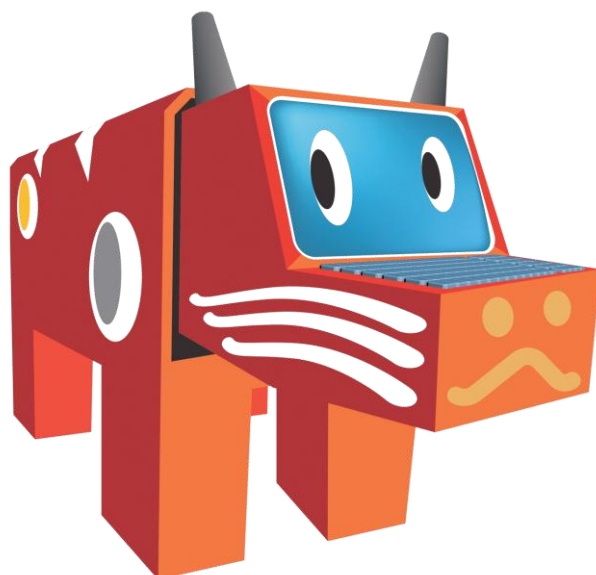


に・ゼロ・いち・に パソコン甲子園2012

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24年11月10日(土) 午後1時50分～午後5時50分



10th
Anniversary

全国高等学校パソコンコンクール実行委員会

解答にあたっての注意事項

特に断りがない限り、以下のことを前提とします。

1. 入力

- ・ 入力フォーマットは、問題中の入力および入力例を参考にしてください。例えば、入力例に
1
2
とあった場合、キーボードの「1」を押して、「Enter」キーを押し、「2」を押して「Enter」キーを押して入力するという意味です。
- ・ 入力データに全角文字はありません。
- ・ 行頭行末に空白は含まれません。
- ・ 不正なデータは入力されませんので、エラー処理を記述する必要はありません。
- ・ 同じ行に複数のデータが与えられる場合、それぞれのデータは半角空白1文字で区切られています。
- ・ 特に断りがない入力データは整数です。

2. 出力

- ・ 出力フォーマットは、問題中の出力及び出力例を参考にしてください。例えば、出力例に
1
2
3
とあった場合、1行目に1を、2行目に2を、3行目に3を出力するという意味です。
- ・ 問題で求められているとおりに出力してください。余計な文字(英数字、記号、空白、改行)を出力した場合や、必要な空白や改行を出力しない場合も不正解となります。
- ・ 出力に全角文字は含めないでください。

3. 入出力例について

- ・ 各問題には、入出力例が示されていますので、よく確認してから解答してください。

問題1 アカ・ベコと40人の盗賊

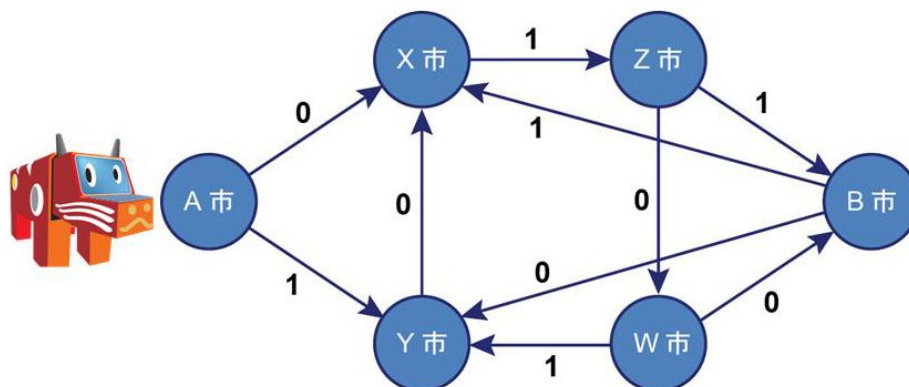
(7点)

(※この問題は審査員特別賞対象問題です)

40人の盗賊から逃れようとしているアカ・ベコは、A市の街中で道に迷ってしまった。アカ・ベコは新しいアジトがあるB市に行きたいのだが地図を盗賊に盗まれてしまった。

盗賊の一人であるコボー氏はアカ・ベコに同情し、気の毒に思っていた。そこで、密かにアカ・ベコに、「あなたがB市に行くお手伝いをしたいが、仲間にばれない様にならないので、直接道順を教えたいが教えることができない。しかし、あなたの質問には答えられる。」と伝言を送った。

コボー氏は、「〇〇という道順はどうか。」という質問をアカ・ベコから受け取ると、それがA市からB市までちょうどたどり付ける道順ならYes、そうでなければNoという答えを伝える。道順のチェックは以下の地図に従って行う。



各都市は一方通行の道でつながっており、それぞれの道には、0か1の数字が付いている。アカ・ベコは数字の並びで道順を指定する。例えば、0100はA市からX、Z、W市を経由してB市にちょうどたどり着く道順である。地図にない道順を選ぶと砂漠に迷い込んでしまい、決してB市にたどり着くことはできない。アカ・ベコは自分がいる都市の名前を知ることはできないため、道順をたどり終えたときにちょうどB市に到達する必要がある。

