

入力

入力は以下の形式で与えられる。

```
N
r1 t1
r2 t2
:
rN tN
```

1行目に友人の数 N ($1 \leq N \leq 50$) が与えられる。続く N 行に、友人の位置が与えられる。友人の位置 r_i ($100 \leq r_i \leq 500$) はオーディオシステムからの距離を表す整数、 t_i ($0 \leq t_i \leq 180$) は円弧に沿って反時計回りに測った角度を表す整数である。

出力

各友人の位置について、計測点の番号を昇順で1行に出力する。

入出力例

入力例	出力例
4	23
300 120	18 23
300 105	17 18 22 23
250 105	17 18
250 90	

問題2 対称3進数

(6点)

1グラム、3グラム、9グラム、27グラムのおもりが1つずつあれば、天びんを使って1グラムから40グラムまで1グラム刻みで量れることが知られています。たとえば、天びんの一方の皿に重さを量りたいものと3グラムのおもりを載せ、もう一方の皿に27グラムと1グラムのおもりを載せて釣り合えば、量りたいものの重さは $27-3+1=25$ グラムだとわかります。さらに、 $1(=3^0)$ グラム、 3^1 グラム、 \dots 、 3^{n-1} グラム、 3^n グラムまでのおもりが1つずつあれば、天びんを使って $(3^{n+1}-1)/2$ グラムまで量れることが知られています。また、天びんが釣り合うようなおもりの置き方は一通りしかないことも知られています。

量りたいものとおもりを天びんに置いて、釣り合うようなおもりの置き方を文字列で表すことができます。 3^i グラムのおもりを量りたいものと同じ皿に載せたときは「-」、もう一方の皿に載せたときは「+」、どちらにも載せなかったときは「0」を文字列の右端から*i*番目に書きます(右端を0番目と数えます)。たとえば、先ほどの25グラムの例は +0-+ と表わせます。

それでは、量りたいものの重さが与えられたとき、天びんが釣り合うようなおもりの置き方を表す文字列を出力するプログラムを作成してください。ただし、3のべき乗グラムのおもりは、どのような重さのものでも必ず1つあるものとします。

(補足：対称3進数について)

量りたいものの重さが*w*のとき、おもりの置き方を表す文字列は*w*の対称3進数になっています。対称3進数とは、3のべき乗の数で位取りを行い、各位に数 1、0、-1 を表す文字を書くことで表した数のことです。上の文字列では、文字「+」、「0」、「-」がそれぞれ数 1、0、-1 に対応します。たとえば、25グラムのものを量るときのおもりの置き方が +0-+ である対称3進数が表す数は、 $1 \times 3^3 + 0 \times 3^2 - 1 \times 3^1 + 1 \times 3^0 = 25$ となります。

入力

入力は以下の形式で与えられる。

w

$w(1 \leq w \leq 100000)$ は量りたいものの重さを表す整数である。

出力

おもりの置き方を表す文字列を出力する。ただし文字列の左端を0にしてはならない。

入出力例

入力例 1	出力例 1
25	+0-+

入力例 2	出力例 2
2	+-

入力例 3	出力例 3
5	+-

問題3 ニッシン館マラソン部

(6点)

アイズ市立ニッシン館は、文武両道を掲げる歴史ある学校です。ニッシン館マラソン部は熱血指導で有名であり、決められた練習時間の間、周回コースをひたすら走るといふ鬼の特訓を行います。このとき、脱水症状で倒れる部員がいないように、部員の実力に合わせて飲み物を提供する給水所を設けています。

各部員が1単位時間あたりに走れる距離(ペース)は部員ごとに決まっており、全員が1単位時間ごとに必ず給水できるように給水所を設けたうえで特訓を行います。部員は1単位時間で着いた給水所で必ず中身の入った容器を取り、そのまま走り続けます。さらに、次の1単位時間で着いた給水所で空の容器を置き、中身の入った容器を取って走り続けるということを繰り返します。また、空の容器は、置いてから1単位時間の時点で飲料が補充され、その瞬間から誰にでも提供できるようになります。

部員は全員同じ地点から、容器を持たずに出発します。決められた練習時間まで走ると練習を終えます(このときも給水が必要です)。複数の部員が同時に同じ給水所に到着する場合もあるので、一か所の給水所で同時に複数の容器が必要になることもあります。

