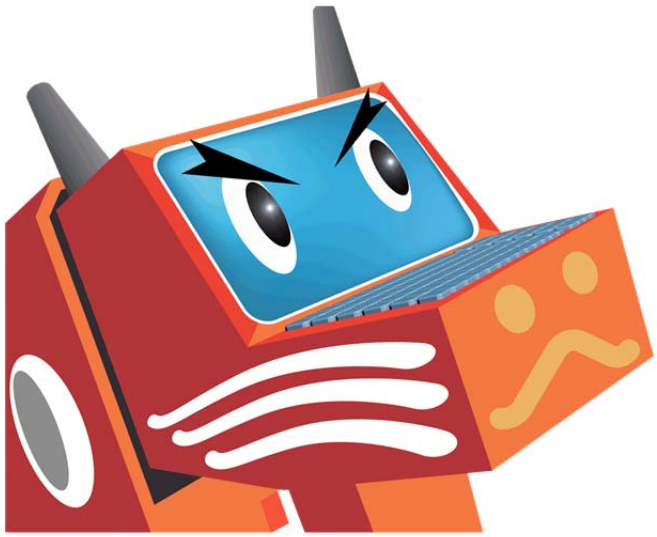
に・ぜろ・いち・ご

パソコン甲子園2015

全国高等学校パソコンコンクール

プログラミング部門 本選問題

平成２７年１１月７日(土)　午後１時４５分～午後５時４５分



全国高等学校パソコンコンクール実行委員会

問題１　直方体　　（６点）

アイヅ放送協会の教育番組(ＡＨＫ教育)では、子供向けの工作番組「あそんでつくろ」という番組を放送しています。今日は画用紙で箱を作る回ですが、用意した長方形の画用紙で直方体ができるかを確かめたいと思います。ただし、画用紙は切ったり折ったりしてはいけません。

課題

６つの長方形が与えられるので、それらを使って直方体が作れるかどうか判定するプログラムを作成せよ。

入力

入力は以下の形式で与えられる。

```
h1 w1
h2 w2
h3 w3
h4 w4
h5 w5
h6 w6
```

入力は６行からなり、それぞれの行に各長方形の縦の長さを表す整数h_i($1 \leqq h_i \leqq 1000$)と横の長さを表す整数w_i($1 \leqq w_i \leqq 1000$)が与えられる。

出力

直方体を作成できる場合には「yes」を、作成できない場合には「no」を出力する。ただし、立方体は直方体の一種なので、立方体の場合でも「yes」と出力する。

入出力例

入力例１	出力例１
2 2 2 3 2 3 2 3 2 2 3 2	yes

入力例２	出力例２
2 2 2 3 2 3 2 3 2 2 2 2	no

問題２　関連商品　（６点）

インターネット通販サイトでは、ユーザが現在見ている商品と同じページに、過去に他のユーザによって、現在見ている商品と一緒に買われた別の商品をいくつか表示してくれます。関連性の高いと思われる商品を提示することで、売り上げを伸ばすことができると考えられているからです。

似たようなことは、一緒に買われることが多い商品を近くに配置する、という工夫として、近所のスーパーマーケットでも目にすることができます（例えば、パンとジャムのような)。ここであなたに考えてもらいたいのは、商品配置の工夫を助けるプログラムを書くことです。今回は、ある基準となる回数を設定し、一緒に買われた回数が基準回数以上である、２つの商品の組み合わせを求めたいと思います。

課題

一緒に買われた商品の情報と基準回数が与えられたとき、基準回数以上一緒に買われた商品２つの組み合わせを出力するプログラムを作成せよ。

入力

入力は以下の形式で与えられる。

```
N F
info1
info2
:
infoN
```

１行目に、一緒に買われた商品の情報の数N(1≦N≦100)と、基準回数F(1≦F≦100)が与えられる。続くN行に、一緒に買われた商品の情報が与えられる。一緒に買われた商品の情報$info_i$は、以下の形式で与えられる。

```
M item1 item2 … itemM
```

