
問題01 スピードスケートバッジテスト (3点)

スピードスケートバッジテストでは、2種類の距離で規定されたタイムを上回った場合に級が認定されます。例えばA級になるには500Mで40.0秒未満、かつ1000Mで1分23秒未満であることが求められます。スピードスケート大会（500Mと1000M）で記録したタイムを入力とし、スピードスケートバッジテスト何級に相当するかを出力して終了するプログラムを作成して下さい。下記の表は、500Mと1000Mのバッジテスト規定タイムです。E級に満たなかった場合には、NAと出力して下さい。また、入力データは、秒単位で入力されます。

男子	500M	1000M
AAA級	35秒50	1分11秒00
AA級	37秒50	1分17秒00
A級	40秒00	1分23秒00
B級	43秒00	1分29秒00
C級	50秒00	1分45秒00
D級	55秒00	1分56秒00
E級	1分10秒00	2分28秒00

入力

500Mタイム 1000Mタイム（実数 実数；半角空白区切り）

出力

判定結果AAA～EまたはNA（半角英数）

入力例1	出力例1
40.0 70.0	B

入力例2	出力例2
72.5 140.51	NA

問題02 リーグ戦星取表(3点)

スポーツの大会にはリーグ戦とトーナメント戦があります。サッカーのリーグ戦では勝・負・引分にそれぞれ点数を付け、その勝ち点で順位を競います。勝ち点はそれぞれ勝(3点)、負(0点)、引分(1点)です。チーム数とリーグ戦の成績を入力とし、成績の良い順(勝ち点の多い順)に並べ替え、チーム名と勝ち点を出力して終了するプログラムを作成してください。勝ち点が高点の場合は入力順に出力してください。チーム名は20文字以内のアルファベットとし、チーム数は10チーム以内とします。



入力

1行目 チーム数(整数)

2行目 チーム名1 勝の数 負の数 引分の数

(半角20文字以内のアルファベット 整数 整数 整数; 半角空白区切り)

3行目 チーム名2 勝の数 負の数 引分の数

(半角20文字以内のアルファベット 整数 整数 整数; 半角空白区切り)

:

:

出力

チーム名, 勝ち点(半角英字, 整数; 半角カンマ区切り)

:

:

入力例1	出力例1
4 Japan 1 0 2 Egypt 1 2 0 Canada 0 2 1 Spain 2 0 1	Spain, 7 Japan, 5 Egypt, 3 Canada, 1

	J	E	C	S	勝	負	引分	勝点
Japan		○	△	△	1	0	2	5
Egypt	×		○	×	1	2	0	3
Canada	△	×		×	0	2	1	1
Spain	△	○	○		2	0	1	7

○: 勝 ×: 負 △: 引分

入力例2	出力例2
3 India 0 2 0 Poland 1 0 1 Italy 1 0 1	Poland, 4 Italy, 4 India, 0

	Ind	P	Ita	勝	負	引分	勝点
India		×	×	0	2	0	0
Poland	○		△	1	0	1	4
Italy	○	△		1	0	1	4

○:勝 ×:負 △:引分

問題03 日数計算(3点)

2つの日付を入力とし、その2つの日付けの間の日数を出力して終了するプログラムを作成してください。日付1 (y1, m1, d1) は日付2 (y2, m2, d2) と同じか、あるいはそれ以前の日付とします。日付1は日数に含め、日付2は含めません。また、うるう年を考慮に入れて計算してください。うるう年の条件は次のとおりとします。

- 西暦年が4で割り切れる年であること。
- ただし、100で割り切れる年はうるう年としない。
- しかし、400で割り切れる年はうるう年である。

入力

y1 m1 d1 y2 m2 d2 (それぞれ整数 ; 半角空白区切り)

出力

日数 (整数)

入力例1	出力例1
2006 9 2 2006 9 3	1

入力例2	出力例2
2006 9 2 2006 11 11	70

入力例3	出力例3
2004 1 1 2005 1 1	366

入力例4	出力例4
2000 1 1 2006 1 1	2192

入力例5	出力例5
2000 1 1 2101 1 1	36890

問題04 パズル(7点)