鬼の特訓を安全に行うために、最低いくつの給水容器が必要になるか求めるプログラムを作成してください。

入力

入力は以下の形式で与えられる。

N R T
p_1
p_2
:
p_N

1行目に部員の数 $N(1 \leq N \leq 100)$ 、周回コースの長さ $R(1 \leq R \leq 1000)$ 、決められた練習時間 $T(1 \leq T \leq 1000)$ が与えられる。続く N 行に、部員 i のペース $p_i(1 \leq p_i \leq 1000)$ が与えられる。

出力

最低いくつの給水容器が必要になるかを1行に出力する。

入出力例

入力例1	出力例1
1 10 20 1	11

入力例2	出力例2
2 5 12 1 2	8

問題4 デッドロックを検出せよ

(8点)

コンピュータにおける「データベース」とは、情報を管理するための入れ物で、「データベースマネジメントシステム(DBMS)」とは、その管理をする仕組みです。複数のユーザが利用するデータベースでは、DBMSは慎重に構築する必要があります。

例えば、倉庫から商品を1つ取り出した人が、データベースに対して以下の処理を行うとしましょう。

- (1) 商品の個数Nをデータベースから読む。
- (2) 新たな商品の個数N-1をデータベースに書き込む。

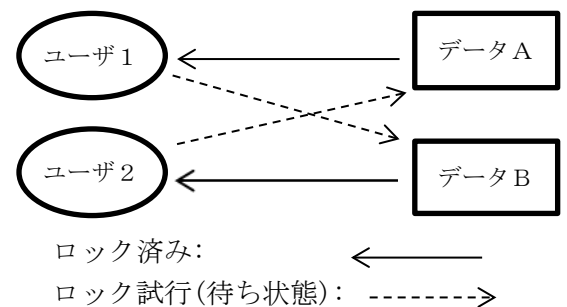
ユーザ1が(1)を終えて(2)を始める前に、別のユーザ2が倉庫から商品を取り出して(1)を行ったとします。ユーザ2もユーザ1と同じ個数を読むので、2人が(2)を終えたときには商品は2個減るのにデータベース上では1個しか減らないというおかしな結果になります。このような問題を防ぐために、DBMSは特定のデータを操作中のユーザに、そのデータを「ロック」する権利を与えます。ロックされていれば、他のユーザはその値を操作できなくなるので、おかしな結果を返すことはなくなります。

これで安全に操作できることは保証されますが、今度は別の問題が起こります。例えば、ユーザ1とユーザ2が以下のような順番でデータAとBをロックしようとしたらどうなるでしょうか？

- ① ユーザ1がデータAのロックを試みる → 成功(データAがロック中になる)
- ② ユーザ2がデータBのロックを試みる → 成功(データBがロック中になる)
- ③ ユーザ1がデータBのロックを試みる → データBがロック中なのでユーザ1は待つ
- ④ ユーザ2がデータAのロックを試みる → データAがロック中なのでユーザ2は待つ

④を実行した時点では、ユーザ1がA、ユーザ2がBをロックしているので、③④は永久に成功しません。これを「デッドロック」と呼びます。DBMSはこれを検出しなければなりません。

ある時点でデッドロックが起きているかどうかは、その時点でのすべてのユーザとデータの依存関係を書き、循環ができていないかどうかで判断できます。依存関係は、ユーザがデータをロック済みの場合はデータからユーザの向きに矢印を、ユーザがデータのロックを試行していて待ち状態になっている場合はユーザからデータの向きに矢印を書くことで表します。



上の①から④の例であれば、④を実行した時点での依存関係は右上のような図になります。このとき、矢印の方向に進むと、ユーザ1→データB→ユーザ2→データA→ユーザ1という循環ができていないため、デッドロックが起きていることがわかります。DBMSの苦労を体験するため、あなたにここでやってもらいたいのは、このようなデッドロックを検出するプログラムを作成することです。

入力

入力は以下の形式で与えられる。

```
N
rel1
rel2
:
relN
```

1行目にユーザとデータの依存関係の数 N ($1 \leq N \leq 1000$) が与えられる。続く N 行に、ユーザとデータの依存関係 rel_i が与えられる。依存関係はlockとwaitの2種類あり、各 rel_i は以下のいずれかの形式で与えられる。

```
u lock d
```