M(1≦M≦10)は、この情報がいくつの商品を含むかを表す。$item_j$は、この買い物で買われた商品の名前であり、英小文字だけから成る長さ 1 以上 30 以下の文字列である。$info_i$の中に同じ商品が与えられることはない。

出力

１行目に基準回数以上一緒に買われた商品２つの組み合わせの数を出力し、２行目以降に組み合わせをすべて出力する。ただし、組み合わせが一つもない場合は２行目以降には何も出力しない。

出力の順番は、組み合わせ内の商品名どうしを、辞書式順序（英和辞書で単語が並んでいる順番）で並べたあと、組み合わせどうしについては以下のようにする。

- 一つ目の商品名どうしを比較して、辞書式順序で早いほうが先。
- 同じ場合は、二つ目の商品名どうしを比較して、辞書式順序で早いほうが先。

商品名はスペース一つで区切り、商品の組み合わせは改行一つで区切る。

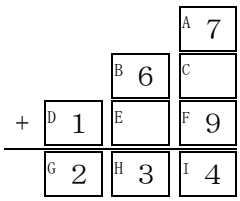
入出力例

入力例１	出力例１
5 2 3 bread milk banana 2 milk cornflakes 3 potato bread milk 4 cornflakes bread milk butter 2 potato bread	3 bread milk bread potato cornflakes milk

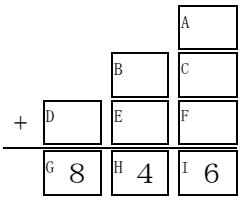
入力例２	出力例２
5 5 3 bread milk banana 2 milk cornflakes 3 potato bread milk 4 cornflakes bread milk butter 2 potato bread	0

問題３　虫食い算　（６点）

足し算は筆算で簡単に計算できますが、もし、いくつかの数字が欠けていたら、欠けている数字を埋めるのは簡単でしょうか？ 例えば、右のような筆算において、１から９の数字が一度しか現れないという条件があるとき、ＣとＥのマスに入る数字はいくつになるでしょう？ この場合、Ｃに入るのは８，Ｅに入るのは５が正解となります。このように、数字がいくつか欠けている演算を虫食い算と呼びます。



１から９の数字が一度しか現れないという条件はそのままで、右のようにもっとたくさんの数字が欠けていたら、正解となる数字の埋め方は一通りしかないのでしょうか？ 実は、必ず一通りに決まるとは限りません。



課題

上の図のような形をした虫食い算の、ＡからＩの各マスの情報が与えられたとき、正しい埋め方が何通りあるかを出力するプログラムを作成せよ。

入力

入力は以下の形式で与えられる。

```
Ａ Ｂ Ｃ Ｄ Ｅ Ｆ Ｇ Ｈ Ｉ
```

１行に、虫食い算のＡからＩのマスに入っている数字の情報が与えられる。ただし、与えられた値が-1のときは、そのマスの数字が欠けていることを表す。-1以外の値は、1から9の整数のいずれかで、それらの間に重複はない。

出力

正しい埋め方が何通りあるかを１行に出力する。

入出力例

入力例１	出力例１
7 6 -1 1 -1 9 2 3 4	1

入力例２	出力例２
7 6 5 1 8 9 2 3 4	0

入力例３	出力例３
-1 -1 -1 -1 -1 -1 8 4 6	12

入力例４	出力例４
-1 -1 -1 -1 -1 -1 -1 -1 -1	168

問題４　貴金属リサイクル　（６点）

会津特産の貴金属であるアイヅニウムをリサイクルするＰＣＫ社は、全国各地にネットワークを持ち、たくさんの回収車でアイヅニウムを集めてきます。この会社は、処理の効率化のために、塊（かたまり）の重さと個数の単位を規格で定めています。

