhiều chữ số, có thể là số âm. Phép chia thực hiện trên số nguyên. Input đảm bảo kết quả phép toán có giá trị tuyệt đối không vượt quá 10^18.

**Output:**

Với mỗi test, in ra giá trị biểu thức tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 3  7  2 3 1 \* + 9 –  7  8 7 5 \* + 9 –  3  -10 -4 + | -4  34  -14 |

**XC02 - TIỀN TỐ**

**Độ khó:**2

**Mô tả đề bài:**

Biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ + \* A B / C D tương ứng với biểu thức trung tố A \* B + C / D

Hãy viết chương trình tính toán giá trị của biểu thức tiền tố.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi số nguyên N (N ≤ 1000), là số lượng toán tử và toán hạng.

Dòng tiếp theo là biểu thức tiền tố với N toán tử và toán hạng. Các toán hạng và toán tử cách nhau bởi một dấu cách.

Các toán hạng có thể có nhiều chữ số, có thể là số âm. Phép chia thực hiện trên số nguyên. Input đảm bảo kết quả phép toán có giá trị tuyệt đối không vượt quá 10^18.

**Output:**

Với mỗi test, in ra giá trị biểu thức tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| Input: | Output |
| 3  7  - + 2 \* 3 1 9  7  - + 8 \* 7 5 9  3  + -10 -4 | -4  34  -14 |

**XC03 - HÌNH CHỮ NHẬT LỚN NHẤT**

**Độ khó:**2

**Mô tả đề bài:**

Cho bảng hình chữ nhật có kích thước N x M, các ô trong bảng chỉ nhận giá trị 0 hoặc 1.

Nhiệm vụ của bạn là hãy tìm hình chữ nhật chứa toàn số 1, các cạnh song song với trục tọa độ và có diện tích lớn nhất có thể.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi hai số nguyên N và M (N, M ≤ 1000).

N dòng tiếp theo, mỗi dòng gồm M số nguyên mô tả bảng.

**Output:**

Với mỗi test, in ra kích thước 2 cạnh của hình chữ nhật (cạnh nhỏ và cạnh lớn). Nếu có nhiều đáp án, in ra đáp án thỏa mãn hiệu của cạnh lớn và cạnh nhỏ là bé nhất có thể.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input:** | **Output** |
| 3  4 6  1 1 1 1 1 1  1 1 1 0 0 0  0 0 0 1 0 1  0 1 0 0 1 0  2 3  1 1 1  1 1 0  9 13  0 0 0 0 0 1 0 0 0 0 0 0 0  0 0 1 1 1 1 1 1 1 0 0 0 0  0 0 1 1 1 1 1 1 1 0 0 0 0  0 1 1 1 1 1 1 1 1 1 0 0 0  1 1 1 1 1 1 1 1 1 1 1 0 0  0 1 1 1 1 1 1 1 1 1 0 0 0  0 0 1 1 1 1 1 1 1 0 0 0 0  0 0 1 1 1 1 1 1 1 0 0 0 0  0 0 0 0 1 1 1 0 0 0 0 1 1 | 2 3  2 2  7 7 |

**XD01 - DI CHUYỂN K BƯỚC**

**Độ khó:**3

**Mô tả đề bài:**

Cho đồ thị vô hướng có N đỉnh dưới dạng ma trận kề. Các đỉnh được đánh số từ 1 đến N. Với mỗi cặp đỉnh S và T trong đồ thị, người ta muốn đếm số cách đi từ S đến T sao cho số cạnh đi qua đúng bằng K. Giả sử các đỉnh và các cạnh có thể được lặp lại nhiều lần trong đường đi.

**Input**

Dòng đầu tiên ghi 4 số N, S, T, K (1 ≤ N ≤ 50; 1 ≤ S, T ≤ N)

N dòng tiếp theo ghi ma trận kề của đồ thị.

* 50% số test ứng với 1 ≤ K ≤ 105
* 50% số test ứng với 1 ≤ K ≤ 1018

**Output**

In ra số tuyến đường thỏa mãn. Nếu kết quả quá lớn, hãy chia dư cho 109 + 7.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 1 1 4  0 1 0  1 0 1  0 1 0 | 2 |

**XD02 - ĐỒ THỊ CÓ HƯỚNG**

**Độ khó:**3

**Mô tả đề bài:**

Cho đồ thị có hướng với N đỉnh và M cạnh. Mỗi lần duyệt đồ thị, bạn có thể lựa chọn bắt đầu từ bất cứ đỉnh nảo và cố gắng đi qua nhiều đỉnh nhất có thể (đỉnh nào đã đi qua sẽ được đánh dấu). Hãy tính xem **cần ít nhất bao nhiêu lần duyệt**để đi qua tất cả các đỉnh của đồ thị.

**Input**

Dòng đầu ghi 2 số N và M ( 1 ≤ N ≤ 1000; 0 ≤ M ≤ 10000)

M dòng sau ghi các cạnh có hướng của đồ thị.