または

```
u wait d
```

$u \text{ lock } d$ は、ユーザ u ($1 \leq u \leq 100$) が、データ d ($1 \leq d \leq 100$) をロック済みであることを表す。

$u \text{ wait } d$ は、ユーザ u ($1 \leq u \leq 100$) が、データ d ($1 \leq d \leq 100$) のロックを試行していて待ち状態であることを表す。

ただし、入力は以下の条件を満たしていると仮定してよい。

- ロックされていないデータに対して、ユーザが待ち状態ではない。
- 二人以上のユーザにロックされているデータはない。
- ユーザ自身がロックしているデータに対して、待ち状態ではない。
- 同じ依存関係は与えられない。

出力

デッドロックが発生しているなら1、発生していないなら0を1行に出力する。

入出力例

入力例 1	出力例 1
4 1 lock 1 2 lock 2 1 wait 2 2 wait 1	1

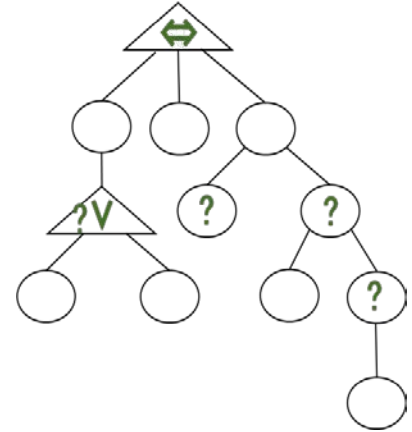
入力例 2	出力例 2
4 3 lock 3 2 wait 3 3 lock 4 2 wait 4	0

問題5 新薬開発

(9点)

英世博士は日々研究を行い、新しい薬を開発しようとしています。新薬を開発するためには、色々な物質を組み合わせることで薬を作り試験を行い、良い薬を見つけなければなりません。様々な組み合わせを試していくうちに、英世博士は物質の組み合わせが樹形図で表せることを突き止めました。

右の図は、薬の調合を表す樹形図の例です。図の中で丸く囲まれたものを物質ノード、三角で囲まれたものを選択ノードと呼びます。物質ノードは物質を表します。選択ノードは、物質の選択を表すもので、それ自体は物質を表しません。選択ノードには or 型 (V が付いたもの) と alt 型 (⇔ が付いたもの) の 2 種類があります。また ? が付いたノードは、それがオプションであることを表します。ただし、選択ノードの子ノード (下向きの枝の先にあるノード) がオプションになることはありません。樹形図に現れる異なる物質ノードは、それぞれ別の物質を表すものとします。



薬の調合を行うときは、樹形図の一番上のノードからはじめて、順々にノードをたどっていきながら以下のようにしてノードを選んでいきます。

- たどり着いたノードがオプションでない物質ノードなら、それを必ず選ぶ。
- オプションの物質ノードなら、それを選ぶかどうかは任意。
- or 型の選択ノードなら、その子から少なくとも一つを選ぶ。ただし、その選択ノードがオプションなら、子を一つも選ばなくてもよい。
- alt 型の選択ノードなら、その子から一つだけを選ぶ。ただし、その選択ノードがオプションなら、子を選ばなくてもよい。

あるノードが選ばれたときだけ、そのノードから下に向かう枝をそれぞれたどっていきます。選ばれなければ、それらをたどることはありません。

あなたは英世博士から、薬の物質の組み合わせを表す樹形図を受け取り、組み合わせの数が全部で何通りあるか求めるよう指示されました。樹形図が与えられたとき、組み合わせの総数を出力するプログラムを作成してください。

入力

入力は以下の形式で与えられる。

```
N
node1
node2
:
nodeN
s1 t1
s2 t2
:
sN-1 tN-1
```


1行目にノードの数 N ($1 \leq N \leq 1000$) が与えられる。続く N 行に、 i 番目のノードの情報 $node_i$ が与えられる。1番目のノードを樹形図の一番上のノードとする。続く $N-1$ 行に s_i 番目のノードからその子である t_i 番目のノード ($1 \leq s_i \neq t_i \leq N$) へ向かう枝が与えられる。 t_1 から t_{N-1} までには、2から N までの数が一度だけ現れる。