塊の重さには「ボッコ」という単位を使います。ｘボッコのアイヅニウムの重さは2^xグラムです。宝石で例えると、「カラット」のようなものです。また、塊の個数には「マルグ」という単位を使います。ｙマルグは2^y個です。１箱に入っている品物の個数である「ダース」のようなものです。ただし、ｘとｙは０以上の整数でなければいけません。

回収車ｉは、a_iボッコの重さのアイヅニウムをb_iマルグずつ集めます。こうして集まったアイヅニウムを、炉の中に入れて溶かし、いくつかのアイヅニウムの塊を再生しますが、なるべくアイヅニウムの塊の数が少なくなるようにします。このとき、集めてきたアイヅニウムの重さの合計と、再生してできるアイヅニウムの重さの合計は変わりません。

課題

回収車が集めたアイヅニウムの塊のボッコ単位の重さとマルグ単位の個数が与えられたとき、再生後のアイヅニウムの塊の数が最小になるような結果を求めるプログラムを作成せよ。

入力

入力は以下の形式で与えられる。

```
N
a1 b1
a2 b2
 :
aN bN
```

１行目に、回収車の数N(1≦N≦100000)が与えられる。続くN行に、回収車iが回収したアイヅニウムの塊の、「ボッコ」単位の重さを表す整数a_i(0≦a_i≦100000)と「マルグ」単位の個数を表す整数b_i(0≦b_i≦100000)が与えられる。

時間制限

入力に対して、実行時間が３秒を超えてはならない。

出力

再生した後に得られるアイヅニウムの塊の数が最小になるような、ボッコ単位の重さとマルグ単位の個数を、重さの小さい順に出力する。

入出力例

入力例１	出力例１
3 2 1 1 3 2 2	3 0 5 0

入力例２	出力例２
1 100000 2	100002 0

問題５　完全平等二国間貿易　（８点）

サイバースペースにあるアイヅ国はワカマツ国と情報貿易を行っています。２つの国はお互いに有用なデータを交換することで経済発展を遂げています。博愛と平等、そして何よりも会津地方の古い言葉である、「ならぬことはならぬものです」を国是とする両国は、定期的に貿易状況の調査を行っています。

調査では、バイト単位でアイヅ国から見たデータ流入量から流出量を引いた値を、１ナノ秒ごとに求めた表が与えられます。その表から、値の総和が０になる最長の区間を見つけます。この区間が長いほど、平等性が保たれていると判断します。

課題

貿易状況が記録された表が与えられたとき、値の総和が０になる最長の区間の長さを求めるプログラムを作成せよ。

入力

入力は以下の形式で与えられる。

```
N
d_1
d_2
:
d_N
```

１行目に、表に書かれた値の数N($1 \leqq N \leqq 200000$)が与えられる。続くN行に、表のi行目に書かれた値を示す整数d_i($-10^9 \leqq d_i \leqq 10^9$)が与えられる。

時間制限

入力に対して、実行時間が３秒を超えてはならない。

出力

表から得られる、総和が０になる最長の区間の長さを１行に出力する。そのような区間が存在しない場合、「0」を１行に出力する。

入出力例

入力例１	出力例１
5 18 102 -155 53 32	3

入力例１では、２行目から４行目までの値の総和が０になるので、最長の区間の長さが3になる。

入力例２	出力例２
4 1 1 -1 -1	4

入力例２では、２行目から３行目の総和が０になるが、１行目から４行目までの値の総和も０になるので、最長の区間の長さが4になる。

問題６　プログラム停止判定　（１０点）

皆さんは、苦労して作ったプログラムを実行してみたら、無限ループになってしまった経験はありませんか？ プログラムの実行が停止するかどうかを、実行しなくても事前に判定できると便利ですよね。

残念ながら、皆さんがふだん使っているプログラミング言語では、あらゆるプログラムに対してそのような判定をすることは不可能です。しかし、それよりもはるかに計算能力の低いプログラミング言語なら、その言語で書いたプログラムが停止するかどうかを判定するプログラムを書ける場合があります。