コボー氏はアカ・ベコからの質問に答えるために、あなたを密かに雇いプログラムの作成を依頼した。アカ・ベコからの質問を入力すると、それがA市からB市にちょうどたどり付ける道順ならYes、そうでなければNoと出力するプログラムを作成してほしい。

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# (シャープ) 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。

p

1行に道順を示す数字 0, 1 の並び p が与えられる。p は100文字を超えない文字列である。

データセットの数は 100 を超えない。

出力

データセットごとに、Yes または No を 1 行に出力する。

入出力例

入力例	出力例
0100	Yes
0101	No
10100	Yes
01000	No
0101011	Yes
0011	No
011111	Yes
#	

4 つ目のデータセットでは、B 市を通りすぎて Y 市に到達する道順なので No と出力する。

5 つ目のデータセットでは、同じ都市を何度か通っているが、B 市にちょうどたどり着く道順なので Yes と出力する。

6 つ目のデータセットでは、X 市から砂漠に迷い込んでしまい B 市にはたどり着けないので No と出力する。

最後のデータセットでは、B 市をいったん通り過ぎてから、X 市と Z 市を経由して B 市にちょうどたどり着く道順なので Yes と出力する。

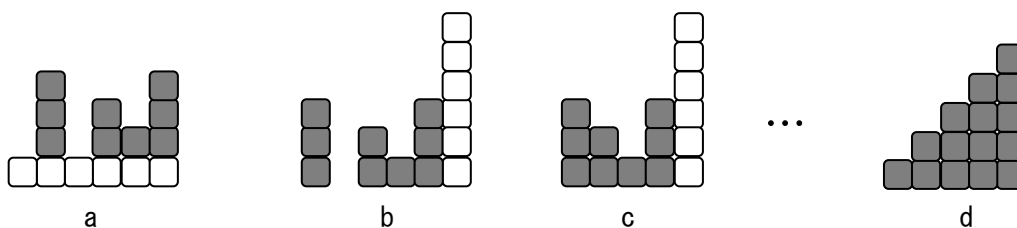
問題2 ブロックの三角形

(7点)

図 a のように積まれたブロックに対し、以下の並べ替え操作を繰り返す。

- (1) 一番下のブロック全て (図 a 中の白のブロック) を右端に新しく積み上げる (残りのブロックは自動的に1段下に落ち、図 b のようになる)。
- (2) ブロックの間に隙間ができたなら、左に詰めて隙間をなくす (図 b から図 c のようになる)。

1以上の整数 k に対して、 $k \times (k+1) / 2$ で表される数 (例: 1, 3, 6, 10, ...) を三角数という。ブロックの総数が三角数の場合、上記の並べ替えを繰り返すと、左端の高さが1で右に向かって1つずつ高くなっていくような三角形になると予想されている (図 d は総数が15個の場合)。