ノードの情報は以下の形式である。

- E オプションでない物質ノード。
- E? オプションである物質ノード。
- type オプションでない選択ノード。typeはAかRのいずれかで、Aはalt型、Rはor型を表す。
- type? オプションである選択ノード。typeの形式は同上。

入力から得られる樹形図は、以下の条件を満たす。

- 選択ノードは2つ以上の子ノードを持つ。
- 選択ノードの子ノードはオプションでない。
- 樹形図の一番上のノードはオプションでない。
- どのノードについても、子ノードの数は10を超えない。

出力

与えられた樹形図から得られるすべての組み合わせの総数を1行に出力する。ただし、出力すべき値は非常に大きくなりうるので、代わりに1,000,000,007で割った余りを出力する。

入出力例

入力例 1	出力例 1
12 A E E E R? E? E? E E E E? E 1 2 1 3 1 4 2 5 4 6 4 7 5 8 5 9 7 10 7 11 11 12	11

入力例 2	出力例 2
10 E R? E R E E A E E E 1 2 1 7 2 3 2 4 4 5 4 6 7 8 7 9 7 10	24

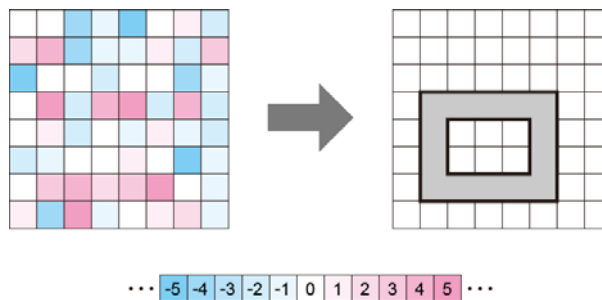
問題6 枠

(12点)

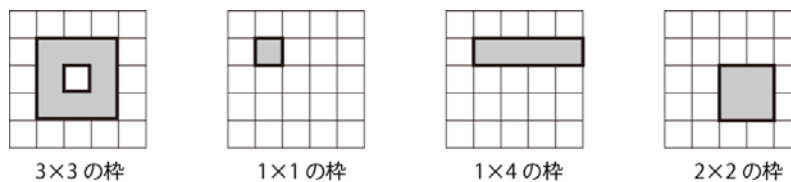
画像の中から有益な情報を抽出する画像認識はコンピュータサイエンスの中でも重要な研究テーマのひとつです。デジタルカメラ、運転支援システム、防犯システムなどに幅広く応用されています。

このような研究のおかげで、私たちは画像解析を行うための多くのソフトウェアやプログラム集を使い様々な処理を行うことができます。一方、自力でプログラムを書いて解析することで、その仕組みを知り、楽しい時間を過ごすことができます。ここでは、一風変わった画像認識を試みましょう。

画像として次のような各ピクセルが整数の値を持つ $N \times N$ のピクセルが入力として与えられます。この画像の中から、線の太さが1ピクセルの長方形の枠（わく）を1つ抽出します。



枠が覆うピクセルの値の和が最大となるような枠を抽出して、その和を報告するプログラムを作成して下さい。ただし、下の図のように、縦、横のピクセル数が1つや2つの場合も枠とみなすものとします。



入力

入力は以下の形式で与えられる。

```
N
p1,1 p1,2 ... p1,N
p2,1 p2,2 ... p2,N
:
pN,1 pN,2 ... pN,N
```

1行目に縦と横のピクセル数 N ($1 \leq N \leq 300$) が与えられる。続く N 行に、 i 行 j 列目のピクセルの値を表す整数 $p_{i,j}$ ($-1000 \leq p_{i,j} \leq 1000$) が与えられる。

時間制限

入力に対して、実行時間が1秒を超えてはならない。

出力

ピクセル値の和が最大となるような枠の、ピクセル値の和を1行に出力する。

入出力例

入力例 1	出力例 1
5 2 0 0 2 0 0 1 0 2 0 0 0 0 -1 0 0 4 0 3 0 -1 0 0 1 0	12

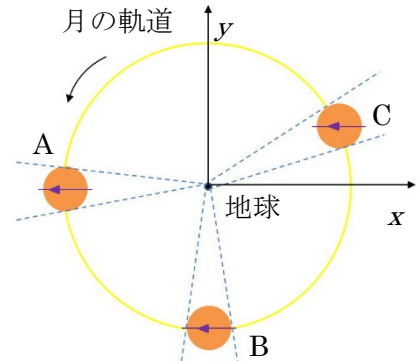
入力例 2	出力例 2
3 0 0 0 0 -1 0 0 0 0	0

問題7 かぐや

(12点)