TinyPowerというプログラミング言語を考えます。この言語のプログラムは行の並びです。プログラムの各行には、先頭に行番号を書き、その後ろに文を一つ書きます。この言語で書ける文の種類は以下の通りです。

文の種類	動作
ADD var_1 var_2 var_3	変数var_2の値とvar_3の値を加算した結果を変数var_1に代入する
ADD var_1 var_2 con	変数var_2の値と定数conを加算した結果を変数var_1に代入する
SUB var_1 var_2 var_3	変数var_2の値からvar_3の値を減算した結果を変数var_1に代入する
SUB var_1 var_2 con	変数var_2の値から定数conを減算した結果を変数var_1に代入する
SET var_1 var_2	変数var_2の値を変数var_1に代入する
SET var_1 con	定数conを変数var_1に代入する
IF var_1 dest	変数var_1の値が０でないときだけ、行番号destにジャンプする
HALT	プログラムを停止させる

行番号は正の整数で、プログラム中に同じ行番号が２つ以上現れることはありません。変数は英小文字一文字で表し、定数と変数の値は整数です。変数の宣言は不要で、変数の初期値は０です。

プログラムの実行は先頭の文から始まり、並んでいる順に文が実行されます。ただし、上の表に書かれたように、IF文の変数の値が０でないときは、変数の後ろに書かれた行番号で指定される行にジャンプし、その行に書かれた文から実行を続けます。プログラムは以下のときに停止します。

- HALT文を実行したとき。
- 負の整数または１６以上の整数を変数に代入しようとしたとき（変数の値は更新されない）。
- プログラムに現れない行番号にジャンプしようとしたとき。
- プログラムの最後の文を実行した後、そこからどの行にもジャンプしないとき。

課題

TinyPowerのプログラムが与えられたとき、それが停まるかどうかを判定するプログラムを作成せよ。

入力

入力は以下の形式で与えられる。

```
N
stmt1
stmt2
 :
stmtN
```

１行目にプログラムの行数N(1≦N≦50)が与えられる。続くN行に、TinyPowerプログラムの文$stmt_i$が与えられる。$stmt_i$は、以下のいずれかの形式で与えられる。

```
line ADD var1 var2 var3
```
または
```
line ADD var1 var2 con
```
または
```
line SUB var1 var2 var3
```
または
```
line SUB var1 var2 con
```
または
```
line SET var1 var2
```
または
```
line SET var1 con
```
または
```
line IF var1 dest
```
または
```
line HALT
```

line,dest(1≦line, dest≦1000)は行番号、var_j(英小文字１文字)は変数、con(0≦con≦15)は定数を表す。$stmt_i$中の区切りは空白１文字とする。なお、プログラム中に変数は必ず１つ以上現れ、異なる変数名は５つまでしか現れないものとする。

時間制限

入力に対して、実行時間が１０秒を超えてはならない。

出力

プログラムが停止するときは、プログラムに現れる変数の結果を、変数名の辞書順に改行区切りで出力し、停止しないときは「inf」を出力する。変数の結果は、変数名と変数の値を「=」で区切って出力する。

入出力例

入力例１	出力例１
6 10 SET c 1 20 SET i 5 100 ADD s s i 110 SUB i i c 120 IF i 100 200 HALT	c=1 i=0 s=15

入力例１は、１から５までの整数の和を計算し、その結果を変数sに格納したあと、HALT文の実行で停止する。

入力例２	出力例２
3 10 SET c 1 120 IF c 10 20 HALT	inf

入力例２は、行番号10でcに1を代入し、次の行番号120のIF文で行番号10に戻ることを繰り返すので、停止しない。

入力例３	出力例３
3 111 SET c 1 12 SUB c c 2 777 SET a 4	a=0 c=1

入力例３は、行番号111でcに1を代入し、次の行番号12でcに-1を代入しようとするので、停止する。このときcの値は-1に更新されない。行番号777は実行されないので、aの値は初期値0のままである。