たろう君は9×9のマス目に1～9の数字を配置するパズルで遊んでいます。このパズルでは以下の規則で数字を配置しなければいけません。

- 同じ列に1つの数字はちょうど1回だけ現われる
- 同じ行に1つの数字はちょうど1回だけ現われる
- 二重線で区切られた3×3の各範囲には、1つの数字はちょうど1回だけ現われる

例えば、下の図1はそのような規則を満たした配置の一つです。しかしたろう君は、図2のようによく規則に反した配置を作ってしまう。左端の列に「2」が2回現われて、「1」が1回も現われず、左から2番目の列に「1」が2回現われて、「2」が1回も現われていません。たろう君を助けるために、数字の配置を読み込んで、その配置が規則を満たしているかどうかを調べ、規則に反していたらその場所を出力し終了するプログラムを作成してください。3つの規則に照らして誤っている(2回以上現れている)数字の前には* (半角アスタリスク) を、誤っていない数字の前には空白を表示してください。

1	2	3	4	5	6	7	8	9
4	5	6	7	8	9	1	2	3
7	8	9	1	2	3	4	5	6
2	3	4	5	6	7	8	9	1
5	6	7	8	9	1	2	3	4
8	9	1	2	3	4	5	6	7
3	4	5	6	7	8	9	1	2
6	7	8	9	1	2	3	4	5
9	1	2	3	4	5	6	7	8

図1

2	1	3	4	5	6	7	8	9
4	5	6	7	8	9	1	2	3
7	8	9	1	2	3	4	5	6
2	3	4	5	6	7	8	9	1
5	6	7	8	9	1	2	3	4
8	9	1	2	3	4	5	6	7
3	4	5	6	7	8	9	1	2
6	7	8	9	1	2	3	4	5
9	1	2	3	4	5	6	7	8

図2

入力

パズルの状態を示す1行9文字、9行からなる数字列 (整数 ; 半角空白区切り)

出力

与えられた数字と*(半角アスタリスク)と空白。誤っている数字の前には*(半角アスタリスク)、誤っていない数字の前には半角空白を付加する。

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

問題05 ポケベル打ち、サイコー！？(7点)

ある日、たろう君は、本文に「519345213244」という数字だけが書いてあるおかしなメールを受け取りました。そのメールは、10歳年上の従姉からだったので電話して聞いてみると、「あ～、急いでたからポケベル打ちで送っちゃった。便利よね～。よろしくねっ！」と切って切られてしまいました。いつもせわしなく、少し強引なところのあるこの従姉をよく知るたろう君は、仕方なく自分で「ポケベル打ち」について調べてみると、10年ぐらい前に一世を風靡したかな入力の方法だということが分かりました。

「ポケベル打ち」では、図1のような変換表に従い‘あ’は11、‘お’は15という具合に、2つの数字で一つの文字を入力していきます。例えば「なると」という文字列を入力するには「519345」と打ちます。従って、どの文字も2つの数字で入力することができるのです。

	1	2	3	4	5	6	7	8	9	0
1	あ	か	さ	た	な	は	ま	や	ら	わ
2	い	き	し	ち	に	ひ	み	・	り	を
3	う	く	す	つ	ぬ	ふ	む	ゆ	る	ん
4	え	け	せ	て	ね	へ	め	・	れ	ゝ
5	お	こ	そ	と	の	ほ	も	よ	ろ	°

図1

携帯電話がまだ広く普及していないころ、高校生はこの方法で公衆電話から友達のパケベルにメッセージを送信していたのだそうです。猛烈なスピードでポケベル打ちができる女子高生もいたとか。最近、仕事も忙しくなってきた従姉は、メールを無意識のうちにポケベル打ちでタイプするようになってしまいました。

そこで、毎回解読に苦勞しているたろう君を助けるために、ポケベル打ちによるメッセージを文字列に変換し出力して終了するプログラムを作成してください。ただし、変換には図2の変換表を使い、英小文字及び、「.」、「?」、「!」、空白のみを対象とします。入力される文字列は200文字以内とします。変換出来ない文字を含むメッセージに対してはNAと出力してください。