いよいよ今月末に「はやぶさ2」が打ち上げられます。4年前に「はやぶさ」が帰ってきたときは、日本中が盛り上がったのを覚えている人も多いと思います。7年前には、「かぐや」が打ち上げられ、月の周りを回りながらたくさんの鮮明な画像を地球に送ってくれました。

右の図は、地球を原点とした空間座標（z軸は紙面から垂直に下から上に向いているものとします）に、月の軌道といくつかの月の位置、月の周りを回るかぐやの軌道を描いたものです。月の軌道は、x軸とy軸を通る平面上にある、原点を中心とする円とします。月の周りを回るかぐやの軌道は、x軸とz軸を通る平面に平行な平面上にある円とし、その中心は月の中心と一致します。月は、その軌道に沿って描かれた矢印の方向に回っているものとします。



右の図で、月の位置はA, B, Cの3つが描かれています。月を横切る直線がかぐやの軌道です。かぐやは月の周りを回っているため軌道は円ですが、z軸の正の方向から見ているので、図のx軸に平行な直線に見えます（月の位置が変わっても、つねにx軸に平行であることを注意してください）。かぐやは、その軌道上に描かれた矢印の方向に回っているものとします。

かぐやが地球から見て月の裏側に隠れてしまうと、地球と直接通信ができなくなります。かぐやの管制を担当するあなたは、かぐやが月の裏側に隠れる時間が、与えられた時間の中でどれだけになるかをプログラムで求めようとしています。

月の地球に対する位置と分単位での時間 t が与えられたとき、その位置から始めて t 分後までの間に、かぐやが月の裏側に隠れる時間を求めるプログラムを作成してください。ただし、地球とかぐやは点、月は半径1800kmの球と考えます。月は半径380000kmの軌道を2500000秒で一周し、かぐやは月の表面から高度100kmの円上を2時間で一周するものとします。かぐやの最初の位置は、かぐやの軌道がz座標で最大の値をとる位置とします。

入力

入力は以下の形式で与えられる。

m t

m ($0 \leq m < 360$)は月の位置を上図のx軸の正の部分からy軸の正の部分に向かって反時計回りに測った角度を整数で表したものである。 t ($1 \leq t \leq 10000$)は分で測った時間を表す整数である。

出力

はじめの位置から t 分経過するまでの間に、かぐやが月の裏側に隠れる時間（分）を実数で出力する。ただし、誤差がプラスマイナス1.0分を超えてはならない。この条件を満たせば小数点以下何桁表示してもよい。

入出力例

入力例 1	出力例 1
90 10	0.0

入力例 2	出力例 2
0 120	47.73

問題8 ネットカフェ

(12点)

あなたはネットカフェを経営しています。今日あなたは、顧客に指摘され続けている問題を解決しようと取り組んでいます。その問題とは、店舗の本棚の単行本が巻数順に並んでおらず、目的の単行本を探しだすのが面倒だという苦情です。

あなたの店舗で一番巻数の多い単行本は「名探偵 赤ベコ」（通称「赤ベコ」）です。あまりに長編なので、特別な本棚を「赤ベコ」のために用意しました。

単行本の各巻の重さと厚さは様々で、本棚の各段の幅と、各段に並べることができる本の重さの上限も様々です。あなたは、次の条件を満足するように本棚に本を並べることになりました。

- 1巻からある巻までの「赤ベコ」が本棚に並んでいる。
- それぞれの段には、巻数順に（途中で抜けている巻がないように）本が並ぶ。
- 各段に並べる本の重さの合計が、その段で定められた重さの上限を超えない。
- 各段に並べる本の厚さの合計が、その段の幅を超えない。