問題７　スケジューラ　（１２点）

あなたはユニークなオペレーティングシステム「ウンズグネ１５」の開発に取り組んでおり、性能を決定付けるスケジューラの設計に頭を悩ませている。スケジューラとは、実行すべき処理をタスクという単位で表現し、それらをどの順序で実行するかを決定するプログラムである。スケジューラはタスクに１から N の番号をつけて管理する。全てのタスクは K 個の属性 $f_1, f_2, ..., f_K$ を持ち、各属性にはそれぞれ固有の値が設定されている。ただし、ある２つのタスクについて、対応する属性の値すべてが同じになることはない。

あるタスクには、そのタスクの実行を始める前までに実行を完了していなければならないタスクが与えられることがある。タスクＡがタスクＢの前に完了していなければならないことを「タスクＡ→タスクＢ」と表す。例えば、タスク１→タスク２、タスク３→タスク２という関係があれば、タスク２を処理する前にタスク１とタスク３の両方の処理が終わっていなければならない。このような関係をタスク間の依存関係という。ただし、あるタスクから依存関係をたどっていって、そのタスクにたどり着くことはない。

スケジューラは依存関係に従って、実行順序を決定する。しかし、依存関係だけでは順序が一通りに定まらない場合がある。そのような場合は、各タスクが持つ属性の値によって、次に処理するタスクを選択する。

ウンズグネ１５のタスクは属性を複数もつため、すべての属性の値を考慮して実行順序を決定する必要がある。そのために、属性を比較する順番を定める評価順序を用いる。評価順序が最も先の属性を比較し、その属性の値が最も大きいタスクを選択する。そのようなタスクが複数ある場合は、評価順序がその次の属性で比較し、以下同様な手順を繰り返す。例えば、以下の３つの属性を持つ３つのタスクについて考える。

タスク＼属性	f_1	f_2	f_3
X	3	3	2
Y	3	2	2
Z	3	1	3

評価順序が $f_1\ f_2\ f_3$、$f_2\ f_1\ f_3$、または $f_2\ f_3\ f_1$に設定されている場合は、タスク X が選ばれる。また、評価順序が $f_1\ f_3\ f_2$、$f_3\ f_1\ f_2$、または $f_3\ f_2\ f_1$に設定されている場合はタスク Z が選ばれる。

ウンズグネ１５のスケジューラの特徴は、属性の評価順序が途中で何度でも変更できることである。評価順序は、ある個数のタスクの実行が完了した時点で変更できる。ただし、スケジューラが最初に使う評価順序はあらかじめ決まっている。

課題

各タスクの属性の値、タスクの依存関係、評価順序の変更情報が与えられたとき、タスクを実行する順序を報告するプログラムを作成せよ。

入力

入力は以下の形式で与えられる。

```
N K
f_{1,1} f_{1,2} … f_{1,K}
f_{2,1} f_{2,2} … f_{2,K}
:
f_{N,1} f_{N,2} … f_{N,K}
D
a_1 b_1
a_2 b_2
:
a_D b_D
e_{0,1} e_{0,2} … e_{0,K}
R
m_1 e_{1,1} e_{1,2} … e_{1,K}
m_2 e_{2,1} e_{2,2} … e_{2,K}
:
m_R e_{R,1} e_{R,2} … e_{R,K}
```

１行目に、タスクの数N(2≦N≦50000)と、各タスクが持つ属性の数K(1≦K≦4)が与えられる。続くN行に、タスクiが持つ属性の値$f_{i,j}$(1≦$f_{i,j}$≦100000)が与えられる。続く１行に、依存関係の個数D(0≦D≦200000)が与えられる。続くD行に依存関係a_i→b_i(1≦a_i,b_i≦N)が与えられる。