ブロックの最初の並びが与えられたとき、あらかじめ決められた回数以下の操作で、上で説明したようなブロックの三角形ができるとき、三角形が得られるまでの最小の操作回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
b1 b2 ... bN
```

各データセットは2行であり、ブロックの最初の並びを表す。 N ($1 \leq N \leq 100$) は、一番下の段にあるブロックの数を示す。 b_i ($1 \leq b_i \leq 10000$) は左から i 番目の位置に積まれているブロックの数を示す。ただし、ブロックの総数は3以上である。

データセットの数は20を超えない。

出力

データセットごとに、三角形ができるまでに行った並べ替え操作の回数を1行に出力する。ただし、三角形が作れない場合や、操作回数が10000回を超える場合は-1を出力する。

入出力例

入力例	出力例
6	24
1 4 1 3 2 4	0
5	10
1 2 3 4 5	-1
10	48
1 1 1 1 1 1 1 1 1 1	5049
9	-1
1 1 1 1 1 1 1 1 1	
12	
1 4 1 3 2 4 3 3 2 1 2 2	
1	
5050	
3	
10000 10000 100	
0	

最初のデータセットが、図に示した場合に対応する。

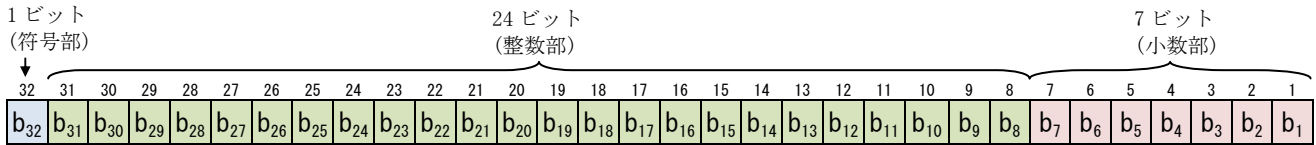
4つ目のデータセットが、ブロックの総数が三角数でないため、三角形が作れない場合に対応する。

最後のデータセットが、ブロックの総数は三角数だが、操作回数が10000回を超える場合に対応する。

問題3 金剛の型

(7点)

以下のように、右から7ビット分が小数部、続く24ビット分が整数部で、一番左の1ビット分を符号部とする32ビット実数型を考える(b_1, \dots, b_{32} は0か1を表す)。



この形式を、人が理解しやすい10進数表現に直すときは、以下のように解釈する。

$$(-1)^{\text{符号部}} \times (\text{整数部} + \text{小数部})$$

上の表現で、整数部の値は $b_8 + 2^1 \times b_9 + 2^2 \times b_{10} + \dots + 2^{23} \times b_{31}$ になる。例えば整数部が

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

のようになっていた場合、整数部の値は以下のように計算される。

$$1 + 2^1 + 2^3 = 1 + 2 + 8 = 11$$

一方、小数部の値は $(0.5)^1 \times b_7 + (0.5)^2 \times b_6 + \dots + (0.5)^7 \times b_1$ になる。例えば小数部が

7	6	5	4	3	2	1
1	0	1	0	0	0	0

のようになっていた場合、小数部は以下のように計算される。

$$0.5^1 + 0.5^3 = 0.5 + 0.125 = 0.625$$

さらに、符号部、整数部、小数部を合わせた以下のようなビット列の場合、

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0	0

ビット列全体が表す10進数は、以下のようになる(ただし、-1の0乗は1)。

$$(-1)^0 \times (1 + 2 + 8 + 0.5 + 0.125) = 1 \times (11.625) \Rightarrow 11.625$$

32ビットのビット列をQ個入力すると、それらのビット列が表す実数の10進表記を一切の誤差無く出力するプログラムを作成してください。

入力

入力は1つのデータセットからなる。入力データは以下の形式で与えられる。

Q
s_1
s_2
:
s_Q

1行目にビット列の数Q ($1 \leq Q \leq 10000$) が与えられる。続くQ行に入力ビット列 s_i が与えられる。入力

ビット列は 16 進数表記で与えられるとする（例: 0111 1111 1000 1000 0000 0000 0000 0000 は 7f880000 と与えられる）。つまり、Q 行のそれぞれには、2 進数を 4 ビット分ずつまとめた 16 進数が 8 個、空白をはさまずに並んでいる。10 進数、2 進数、16 進数の対応は以下の表のとおりである。

10 進数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2 進数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
16 進数	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f

16 進数の英字 a から f までは小文字で与えられる。

出力

各ビット列が表す実数の 10 進表記を 1 行ずつ出力する。ただし、小数部において、ある桁以下がすべて 0 の場合、その桁以下は省略するものとするが、例外として小数部が 0 の場合は小数部には 0 ひとつを出力する。

入出力例

入力例	出力例
8	0.0
00000000	-0.0
80000000	1.0
00000080	0.5
00000040	1.5
000000c0	2.0
00000100	-15.0
80000780	-30.875
80000f70	

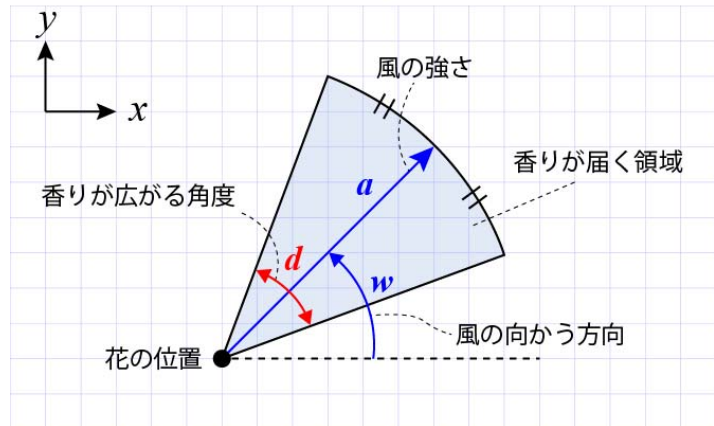
問題4 風よ、私の梅の香りを届けておくれ！

(10点)

引っ越しが決まり、この地を去ることになった。この土地自体に未練は無いが、1つだけ気になることがある。それは、庭に植えた梅の木のことだ。私は毎年、この梅が花を咲かすことを楽しみにしていた。ここを離れた後は春の楽しみが1つ減ってしまう。私の梅の香りだけでも風に乗って引っ越し先の家まで届き、春を楽しませてはくれないものか。

日本には春を象徴する3つの花がある。梅・桃・桜の3つだ。引っ越し先には、私の梅以外にも、これらの花の香りが届くだろう。しかし、私の梅の香りだけが届く日数が最も多い家に住みたい。

図のように、花の香りは扇状に広がり、その領域は風の向かう方向と強さによって決まる。扇形は風の向かう方向 w を中心にして対称に広がり、風の強さ a を半径とする領域をもつ。香りが広がる角度 d は花の種類によって決まっているが、風の向かう方向と強さは日によって異なる。ただし、同じ日では、すべての場所で風の向かう方向と強さは同じである。



手元には、私の梅以外の梅・桃・桜の位置と花の種類ごとの香りが広がる角度、引っ越し先の家の候補のデータがある。さらに、数日分の風の向かう方向と強さのデータもある。私の梅以外の梅・桃・桜の木と家の位置は、私の梅の位置を原点とした座標で示されている。

これらのデータを使って、私の梅の香りだけが届く日数の最も多い家を探すためのプログラムを書いてみることにしよう。私は有能なプログラマーなのだから！

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2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。