	1	2	3	4	5	6
1	a	f	k	p	u	z
2	b	g	l	q	v	.
3	c	h	m	r	w	?
4	d	i	n	s	x	!
5	e	j	o	t	y	空白

図2

入力

1行目 メッセージの数 (整数)

2行目 メッセージ1

3行目 メッセージ2

:

:

出力

1行目 変換後のメッセージ1、または NA

2行目 変換後のメッセージ2、または NA

:

:

入力例	出力例
8	naruto
341143514535	NA
314	do you wanna go to aizu?
143565553551655311343411652235654535651124615163	yes sure!
551544654451431564	NA
4	na
3411	?????
6363636363	end
153414	

問題06 そろばん(7点)



そろばんを習い始める友人からのお願いで、あなたはそろばんの珠の並びを表示するプログラムを作成することになりました。ある数字を入力として、そろばんの珠の並びを出力して終了するプログラムを作成してください。

ただし、表示するそろばんの桁数は5桁とし、0から9までの珠の配置はそれぞれ * (半角アスタリスク)、” ” (半角空白)、= (半角イコール) を用いて以下のようにと表すものとします。

0	1	2	3	4	5	6	7	8	9
*	*	*	*	*					
=	=	=	=	=	=	=	=	=	=
	*	*	*	*		*	*	*	*
*		*	*	*	*		*	*	*
*	*		*	*	*			*	*
*	*	*		*	*	*	*		*
*	*	*	*	*	*	*	*	*	*

入力

5桁までの数字 (整数)

出力

そろばんの珠の配置 (半角文字列)

入力例1	出力例1
2006	<pre> **** * ===== * * **** * *** ***** ***** </pre>

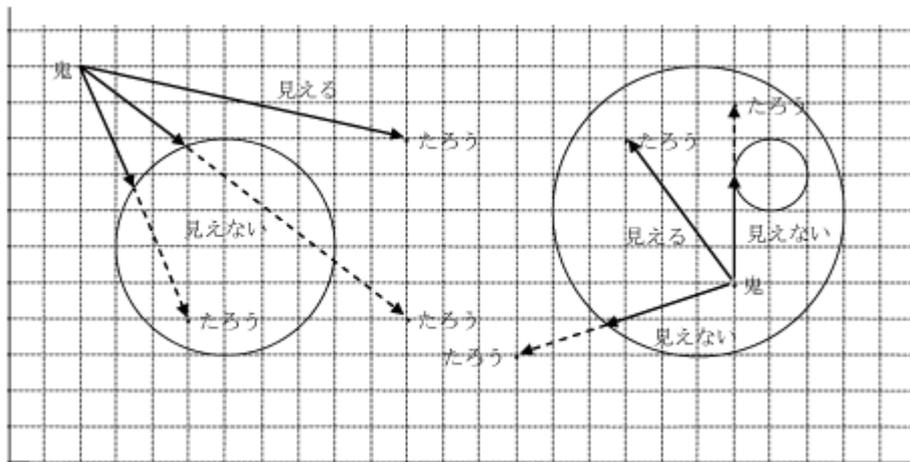
入力例2	出力例2
1111	<pre> ***** ===== **** * ***** ***** ***** </pre>

問題07 かくれんぼう支援システム(13点)

たろう君はかくれんぼうが苦手です。隠れるとすぐ見つっちゃうし、隠れている子をなかなか見つけることができません。見かねたお父さんは、超高性能位置探索システムを作ってくれました。これを使うと自分の位置を含めて、友達の位置を正確に知ることができます。鬼になったら、隠れている子を探すことが簡単にできるようになりました。見つけたい子の位置に行けば、それでOKのはずです。これを使っていることがばれると遊んでもらえなくなるので内緒で使います。

たろう君はさらにこのシステムを進化させ、鬼から自分が見えるか否かを判定する機能を付加することを思いつきました。これができれば、「もういいかい」と言われても、見える位置にいれば「まあだよ」だし、見えない位置にいれば「もういいよ」ですね。