続く１行に、最初の評価順序$e_{0,j}$(1≦$e_{0,j}$≦K)が与えられる。続く１行に、評価順序の変更回数R(0≦R＜N)が与えられる。続くR行に、評価順序の変更情報が与えられる。i回目の変更情報は、実行が完了したタスクの個数m_i(1≦m_i＜N)と評価順序$e_{i,j}$(1≦$e_{i,j}$≦K)からなり、全部でm_i個のタスクの実行が完了した時点で、評価順序を$e_{i,1}$，$e_{i,2}$,…，$e_{i,K}$に変更することを示す。

評価順序の変更情報は以下の条件を満たす。

- $e_{i,1}$，$e_{i,2}$，…，$e_{i,K}$中に同じ値は２つ以上現れない。
- i＜jのとき、m_i＜m_jである。

時間制限

入力に対して、実行時間が３秒を超えてはならない。

出力

スケジューラが処理する順番に、タスクの番号を出力する。

入出力例

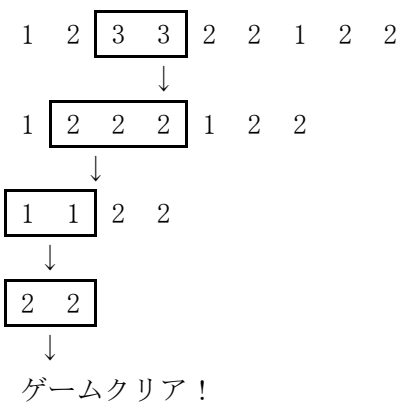
入力例１	出力例１
5 3 1 5 2 3 8 5 1 2 3 5 5 5 4 8 2 0 1 2 3 2 2 2 3 1 4 3 1 2	4 5 2 1 3

入力例２	出力例２
5 2 1 1 2 1 3 1 4 4 5 2 3 1 4 2 4 2 5 1 2 1 3 2 1	3 2 5 1 4

問題８　消える数列、消えない数列　　（１４点）

ただお君は頭の体操をするために、数列を使ったゲームをしています。このゲームでは、はじめに、１から９までの数字がランダムに並んだ列が与えられます。ただお君は、数列からその一部分を消していきます。ルールは、以下の通りです。

- 数列から、同じ数字が２つ以上並んでいる部分を適当に選ぶ。その部分を含み、連続して現れている同じ数字をすべて消す。
- 消した部分の右側に数列が残っていた場合は、それを左に詰めて、数列を１つにまとめる。
- 上の２つの操作を繰り返した結果、すべての数字が消えればゲームクリアとなる。



例えば、右上の図のような 1,2,3,3,2,2,1,2,2 という数列の場合、
左から数えて、３番目、４番目の３を消すと 1,2,2,2,1,2,2
左から数えて、２番目から４番目の２を消すと 1,1,2,2
左から数えて、１番目と２番目の１を消すと 2,2
左から数えて、１番目と２番目の２を消すと、ゲームクリアとなります。

ただし、どのように数字を消してもクリアできない数列があります。たとえば、1,2,3,3,1,2 や 1,2,3,1,2,3 などの数列です。短い数列であれば、ただお君でもクリアできるかどうかがすぐに分かり、クリアできないと分かれば違う数列にチャレンジできますが、長い数列になるとそう簡単にはいきません。

課題

与えられた数列が上のゲームをクリアできるかどうか判定するプログラムを作成せよ。

入力

入力は複数のデータセットからなる。入力の終わりはゼロ１つの行で示される。各データセットは以下の形式で与えられる。

```
N
c1 c2 … cN
```

１行目の N(1≦N≦100)は、数列の長さを表す整数である。２行目には１つの空白で区切られた N 個の整数 c_i(1≦c_i≦9)が与えられる。c_iは数列の i 番目の数字を示す。