```
H R
hx1 hy1
hx2 hy2
:
hxH hyH
U M S du dm ds
ux1 uy1
ux2 uy2
:
uxU uyU
mx1 my1
mx2 my2
:
```

```
mxM myM
SX1 SY1
SX2 SY2
:
SXS SYS
W1 a1
W2 a2
:
WR aR
```

1 行目に引っ越し先の家の候補の数 H ($1 \leq H \leq 100$) と風の記録の数 R ($1 \leq R \leq 100$) が与えられる。続く H 行に引っ越し先の家の位置が与えられる。 hx_i と hy_i は i 番目の家の x 座標と y 座標を示す -1000 以上 1000 以下の整数である。

続く 1 行に、私の梅以外の梅の木の数 U と、桃・桜の木の数 M 、 S 、梅・桃・桜の香りが広がる角度 du 、 dm 、 ds が与えられる。 U 、 M 、 S の範囲は 0 以上 10 以下である。角度の単位は度であり、 1 以上 180 未満の整数である。

続く U 行に私の梅以外の梅の木の位置、続く M 行に桃の木の位置、続く S 行に桜の木の位置が与えられる。 ux_i と uy_i 、 mx_i と my_i 、 sx_i と sy_i はそれぞれ、 i 番目の梅・桃・桜の木の x 座標と y 座標を示す、 -1000 以上 1000 以下の整数である。

続く R 行に風の記録が与えられる。 w_i ($0 \leq w_i < 360$) と a_i ($0 < a_i \leq 100$) は i 日目の風の向かう方向と強さを表す整数である。風の向かう方向は、 x 軸の正の向きから反時計回りに測った角度で表し、単位は度とする。

入力は以下の条件を満たすと考えてよい。

- 入力される座標はすべて異なるものとする。
- 原点には私の梅以外無い。
- どの花についても、花の香りが届く領域の境界から距離が 0.001 以内には家は無い。

データセットの数は 50 を超えない。

出力

データセットごとに、私の梅の香りだけが届く日数の最も多い全ての家の番号を、小さい順に 1 行に出力する。家の番号の間は空白 1 つで区切る。行の終わりには空白文字を出力しない。

ただし、どの家についても、私の梅の花の香りだけが届く日が 1 日もなければ、NA と出力する。

入出力例

入力例	出力例
6 3 2 1 1 2 5 2 1 3 1 5 -2 3 1 1 1 90 30 45 3 -4 -3 0 2 -2 45 6 90 6 135 6 2 1 1 3 5 2 0 1 1 90 30 45 -3 0 2 -2 45 6 0 0	5 6 NA

問題5 モジュロ・クエリ

(10点)

あなたに N 枚のカードを渡します。どのカードにも一つだけ自然数が書いてあります。ただし、同じ数が書いてあることはありません。

これから質問として、適当な自然数を言います。あなたが持っているカードに書いてある数を私が言った数で割ったときに得られる余りのうち最も大きなものを答えてください。

たとえば、あなたは 3 枚のカードを持っていて、それぞれ 9、3、8 と書いてあるとします。私が「4」と言ったら、9 と 3 と 8 をそれぞれ 4 で割った余りを求めてください。余りはそれぞれ 1、3、0 ですが、この中でもっとも大きな余りは 3 なので、3 が正しい答えになります。

では始めましょうか。え？ カードがいっぱいあるとたいへんだ？ しょうがないですね。それではコンピュータを使って最大の余りを見つけることにしましょう。

カードに書いてある数を、質問された数で割った余りのうち、最大のものを見つけるプログラムを作成してください。なお、質問は 1 回だけでなく何度もしますが、同じ数を 2 回以上質問することはありません。

入力

入力は 1 つのデータセットからなる。入力データは以下の形式で与えられる。