これらの条件を満たしたとき、この本棚に最大で何巻まで「赤ベコ」を並べることができるかを求めるプログラムを作成してください。

入力

入力は以下の形式で与えられる。

```
M N
w1 t1
w2 t2
:
wM tM
c1 b1
c2 b2
:
cN bN
```

最初の1行に「赤ベコ」の巻数 M ($1 \leq M \leq 200000$) と本棚の段数 N ($1 \leq N \leq 15$) が与えられる。続く M 行に、「赤ベコ」の単行本 i 巻目の重さ w_i ($1 \leq w_i \leq 100$) と厚さ t_i ($1 \leq t_i \leq 100$) を表す整数が与えられる。続く N 行に、本棚の i 段目の重さの上限 c_i ($1 \leq c_i \leq 10^8$) と幅 b_i ($1 \leq b_i \leq 10^8$) を表す整数が与えられる。

時間制限

入力に対して、実行時間が2秒を超えてはならない。

出力

本棚に並べることができる最大の「赤ベコ」巻数を1行に出力する。

入出力例

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

入力例 2	出力例 2
2 2 1 2 2 1 2 1 2 1	0

入力例 3	出力例 3
3 2 1 2 2 2 2 1 3 3 2 2	2

入力例 4	出力例 4
3 2 1 2 2 1 2 2 2 2 3 3	3

問題 9 未知の病原菌

(14点)

英世博士は未知の病原菌を発見しました。この病原菌は、アクダマキンとゼンダマキンと呼ばれる二種類の菌が、一直線に連なった鎖状の構造をしています。人類のために、この病原菌を無害化したいと考えています。

この病原菌は、長さが2以下になると力が弱まり、免疫力によって無害化されることが分かっています。英世博士は、この病原菌を任意の場所で切断して、前半と後半の2つの鎖にすることができます。また、2つの鎖を連結して1つの鎖にすることもできます。

しかし注意しなければいけないのは、アクダマキンの数が多い鎖はきわめて有害だということです。ある鎖においてアクダマキンの数がゼンダマキンの数よりも多くなると、その瞬間アクダマキンは無制限に増殖を始めます。これは長さ2以下の鎖についても例外ではないので、慎重に鎖を切断していかなければなりません。

どの瞬間においてもアクダマキンの数の多いような鎖を作ることなく、一本の鎖を長さ2以下にして無害化することは可能でしょうか。英世博士は、助手であるあなたに無害化が可能かどうか判定するプログラムを作成するよう指示しました。無害化が可能ならばその操作を出力し、不可能ならば不可能であると出力するプログラムを作成してください。ただし、その操作のステップ数が最小である必要はありません。

入力

入力は以下の形式で与えられる。

```
Q
str1
str2
:
strQ
```

1行目に病原菌の数 Q ($1 \leq Q \leq 300$) が与えられる。続く Q 行に各病原菌の初期状態を表す 'o' (ゼンダマキン) および 'x' (アクダマキン) からなる1つの文字列 str_i が与えられる。文字列 str_i の長さは1以上100以下である。

出力

str_i について、要求を満たす切断・結合操作の列が存在する場合、出力の最初の行にはその操作列の長さを表す整数 n を出力し、続く n 行に操作の内容を1行に1操作ずつ、最初の操作から順に出力する。

切断操作は以下の形式で表すこと。

```
split a p
```

a は切断する鎖の識別番号を表す整数であり、 p はその鎖を切断する位置である。この操作の結果、鎖 a は先頭から p 番目 (先頭を0から始める通し番号) の菌の直後で切断される。新しくできる2つの鎖のうち、前半のものに識別番号 $m+1$ が、後半のものに識別番号 $m+2$ が付与される (ここで、 m はこれまでに付与された最も大きな識別番号を表す)。また、鎖 a は消滅する。

結合操作は以下の形式で表すこと。

```
join a b
```

a, b は結合する鎖の識別番号を表す整数である。この操作の結果、鎖 a の末尾に鎖 b の先頭を結合した新しい鎖が作られる。新しく作られた鎖には、識別番号 m+1 が付与される(ここで、m はこれまでに付与された最も大きな識別番号を表す)。また、鎖 a, b は消滅する。

入力として与えられる最初の鎖には、識別番号 0 が付与されている。

操作の結果、問題の要求を満たすように鎖が分解されていた場合、どのような操作でも正答と判定される。操作列の長さも必ずしも最短である必要はない。ただし、操作列の長さは 20000 以下でなければならない。データセットにおいて、鎖が分解可能な場合、必ずこの条件を満たす操作列が存在することが保証される。