いつも遊んでいる公園には大小さまざまな円筒形の壁があります。この壁は外からは中は見えないし、中から外も見えません。鬼と一緒に中に入ると別の壁がなければ見えちゃいます。



たろう君はアイデアはいいのですが、ソフトを作るのが苦手です。そこで親友のあなたはたろう君に代わって、「鬼から見えないこと確認システム」のソフトウェアを作ることになりました。公園の壁は固定されていますが、たろう君と鬼のいろいろな位置情報に対して見えるか否かを判定する必要があります。公園にある壁の情報(中心の座標 (wx, wy) と半径 r) とたろう君と鬼の位置情報(たろう君の位置の座標 (tx, ty) と鬼の位置の座標 (sx, sy)) を入力とし、その位置で鬼からたろう君が見えるか否かを判定し終了するプログラムを作成してください。

なお、鬼からたろう君が見える場合にはDanger、見えない場合にはSafeと出力してください。鬼とたろう君の位置を結んだ線分上に壁がある場合には見えないものとし、鬼もたろう君も壁の上にはいないものとします。公園内の位置の座標は0以上255以下の整数の組で表されます。公園にある壁は100個以下で、いずれの半径も整数であるとします。また、円筒形の壁の一部だけが公園にあるということではなく、全ての壁は円筒として、公園の内部に入っています。

入力

n 円筒形の壁の数 (整数)
wx1 wy1 r1 壁の情報1 (中心の座標 (wx1, wy1) と半径 (r1))
(それぞれ整数 ; 半角空白区切り)
wx2 wy2 r2 壁の情報2 (中心の座標 (wx2, wy2) と半径 (r2))
(それぞれ整数 ; 半角空白区切り)
:
:
wxn wyn rn 壁の情報n (中心の座標 (wxn, wyn) と半径 (rn))
(それぞれ整数 ; 半角空白区切り)
m たろう君と鬼の位置情報の個数
tx1 ty1 sx1 sy1 位置情報1 (たろう君の位置 (tx1, ty1) と鬼の位置 (sx1, sy1))
(それぞれ整数 ; 半角空白区切り)
tx2 ty2 sx2 sy2 位置情報2 (たろう君の位置 (tx2, ty2) と鬼の位置 (sx2, sy2))
(それぞれ整数 ; 半角空白区切り)
:
:
txm tym sxm sym 位置情報m (たろう君の位置 (txm, tym) と鬼の位置 (sxm, sym))
(それぞれ整数 ; 半角空白区切り)

出力

位置情報1の判定結果DangerまたはSafe (半角英字)
位置情報2の判定結果DangerまたはSafe (半角英字)
:
:
位置情報mの判定結果DangerまたはSafe (半角英字)

入力例	出力例
3	Safe
6 6 3	Safe
19 7 4	Danger
21 8 1	Safe
6	Danger
5 4 2 11	Safe
12 4 2 11	
11 9 2 11	
14 3 20 5	
17 9 20 5	
20 10 20 5	

問題08 列車(13点)



26両以下の編成の列車があります。それぞれの車両には、英小文字のaからzまでの識別記号が付いています。同じ記号が付いている車両はありません。ただし、車両を連結する順番は自由とします。

列車の中を車掌が巡回します。車掌は、列車の中を行ったり来たりして巡回するので、同じ車両を何度も通ることがあります。ただし、すべての車両を最低一回は巡回するものとします。また、巡回をはじめる車両や巡回を終える車両が列車の一番端の車両とは限りません。

ある車掌が乗ったすべての列車の巡回記録があります。そこから分かる各列車の編成を先頭車両から出力して終了するプログラムを作成してください。

巡回記録は1行が1つの列車に対応し、1行の長さは1024文字を越えないものとします。各行は、英小文字を1文字ずつ<-または->で区切った文字列でできています。<-は前方の車両への移動、->は後方の車両への移動を表します。例えば、a->b<-a<-cは車両aから後方の車両であるbに移り、bから前方のaに移り、aから前方のcへ移ったことを表します。この場合の列車の編成は先頭車両からcabとなります。