```
N Q
c1 c2 ... cN
q1
q2
:
qQ
```

1 行目にカードの枚数 N ($2 \leq N \leq 300000$) と質問の回数 Q ($2 \leq Q \leq 100000$) が与えられ、2 行目にカードに書かれた数 c_i ($1 \leq c_i \leq 300000$) が与えられる。続く Q 行に質問として与えられる数 q_i ($1 \leq q_i \leq 300000$) が与えられる。

出力

質問ごとに最大の余りを 1 行に出力する。

入出力例

入力例	出力例
3 3	3
9 3 8	3
4	4
6	
5	

問題6 イヅア国語辞典

(10点)

あなたはイヅア国の公用語であるイヅア語の国語辞典と、イヅア語のアルファベット（文字の一覧表）を手に入れました。イヅア語のアルファベットには N 種類の文字があります。イヅア語の国語辞典に現れる単語の順番は、イヅア語のアルファベット順に並んでいます。

辞典を見ると、載っているどの単語も N 文字で、しかも、 N 種類の文字をひとつずつ含んでいることがわかりました。さらに調べてみると、辞典には N 種類の文字の可能な並び方がすべて書かれていることを発見しました。

この発見から、あなたはある単語が辞典の何番目に出てくるかわかるようになりました。この知識を利用してイヅア国の人を驚かせてみましょう。まず、 N 種類の文字をひとつずつアルファベット順に並べます。次に、任意の2文字の順番を入れ替える操作を R 回繰り返してもらいます。あなたは、出来上がった単語がイヅア国語辞典の何番目に出てくるか当てて見せます。そのための準備として、国語辞典中の単語の場所を求めるプログラムを作成してください。ただし、アルファベット順で最初の単語を0番目の単語とします。

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
R
s1 t1
s2 t2
:
sR tR
```

1行目にアルファベットを構成する文字の数 N ($1 \leq N \leq 100000$) が与えられ、2行目に文字を入れ替えてもらう回数 R ($0 \leq R \leq 50$) が与えられる。続く R 行に、入れ替えられる文字の位置の組が与えられる。 s_i と t_i ($1 \leq s_i < t_i \leq N$) は、先頭から数えて s_i 番目と t_i 番目の文字を i 回目に入れ替えることを表す。

データセットの数は100を超えない。

出力

データセットごとに、入れ替えが終わった時点で得られた単語が国語辞典の何番目に現れるかを示す数を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	出力例
3	3
2	4
1 2	
2 3	
4	
2	
2 3	
2 4	
0	

入出力例の最初のデータセットについて説明する。説明のため、アルファベットは小さい方から順番に、'A', 'B', 'C' の3文字からなるとする。これらの文字をアルファベット順に並べると"ABC"という文字列が得られる。この文字列の1番目の文字'A'と2番目の文字'B'を入れ替えると、"BAC"という文字列が得られる。次に、得られた文字列の2番目の文字'A'と3番目の文字'C'を入れ替えると、単語"BCA"が得られる。辞典には、先頭から順に"ABC", "ACB", "BAC", "BCA", "CAB", "CBA"の6種類の単語が載っている。"BCA"はこの中で3番目(最初の"ABC"が0番目であることに注意)に現れるので、3を1,000,000,007で割った余りである3が答えになる。

問題7 寂しがり屋のついた嘘

(10点)

ポケットモンスターというゲームがあります。モンスターを捕まえ、育て、戦わせたり交換したりするゲームで、日本中で大流行しています。もちろん、わたしのクラスでも。だからわたしもこのゲームでみんなと遊べるようにモンスターを育てているけど、悲しいことに対戦する相手がいません。わたしは積極的に人に話しかけるのが得意なタイプではないからです。だから今日も、クラスメイトが誘ってくれるのを待ちながら、独り寂しくモンスターのレベルを上げています。

そしてついに、待ち望んでいた瞬間が訪れました。クラスメイトの一人が対戦しないかとわたしに話を持ちかけてきたのでした。彼女は自慢気に、彼女のモンスターたちを見せてくれました。

一般的なポケットモンスターの対戦ルールでは、各プレイヤーがそれぞれ N 匹のモンスターを用意し、互いに 1 匹ずつ戦わせます。各モンスターには成長の度合いを示す「レベル」という数値があり、レベルが大きいモンスターが必ず勝ちます。同レベルのときは引き分けです。一度戦ったモンスターは、それ以上戦わせることができません。こうして一対一の戦いを N 回行ったあと、過半数の勝利を収めたプレイヤーが勝者となります。

わたしは、彼女のモンスターを見た瞬間、嫌な予感がしました。彼女のモンスターはどれも、とても強そうなのです。わたしと戦ってくれるのなら負けてもかまいません。でも、あまりにもひどい負け方をしたら、『こいつと戦っても面白くないな』とか思われて、もう遊んでくれなくなるんじゃないかな。それは、とても嫌です。だから、

「ごめん、わたしまだモンスターを N 匹持ってなくて」

嘘をつきました。

N 回勝負では勝ち目がなくても、それより少ない数のモンスターで勝負する特別ルールなら、もしかしたら勝てるかもしれません。さっきは負けたってかまわないと言ったけど、やっぱり勝てたら嬉しいです。

彼女はこの特別ルールを受け入れてくれました。つまり、二人が持っているモンスターからお互いに k 匹選んで一匹ずつ順番に戦わせ、過半数の(つまり、 $k/2$ より大きな数の)勝利を収めた方が勝ちとなります。