データセットの数は 20 を超えない。

出力

上に示されたルールで数列を消すことができる場合は「yes」、できない場合は「no」を出力する。

入出力例

入力例	出力例
8 1 2 3 3 2 1 2 2 7 1 2 2 1 1 3 3 16 9 8 8 7 7 6 5 4 4 5 1 1 2 2 3 3 5 1 1 2 2 1 0	yes yes no yes

問題９　線分配置　　（１６点）

Ａ大学は今年もプログラミングコンテストを開催する。作題チームの一員であるあなたは、計算幾何学の問題の入力データの作成を担当することになった。あなたが作りたい入力データは、x 軸または y 軸に平行で、互いに触れ合うことのない線分の集合である。あなたは、次のアルゴリズムに基づいたデータ生成プログラムを開発して、入力データを生成する。

1. xy 平面上の線分の集合 T を空にする。
2. 次の処理を N 回繰り返す。
   - x 軸または y 軸に平行な適当な線分 s を作る。
   - s が T 内のどの線分にも触れない場合は s を T に追加し、触れる場合は s を追加しない。

課題

x 軸または y 軸に平行な N 本の線分を順番に入力し、各線分が平面上に追加されるかどうかを判定するプログラムを作成せよ。

入力

入力は以下の形式で与えられる。

```
N
px1 py1 qx1 qy1
px2 py2 qx2 qy2
:
pxN pyN qxN qyN
```

１行目に線分の数 N(1≦N≦100000)が与えられる。続く N 行に、i 番目に追加したい線分の情報が与えられる。各行に与えられる４つの整数 px_i, py_i, qx_i, qy_i $(0 \leqq px_i, py_i, qx_i, qy_i \leqq 10^9)$ は、それぞれ i 番目の線分の端点の x 座標、y 座標、もう一つの端点の x 座標、y 座標を表す。ただし、線分の長さは１以上である。

出力

各線分について、追加される場合「1」を、追加されない場合「0」を１行に出力する。

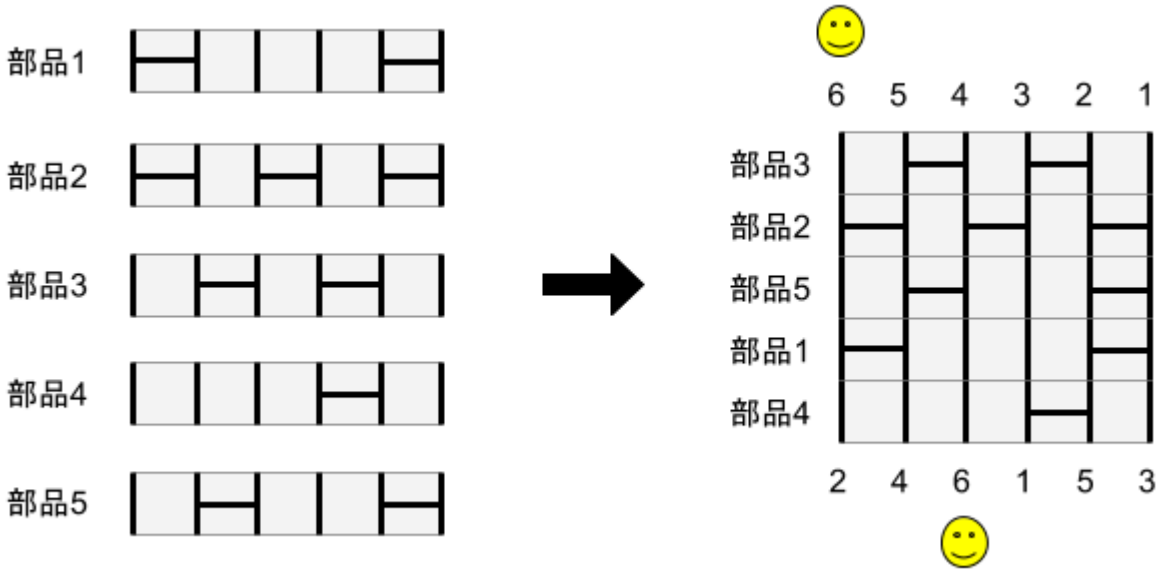
入出力例

入力例	出力例
9 0 2 5 2 1 3 1 7 0 6 3 6 2 4 8 4 4 0 4 5 6 3 6 0 5 6 7 6 8 3 8 7 6 5 11 5	1 1 0 1 0 1 1 0 1