**Output**

Ghi ra một số nguyên duy nhất là số lần duyệt ít nhất tính được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 4  1 2  1 3  4 1  5 1 | 2 |

**XD04 - CÂY KHUNG NHỎ NHẤT**

**Độ khó:**3

**Mô tả đề bài:**

Bài toán cây khung nhỏ nhất trong đồ thị có trọng số đã rất quen thuộc. Ta xét phiên bản mới của bài toán này được phát biểu trên không gian tọa độ Oxy.

Cho N điểm trong không gian Oxy. Giả sử đã có M đoạn thẳng nối giữa một số điểm. Hãy tính tổng độ dài nhỏ nhất có thể của các đoạn thẳng cần bổ sung để đảm bảo các điểm đều có thể đi đến nhau qua các đoạn thẳng.

**Input**

Dòng đầu tiên ghi hai số N và M (0 < N,M ≤ 1000).

N dòng tiếp theo, mỗi dòng ghi tọa độ một điểm. Các giá trị tọa độ đều là các số nguyên không âm và không quá 106.

M dòng tiếp theo, mỗi dòng ghi một cặp (i,j) với ý nghĩa là điểm thứ i trong danh sách phía trên đã có đoạn thẳng nối đến điểm thứ j (chỉ số tính từ 1).

**Output**

Ghi ra giá trị tổng độ dài nhỏ nhất có thể của các đoạn thẳng cần bổ sung. Giá trị in ra 2 chữ số phần thập phân và không làm tròn.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 1  1 1  3 1  2 3  4 3  1 4 | 4.00 |

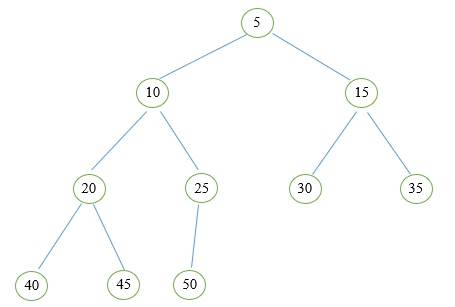
**XE03 - CÂY NHỊ PHÂN GẦN CÂN BẰNG - 1**

**Độ khó:**2

**Mô tả đề bài:**

Cho mảng A[] có N phần tử. Lấy phần tử A[0] làm gốc của cây, xây dựng cây nhị phân gần hoàn chỉnh với các phần tử còn lại A[1], A[2], …, A[N-1]. Nhập lần lượt từng mức của cây, mỗi mức gồm có 1, 2, 4, 8, … phần tử.

Ví dụ với mảng A[] = {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}, ta xây dựng được cây nhị phân như sau:



Hãy in ra kết quả phép duyệt inorder của cây nhị phân gần hoàn chỉnh này.

**Input:**

Dòng đầu tiên là số nguyên bộ test T (T ≤ 100).

Mỗi test bắt đầu số nguyên N (1 ≤ N ≤ 10 000).

Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 10 000).

**Output:**

Với mỗi test, in ra kết quả của phép duyệt **inorder**trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  10  5 10 15 20 25 30 35 40 45 50  10  1 2 3 4 5 6 7 8 9 10 | 40 20 45 10 50 25 5 30 15 35  8 4 9 2 10 5 1 6 3 7 |

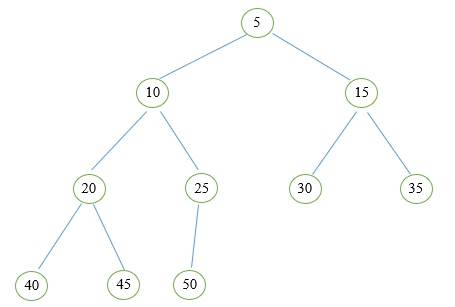
**XE04 - CÂY NHỊ PHÂN GẦN CÂN BẰNG - 2**

**Độ khó:**2

**Mô tả đề bài:**

Cho mảng A[] có N phần tử. Lấy phần tử A[0] làm gốc của cây, xây dựng cây nhị phân gần hoàn chỉnh với các phần tử còn lại A[1], A[2], …, A[N-1]. Nhập lần lượt từng tầng của cây, mỗi tầng gồm có 1, 2, 4, 8, … phần tử.

Ví dụ với mảng A[] = {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}, ta xây dựng được cây nhị phân như sau:



Hãy in ra kết quả phép duyệt **postorder** của cây nhị phân gần hoàn chỉnh này.

**Input:**

Dòng đầu tiên là số nguyên bộ test T (T ≤ 100).

Mỗi test bắt đầu số nguyên N (1 ≤ N ≤ 10 000).

Dòng tiếp theo gồm N số nguyên A[i] (0 ≤ A[i] ≤ 10 000).

**Output:**

Với mỗi test, in ra kết quả của phép duyệt **postorder**trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  10  5 10 15 20 25 30 35 40 45 50  10  1 2 3 4 5 6 7 8 9 10 | 40 45 20 50 25 10 30 35 15 5  8 9 4 10 5 2 6 7 3 1 |