不正な操作列が出力された場合、誤答と判定される。不正な操作列には、以下の場合が含まれる。

- 切断操作 split a p において、 $0 \leq p < (\text{鎖 } a \text{ の長さ}) - 1$ が満たされていない場合。
- 切断・結合操作の対象となる鎖の識別番号がまだ生成されていないものである場合や、既に別の操作の対象となったため消滅している場合。
- 結合操作 join a b において、a と b が等しい場合。

要求を満たす切断・結合操作の列が存在しない場合、“-1”と出力する。

入出力例

入力例	出力例
6 o o o x x x x o o o o x x x o x x o o x x o o o x x o o o o o	-1 7 split 0 0 join 2 1 split 3 4 split 4 0 join 7 6 split 8 2 split 9 0 3 split 0 1 split 2 1 split 4 1 -1 0 1 split 0 0

例えば、入力例の 2 番目の病原菌 o o o o x x x は、split 0 0 により o(1) と o o o x x x(2) ができる。ここで、()内の数字は識別番号を表す。

join 2 1 により oooxxxo(3) ができ 1 と 2 は消滅する。

split 3 4 により oooxx(4) と xo(5) ができる。このとき { oooxx(4), xo(5) } の鎖が存在する。

split 4 0 により o(6) と oox(7) ができる。{ xo(5), o(6), oox(7) }

join 7 6 により ooxxo(8) ができる。{ xo(5), ooxxo(8) }

split 8 2 により oox(9) と xo(10) ができる。{ xo(5), oox(9), xo(10) }

split 9 0 により { xo(5), xo(10), o(11), ox(12) } となって終了する。

問題 10 アカベコ国王の配慮

(15点)

アカベコ国の国王には2人の王子がいます。国王は自分が退位するとき国を2つに分割し、それぞれの王子に一つずつ国を治めさせることにしました。新しい国の名前はアカ国とベコ国です。アカベコ国には N 個の町と、2つの町を繋ぐ M 本の道があります。国王は、以下の手順でアカベコ国の町と一部の道を2つの国に配分することにしました。

- ① 町を2つ選び、それぞれアカ国とベコ国に配分する。
- ② すでに配分された町 s を選ぶ。さらに、町 s から1本の道で繋がっている、まだ配分されていない町 t を選ぶ。そして、町 s 、 t 間の道と町 t を、町 s が配分された国に配分する。
- ③ ②を、行えなくなるまで繰り返す。

実は2人の王子はあまり仲が良くないので、国王は2つの国の距離をなるべく大きくしたいと考えています。ここで、2つの国の距離とは、アカ国の町とベコ国の町を繋ぐ道の中で、最も短い道の長さです。

アカベコ国の町と道の情報が与えられたとき、分配後のアカ国とベコ国の距離の最大値と、そのような距離になる配分が何通りあるかを求めるプログラムを作成してください。ただし、2つの配分結果は、アカ国とベコ国に異なる町か道が配分された場合に区別されます。

入力

入力は以下の形式で与えられる。

```
N M
s1 t1 d1
s2 t2 d2
:
sM tM dM
```

1行目は2つの整数からなる。 N ($2 \leq N \leq 100$)は町の数、 M ($N-1 \leq M \leq N(N-1)/2$)は道の数を表す。続く M 行に2つの町を繋ぐ道が与えられる。 s_i と t_i ($1 \leq s_i \neq t_i \leq N$)は i 番目の道が繋ぐ2つの町の番号を表す。 d_i ($1 \leq d_i \leq 10^9$)は i 番目の道の長さを表す。

入力は以下の条件を満たす。

- どの2つの町もいくつかの道を使い行き来が可能である。
- どの2つの町の間にも2本以上の道はない。
- 同じ長さの道は5本以下である。

出力

分配後のアカ国とベコ国の距離の最大値と組み合わせの数を、空白区切りで1行に出力する。ただし、分配後の組み合わせの数は非常に大きくなりうるので、代わりに1,000,000,007で割った余りを出力する。

入出力例

入力例	出力例
6 7 1 2 1	7 18

2 3 2	
3 1 3	
4 5 4	
5 6 5	
6 4 6	
1 4 7	