問題１０　あみだくじ　　（１６点）

PCK 君はみんなでゲーム大会をしています。このゲーム大会では、大会の最後にあみだくじで順位を入れ替えます。大会には N 人のプレイヤーが参加しており、あみだくじには N 本の縦棒があります。

あみだくじは、図のように N-1 段の部品からできており、それぞれ 1 から N-1 の番号が割り当てられています。各部品は、あみだくじの一部を横方向に切り取った部分です。各部品にはいくつかの横棒が引かれていますが、部品の中の横棒はすべて同じ高さにあります。横棒同士がつながることはありません。



大会の最後に、順位の高い人から右から左の順に縦棒が割り当てられます。PCK 君は現時点で最下位なので、左端からスタートです。例えば、上図の組み立て方では、６位だった PCK 君は、このあみだくじによって４位（右から４番目の棒）に浮上することができます。

このゲームでは、最下位の人にあみだくじを組み立てる権利が与えられます。PCK 君はうまくあみだくじの部品の順番を決めて、逆転優勝を狙っています。ただし、部品を回転することはできません。

（※補足：あみだくじのたどり方について）
あみだくじのある縦棒の上端から出発して上から下へ進む。ただし、横棒がある地点ではその横棒でつながった別の縦棒に移動する。これを、縦棒の下端にたどり着くまで繰り返す。

課題

ゲームの参加人数とあみだくじの部品の情報を入力し、PCK 君が優勝できるかどうか判定するプログラムを作成せよ。優勝できる場合、そのあみだくじの部品の並びを１つ出力せよ。ただし、そのような並べ方が複数ある場合は、与えられた部品の番号で辞書順最小のものを出力せよ。

入力

入力は以下の形式で与えられる。

```
N
b1,1 b1,2 … b1,N-1
b2,1 b2,2 … b2,N-1
:
bN-1,1 bN-1,2 … bN-1,N-1
```

１行目に大会の参加者数N(2≦N≦500)が与えられる。続くN-1行にi番目の部品の横棒の情報が与えられる。$b_{i,j}$が1であるとき、i番目の部品の、左からj本目の縦棒からj+1番目の縦棒へ横棒が引かれていることを表す。$b_{i,j}$が0であるとき、i番目の部品の、左からj本目の縦棒からj+1番目の縦棒へ横棒は引かれていないことを表す。$b_{i,j}$が1であるとき、$b_{i,j+1}$が1となるような部品は与えられない。また、横棒の総数は10000を越えない。

時間制限

入力に対して、実行時間が３秒を超えてはならない。

出力

PCK 君が優勝できる場合，１行目に「yes」と出力する。続く N-1 行に、あみだくじの上から順に、部品の番号の並びを出力する。そのような並びが複数ある場合、辞書順最小である並びを出力する。PCK 君が優勝できない場合、１行に「no」と出力する。

入出力例

入力例１	出力例１
6 1 0 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0 1 0 0 1 0 0 1	yes 1 3 2 4 5

入力例２	出力例２
5 0 1 0 1 0 1 0 1 1 0 1 0 1 0 0 1	yes 4 1 3 2

４ １ ３ ２ と ４ ２ ３ １ の２通りの組み立て方が可能だが、辞書順で小さい方の ４ １ ３ ２ を出力する。

入力例３	出力例３
5 1 0 0 1 0 1 0 1 1 0 0 0 0 1 0 1	no