さっきわたしは、彼女のモンスターのレベルを知ってしまいました。彼女がどのモンスターを選び、どんな順番で出してくるのかはわかりません。でも、対戦させるモンスターの数 k をわたしがうまく決めれば、彼女がどんな選択をしようとわたしが勝つことができるかもしれません。

皆さんにお願いします。モンスターの数 N と、二人が持っているモンスターのレベルを入力すると、彼女がどんな選択をしようとわたしが勝てるような最小のモンスターの数 k を出力するプログラムを作成して下さい。

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
a1 a2 ... aN
b1 b2 ... bN
```

1 行目にモンスターの数 N ($1 \leq N \leq 40000$) が与えられる。2 行目に自分のモンスターのレベル a_i ($1 \leq a_i \leq 100000$) が与えられる。3 行目にクラスメイトのモンスターのレベル b_i ($1 \leq b_i \leq 100000$) が与えられる。

データセットの数は 50 を超えない。

出力

データセットごとに、 k ($1 \leq k < N$) が存在すれば最小値を 1 行に出力する。もし、 k が存在しないか、 k が N に等しければ NA と出力する。

入出力例

入力例	出力例
10	3
4 9 1 9 5 9 2 3 2 1	1
8 7 6 5 10 5 5 4 7 6	NA
5	
4 3 2 5 1	
4 4 4 4 4	
4	
4 1 3 2	
4 3 2 1	
0	

1 つ目のデータセットでは、わたしは 3 匹での対戦を提案し、レベルが [9, 9, 9] のモンスターを選べば良い。相手のレベル 10 以外のどのモンスターにも勝てるので、少なくとも 2 勝できる。2 匹での対戦では 1 勝 1 敗になる可能性があり、これは過半数の勝利とは言えないので、2 は答えにならない。

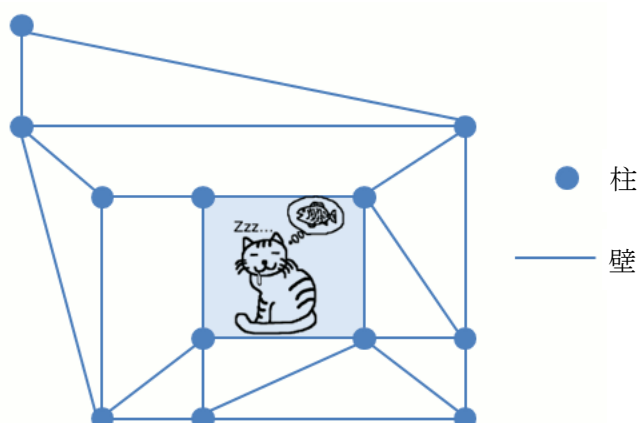
2 つ目のデータセットでは、わたしは最強のモンスターを 1 匹選べば良い。

3 つ目のデータセットでは、わたしがどの k 匹を選び、どの順番で出したとしても、彼女が全く同じモンスターを同じ順番で出してきて引き分けになる可能性がある。したがって、「彼女がどんな選択をしようとしたとしてもわたしは勝つことができる」とは言い切れない。

問題8 ねこまっしぐら2

(13点)

あるところに大きな古いお屋敷があり、1匹のねこが住み着いていました。図のように、お屋敷は上空から見ると凸多角形になっており、まっすぐな壁で囲まれたいくつかの部屋で造られています。1枚の壁はその両端にある柱によって支えられています。お屋敷はとても古いため、どの壁にもねこが通ることができるくらいの穴が1つ空いています。ねこは壁の穴をとお屋敷の部屋と部屋または部屋と外を行き来することができます。



お屋敷のご主人は、ねこに餌をあげるために、口笛を吹いてねこをお屋敷の外へ呼び出します。ねこはとても賢く、ご主人が口笛を吹くと、「最良の選択」でお屋敷の外へ抜け出します。つまり、最も少ない回数だけ穴をとお屋敷の外へ出ます。

ねこはとても気まぐれで、どの部屋にいるか分かりません。そのため、ねこがお屋敷の外へ出るのにかかる時間は日によって異なり、ご主人はどれだけ待てばよいかわからず困っていました。ある時ご主人は、ねこが穴を通り抜けるときに、とても時間がかかっていることに気が付きました。ということは、ねこが外に出るのにかかる時間は穴を通る回数によって決まることとなります。ご主人は、ねこが「最良の選択」をした場合の穴を通る回数の最大値が分かれば、ねこがどの部屋にいたとしても、最大どれだけ待てばよいか分かるのではないかと考えました。ご主人はお屋敷の間取りを知っていますが、お屋敷はとても大きく、自分では計算することができません・・・。

お屋敷の間取りを読み込み、ねこが「最良の選択」をした場合、外に出るまでに通る穴の数の最大値を出力するプログラムを作成して下さい。

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2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。