入力

巡回記録の個数n (整数)

巡回記録1 (1024文字までの半角文字列)

巡回記録2 (1024文字までの半角文字列)

⋮

巡回記録n (1024文字までの半角文字列)

出力

先頭車両からの列車の編成を現す文字列 (半角)

入力例	出力例
4	aecbd
a->e->c->b->d	edacb
b<-c<-a<-d<-e	bacde
b->a->c<-a->c->d<-c<-a<-b->a->c->d->e<-d	bcdae
a->e<-a<-d->a->e<-a<-d<-c->d->a<-d<-c<-b->c->d<-c	

問題09 博士の奇妙な素粒子(19点)

博士： ピーター君。やったよ。

ピーター： またですか。今度はどんなくだらない発明ですか。

博士： あの幻の素粒子アキシオンの検出器を発明したんじゃ。

ピーター： アキシオンといえば、欧州合同原子核研究機構（CERN）をはじめとする研究者たちが血眼で追っかけているという、あれですよ。本当ですかあ？

博士： 本当だってばよ。細かい説明は省くが、非常に強力な磁場を内蔵する特殊な光電管が光ることによって、通過するアキシオンを検出する。

ピーター： 他に先んじて検出すれば、小柴先生のニュートリノ検出に匹敵するノーベル賞級の研究ですよ。

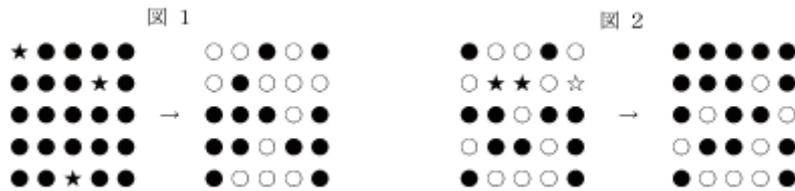
これで役立つ研究ばかりしている「ダメ研究室」などという汚名も返上できますね。

博士： そうだとも。小柴先生の「スーパーカミオカンデ」にあやかっ、この装置を、（悪口言ったら）「タダジャオカンデ」と命名した。

ピーター： ちょっと苦しいって言うか、卑屈って言うか・・・。

博士： それはいいとして、この装置ちょっとした癖があるんじゃよ。

アキシオン粒子がある光電管を通過すると、感度の関係でその光電管と隣接する上下左右の光電管が反応する。



ピーター： つまり、図1の左側★印の光電管を粒子が通過した場合、右側のように点灯するということですね。

（図は 5 × 5 の例。黒は消灯・白は点灯状態。以下同じ。）

博士： また反応とは、光電管の状態が反転する。つまり消えている光電管は光り、光っている光電管は消える。

ピーター： つまり、図 2 の左側の★印や☆印を粒子が通過した場合、右側のような状態になると。

博士： これを正方形になんと 100 個（10 × 10）配置して待ち受けるわけだ。

ピーター： これだけの大発明、ノーベル賞選考委員会も「ホッチャオカンデ」なんて。

博士： おおピーター君、君も我が研究室の作風になじんできたようだね。いい感じだ。では、早速実験を始めよう。まずはこの装置、現在ランダムに光電管が点いているので、

実験が始められるように全部消えた状態にリセットしてくれたまえ。

なあに、どの光電管にアキシオン粒子を当てれば全部消えるか考えればいだけじゃよ。

簡単だろう？

ピーター： 考えるのはいいんですが博士。当てるためには、

幻のアキシオン粒子を発生させて打ち込める装置がないといけないんじゃないですか。

博士： ……。

博士とピーター (同時に) コリャアカンデ! だははは・・・。

ー :

ということで、本日も和気あいあいと暮れてゆく博士の研究室ですが、例によって話は全く進みそうもありません。しょうがないので、ピーター君に代わってプログラムを作成してあげてください。プログラムは次のようなものになります。

- A. 装置の光電管の状態を 10×10 の配列として入力します。0は消灯中、1は点灯中を表します。0と1以外のデータは含まれません。
- B. 入力した装置の状態をすべて消灯にするために、アキシオン粒子を通過させる位置を計算して出力します。これは、入力と同じ 10×10 の配列で光電管の位置を表します。「0は通過させない」、「1は通過させる」ことを表します。すべてを消灯する方法は必ず1通りだけ存在するものとします。
なお、以下はヒントです。
 - a. 粒子を通過させる順番は結果に関係ありません。
 - b. 粒子を偶数回通過させることは通過させないことと同じです。また、奇数回通過させることは1回通過させることと同じです。
 - c. 以上から、すべての光電管について0 (通過させない) と 1 (通過させる) のどちらかを選択することがすなわち解答になります。
- C. プログラムを終了します。

入力

1 行目 装置の1行目の光電管の状態を示す 10 個の数字 (整数 0 または 1; 半角空白区切り)
2 行目 装置の2行目の光電管の状態を示す 10 個の数字 (整数 0 または 1; 半角空白区切り)
:
:

出力

1 行目 装置の1行目の光電管に粒子を通過させるかを示す 10 個の数字 (整数 0 または 1; 半角空白区切り)
2 行目 装置の2行目の光電管に粒子を通過させるかを示す 10 個の数字 (整数 0 または 1; 半角空白区切り)
:
:

入力例1	出力例1
0 1 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 0 0 0	0 0 0 0 1 1 0 0 0 0
0 0 0 0 1 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1	0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0 0 0

問題10 ジグソーパズル(25点)

子供から大人まで脳トレ系パズルゲームが流行しています。我々も遅れを取らぬよう、パズルゲームを作ってみることで遊ぶことにしました。

我々が考えたのはジグソーパズルのゲームで、未完成の部分埋めるのに必要なピースを選ぶというものです。図1(a)はパズルの枠の例です。黒が埋まっている部分、白が未完成の部分です。このパズルを完成するのに使えるピースが図1(b)のように与えられます。この中から、枠の白い部分を埋めることができる黒いピースを1つ以上選択します。例えば、図2の選択例1のような組み合わせなら正解となります。一方、選択例2のような組み合わせではパズルは完成しないので不正解となります。また、選択したピースが余ってしまう場合も不正解です。このように、プレイヤーは適切なピースを選択することでゲームをクリアします。

そこでこのパズルゲームで用いる判定プログラムを開発することになりました。未完成のパズル、ピースの候補、そしてプレイヤーが選んだピースの組み合わせを入力とし、プレイヤーが適切なピースを選ぶことができているならばYESを、そうでなければNOと出力して終了するプログラムを作成してください。

この問題では、パズルの枠は $H \times W$ の配列で表し、未完成の部分を. (半角ピリオド)、完成している部分を# (半角シャープ) によって与えます。パズルの枠の大きさは最大 20×20 とします。また、各ピースは $h \times w$ の配列で表し、ピースを構成する部分を# (半角シャープ)、そうでない部分を. (半角ピリオド) によって与えます。与えられる各ピースは、元の状態から90度、180度、270度だけ回転することができます。また、各ピースの大きさは最大 20×20 とし、与えられるピースの数 (n) は最大10とします。



図1



図2

入力

1行目は、パズルの枠の大きさH（縦） W（横）（整数；半角空白区切り）

2行目以降は、パズルの盤面を表す1行W文字の文字列H行（半角.（ピリオド）または#（シャープ））

：

ピースの数n

1番目のピースの配列のサイズh1（縦） w1（横）（整数；半角空白区切り）

1番目のピースの配列を表す1行w1文字の文字列h1行

：

プレイヤーの人数p

1番目のプレイヤーの選択したピースの情報 k1 t1 t2 ... tk（整数；半角空白区切り）

k1 : そのプレイヤーが選択したピースの数

t1~tk : プレイヤーの選択した 個のピースの番号

：

出力

1番目のプレイヤーが選択したピースの正否YESまたはNO

2番目のプレイヤーが選択したピースの正否YESまたはNO

：

p番目のプレイヤーが選択したピースの正否YESまたはNO

出力例

YES

NO

YES

NO

YES

NO

NO

YES