```
C W
x1 y1
x2 y2
:
xC yC
s1 t1
s2 t2
:
sW tW
```

1行目に柱の数 C ($3 \leq C \leq 100$) と壁の数 W ($3 \leq W \leq 300$) が与えられる。続く C 行に柱の座標が与えられる。 x_i ($-1000 \leq x_i \leq 1000$) は i 番目の柱の x 座標、 y_i ($-1000 \leq y_i \leq 1000$) は i 番目の柱の y 座標をそれぞれ示す整数である。柱にはそれぞれ1から C までの番号が割り当てられている。続く W 行に壁の情報が与えられる。

$s_i (1 \leq s_i \leq C)$ 、 $t_i (1 \leq t_i \leq C)$ は壁の両端を支える柱の番号を示す。

入力は以下の条件を満たすと考えてよい。

- 異なる柱が同じ位置を占めることはない。
- 異なる壁が重なったり、柱以外で交差することはない。
- 1つの柱は2つ以上の壁を支えている。
- 壁の長さは0より大きい。
- 柱は壁の両端だけにある。つまり、柱が壁の途中にあることはない。
- 壁は異なる2つの部屋、あるいは部屋と外とを仕切る。
- 任意の異なる2つの柱を選んだとき、互いの柱には壁をたどって到達できる。

データセットの数は50を超えない。

出力

データセットごとに、ねこが「最良の選択」をした場合、外に出るまでに通る穴の数の最大値を1行に出力する。

入出力例

入力例	出力例
4 4	1
0 0	3
1 0	
1 1	
0 1	
1 2	
1 4	
2 3	
3 4	
12 22	
2 0	
4 0	
8 0	
4 2	
6 2	
8 2	
2 4	
4 4	
6 4	
0 6	
8 6	
0 8	
1 2	
2 3	
1 4	
1 7	
1 10	
2 4	
2 5	

3 5	
3 6	
4 5	
4 8	
5 6	
5 9	
6 9	
6 11	
7 10	
7 8	
8 9	
9 11	
10 11	
10 12	
11 12	
0 0	

2つ目のデータセットが、冒頭の図に示したお屋敷の間取りに対応する。図のように、薄く色が塗られた部屋にねこがいるときは、「最良の選択」で外に出るまでに通り抜ける穴の数は3つである。それ以外の部屋にねこがいるときは、2つ以下で外に出られる。したがって、ねこが「最良の選択」をしたとき、外に出るまでに通り抜ける穴の数の最大値は3になる。

問題9 図画工作

(13点)

イヅア大学附属小学校は日本有数の競技プログラマー養成校として有名です。この学校の教員は幅広いアルゴリズムの知識を持ち、日々それを活用しています。教員であるあなたは、今年は図画工作の授業を担当することになりました。この授業では、児童全員がそれぞれ1年間で一つの課題を完成させることになっています。授業の概要は以下のとおりです。

- 1年間で授業はD回（同じ日に2回以上授業はない）あり、その全てが課題制作に充てられる。
- 制作する課題はM種類用意されている。
- それぞれの児童に、M種類の中から課題を1つずつ割り当てる。
- 児童はN人であり、N人それぞれに異なる課題が割り当てられる。

児童は、K種類ある部品のうちいくつかの種類の部品を使用して課題を完成させます。課題の制作の概要は以下のとおりです。

- 課題ごとに、使用すべき部品の種類と数はあらかじめ決められている。
- 課題が完成するまでに使用する部品の種類と数は、その課題で使用すべき部品の種類と数に、過不足なく一致していなければならない。
- 異なる課題でも、使われる部品の種類と数がすべて同じ場合もある。
- どの課題も、同じ種類の部品は2個までしか使用できない。
- 部品を使用する順序は課題の完成に影響を与えない。
- いくつかの部品が入っている袋が事前にP個用意されている。ただし、異なる袋でも、入っている部品の種類と数がすべて同じ場合もある。
- 教員は、児童1人につき袋を1つだけ渡すことができる（袋を渡さない児童がいてもよい）。
- 2人以上の児童に同じ袋を渡すことはできない（反対に、誰にも渡されない袋があってもよい）。
- 袋を渡された児童は、袋の中に入っている部品をすべて、自分が制作する課題に使わなければならない。

袋に入っている部品以外で課題に使用する部品は、別途購入する必要があります。部品の購入に関する条件は以下のとおりです。

- 部品は授業の日だけ購入でき、その日にしか使えない。
- それぞれの課題について、1回の授業でL個までの部品を購入することができる。
- 部品の種類によって価格が設定されており、購入する日によって価格が変動する。ただし、どの種類も品切れになることはない。

あなたは、このような条件下で、授業にかかる経費をなるべく抑えたいと考えています。そこで、児童全員の部品購入費の合計の最小値を計算するプログラムを作成することにしました。

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3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。

D	K	L
$c_{1,1}$	$c_{1,2}$	$\dots c_{1,K}$
$c_{2,1}$	$c_{2,2}$	$\dots c_{2,K}$
:		
$c_{D,1}$	$c_{D,2}$	$\dots c_{D,K}$
M	N	P
$r_{1,1}$	$r_{1,2}$	$\dots r_{1,K}$
$r_{2,1}$	$r_{2,2}$	$\dots r_{2,K}$
:		
$r_{M,1}$	$r_{M,2}$	$\dots r_{M,K}$
$t_{1,1}$	$t_{1,2}$	$\dots t_{1,K}$
$t_{2,1}$	$t_{2,2}$	$\dots t_{2,K}$
:		
$t_{P,1}$	$t_{P,2}$	$\dots t_{P,K}$

D ($1 \leq D \leq 8$) は授業の回数、K ($1 \leq K \leq 8$) は部品の種類の数、L ($1 \leq L \leq 8$) は1日に購入することができる部品の数を示す。 $c_{d,k}$ ($1 \leq c_{d,k} \leq 100$) はd日目の部品kの価格を示す。

M ($1 \leq M \leq 200$) は課題の種類の数、N ($1 \leq N \leq M$) は生徒の人数、P ($1 \leq P \leq 200$) は袋の数を示す。

$r_{m,k}$ ($0 \leq r_{m,k} \leq 2$) は課題mに必要な部品kの個数、 $t_{p,k}$ ($0 \leq t_{p,k} \leq 2$) は袋pに入っている部品kの個数を示す。

部品を1つも必要としない課題や、空の袋は与えられないものとする。

データセットの数は20を超えない。

出力

データセットごとに、生徒全員の部品購入費の合計の最小値を1行に出力する。N種類の作品を完成させることができない場合は-1を出力する。

入出力例

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

問題 10 鉄道路線

(13 点)

ある国の鉄道網に、自動改札を導入することになりました。導入にあたって難しい問題の一つは、与えられた切符で、指定された駅の間を移動できるかどうかを判定することです。それぞれの切符には乗車駅と降車駅が記載されています。この切符でできるのは、「乗車駅で乗って、降車駅で降りる」ことだけでなく、途中乗車や途中下車も許されています。

この鉄道網には S 個の駅があり、そのうち R 組の駅は隣り合っていて、他の駅を経由せずに双方向に鉄道で移動することができます。隣り合った駅を結ぶ線路はひとつしかありません。隣り合った駅間の距離は、この線路に沿って測った距離です。ある駅からある駅までの経路は鉄道網の形状によっては複数通り考えられますが、そのうち最も距離が短くなるような経路を最短経路と呼ぶことにします。そのような経路が複数ある場合、どちらも最短経路として認められます。

乗車駅 a 、降車駅 b の切符で駅 c から駅 d まで移動できるのは、以下の条件をすべて満たす経路 p が存在するときです。

- 経路 p は、駅 a から駅 b への最短経路である。
- 経路 p は、駅 a から出発し、駅 c 、駅 d の順に経由し、駅 b で終わる経路である。また、駅 c から駅 d の区間はこの 2 駅の最短経路になっている。

路線図と切符の情報が与えられます。次に、始点と終点の組がいくつか与えられるので、その切符で始点から終点へ移動できるかどうかを判定するプログラムを作成してください。

入力

入力は 1 つのデータセットからなる。入力データは以下の形式で与えられる。

```
S R
u1 v1 w1
u2 v2 w2
:
uR vR wR
a b Q
c1 d1
:
cQ dQ
```

1 行目は 2 つの整数からなる。 S ($2 \leq S \leq 100000$) は鉄道路線図に現れる駅の数、 R ($1 \leq R \leq 200000$) は隣り合った駅の組の数である。続く R 行に、隣り合った駅の間を直接つなぐ線路の情報が与えられる。 u_i と v_i ($1 \leq u_i, v_i \leq S$) は i 番目の線路の両端の駅の番号を示す。 w_i ($1 \leq w_i \leq 1000$) はこれらの駅間の距離を表す整数である。ただし、各駅には 1 から S までの番号が重複なく割り振られており、 $u_i \neq v_i$ とする。

続く 1 行は 3 つの整数からなる。最初の 2 つの整数は切符の区間を表し、 a は乗車駅、 b は降車駅 ($1 \leq a, b \leq S$) である。3 つ目の整数 Q ($1 \leq Q \leq 40000$) は質問の数を示す。続く Q 行に質問が与えられる。 c_i と d_i ($1 \leq c_i, d_i \leq S$) は i 番目の質問の乗車駅と降車駅を示す。ただし、 $a \neq b$ 、 $c_i \neq d_i$ とする。

出力

質問ごとに、与えられた切符で移動できるなら Yes を、できないなら No を 1 行に出力する。

入出力例

入力例	出力例
6 7	Yes
1 2 3	Yes
1 4 1	Yes
2 3 5	Yes
4 3 1	No
3 6 2	No
4 5 2	
5 6 1	
1 6 6	
1 6	
4 3	
4 6	
5 6	
2 6	
2 5	