



数当て (Guess Them All)

あなたは JOI ちゃんと数当てゲームをしている。まず、JOI ちゃんは 1 以上 N 以下の整数 1 つずつをある順番に並べ替えた数列を用意する。この数列を「正解」と呼ぶことにしよう。あなたは「予想」という行動を L 回まで行うことができ、 L 回以内に正解を当てることができればあなたの勝ちである。

1 回の「予想」では、あなたは 1 以上 N 以下の整数 N 個を並べた数列を JOI ちゃんに伝える。予想の数列には、同じ整数が複数個含まれていてもかまわない。1 回の予想ごとに、予想の数列と正解とが何か所で一致したかを知ることができる。

あなたはこのゲームに勝つため、プログラムを書くことにした。

課題

正解の数列を指定された回数以下の予想で当てるプログラムを作れ。

制限

数列の長さ N と予想回数の上限 L は「採点基準」の項目で述べる。

入力

標準入力から以下の入力を読み込め。

- 1 行目には整数 N が書かれている。

やりとり

入力データを読み込んだ後、あなたのプログラムは、標準出力への予想の書き出しと、標準入力からの回答の読み込みを交互に行わなければならない。各予想は、予想した数列の i 番目の整数を i 行目とした N 行を標準出力に書き出さなければならない。予想には 1 より小さい整数や N より大きい整数が含まれていてはならない。各回答は、予想の数列と正解の数列が何か所で一致したかを表す 1 個の整数からなる 1 行として、標準入力から与えられる。回答として N が返ってきた際は、数列を当てることに成功したということなので、その時点でプログラムを終了し、その後一切の出力を行ってはならない。

注意

全ての採点用データについて、適切なアルゴリズムを用いれば、正解の数列が 1 以上 N 以下の整数のどのような並べ替えであっても、 L 回以下の予想で当てることができる。



重要な注意

標準入力を読み込むときに、読み込みが誤ってブロックされてしまうことを防ぐ必要がある。そのため、読み込みを行う前に `fflush(stdout);` などを記述しておくことを勧める。以下が採点システムとのやりとりを行うサンプルプログラムである。

```
#include <stdio.h>
int main(void) {
    int i, N, ret;
    scanf("%d", &N);
    for (;;) {
        for (i = 1; i <= N; i++) {
            printf("%d\n", i);    // 予想
        }
        fflush(stdout);          // 回答を読む前に標準出力をフラッシュする
        scanf("%d", &ret);       // 採点システムからの回答を読む
        if (ret == N) break;
    }
    return 0;
}
```

採点基準

採点用データのうち、配点の 10%分については、 $1 \leq N \leq 7, L = 6000$ をみたとす。

採点用データのうち、配点の 40%分については、 $1 \leq N \leq 50, L = 3000$ をみたとす。

採点用データのうち、配点の 30%分については、 $1 \leq N \leq 100, L = 1000$ をみたとす。

採点用データのうち、配点の 20%分については、 $1 \leq N \leq 100, L = 700$ をみたとす。



入出力の例

入力例	出力例
3	3 2 1
1	1 2 3
0	2 3 1
3	

この例では 3 回の予想で正解に達している.



カードキー (Keycards)

JOI の春合宿が行われる施設の宿泊棟の部屋の鍵は、カードに穴がいくつか開いた形状をしている。穴を開ける位置の候補は N 個あり、これらのうちいくつかに穴を開けた 2^N 枚の異なる鍵が作られた。

あなたは、JOI の春合宿のための 1 枚以上 2^N 枚以下の鍵をまとめて受け取った。穴を開ける候補の位置を合わせて鍵を重ねてみたあなたは、ちょうど K か所、受け取ったすべての鍵に穴が開いている位置があることに気づいた。

このようなことが起こるような、受け取った鍵の組み合わせは何通りあるだろうか。答えを 1 000 000 007 (素数) で割った余りを求めよ。

課題

N と K が与えられたとき、答えを 1 000 000 007 で割った余りを求めるプログラムを作成せよ。

制限

$1 \leq N \leq 1\,000\,000$ 穴を開ける位置の候補の個数

$0 \leq K \leq N$ 受け取ったすべての鍵に穴が開いている位置の個数

入力

標準入力から以下の入力を読み込め。

- 1 行目には整数 N, K が空白を区切りとして書かれている。

出力

標準出力に鍵の組み合わせの個数を表す 1 行を出力せよ。出力の 1 行目には、答えを 1 000 000 007 で割った余りを書け。

採点基準

採点用データのうち、配点の 30% 分については、 $N \leq 10$ を満たす。

採点用データのうち、配点の 70% 分については、 $N \leq 1\,000$ を満たす。



入出力の例

入力例 1	出力例 1
3 3	1

$N = 3$ の時、鍵は全部で 8 枚存在するが、ここでは以下のように鍵に名前をつける。

穴の開き方	名前
	ϕ
	A
	B
	C
	AB
	BC
	AC
	ABC

たとえば ABC という 1 枚の鍵だけを受け取った場合、受け取ったすべての鍵に穴が開いている位置の数は 3 か所である。また、それ以外の受け取り方では、受け取ったすべての鍵に穴が開いている位置の数が 3 か所になることはない。したがって、条件を満たす鍵の受け取り方は 1 通りである。

入力例 2	出力例 2
3 2	6

{AB, ABC} という 2 枚の鍵を受け取った場合、受け取ったすべての鍵に穴が開いている位置の数は 2 か所である。同様に、{AC, ABC}, {BC, ABC}, {AB}, {AC}, {BC} という受け取り方をした場合にも、受け取ったすべての鍵に穴が開いている位置の数は 2 か所となる。

入力例 3	出力例 3
3 1	30

条件を満たす鍵の受け取り方は、{A}, {A, AB}, {A, AC}, {A, ABC}, {A, AB, AC}, {A, AB, ABC}, {A, AC, ABC}, {A, AB, AC, ABC}, {AB, AC}, {AB, AC, ABC} など 30 通りである。

入力例 4	出力例 4
3 0	218



しりとり (Shiritori)

JOI国では、100種類の文字が使われている。これらの文字は、コンピュータ上では直接表すことが難しいので、代わりに以下の表記が用いられる。

00	01	02	03	04	05	06	07	08	09
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

すなわち、数字2つによる 10×10 通りで表される。JOI国の辞書では、この表によって定まる文字の順番によって単語を並べている。表の上の行にある文字がより早く、同じ行の文字ではより左の文字が早い。

さて、JOI国では今、しりとりが一大ブームである。しりとりとは、参加者が順番に、前の人が出した単語の最後の文字から始まる単語を言っていくゲームである。一度言われた単語を使うことはできない。

ある日、あなたは友達と「5しりとり」で遊んでいた。「5しりとり」では、通常のしりとりのルールに加え、用いる単語はすべて5文字でなければならない。あなたは「5しりとり」で言われた N 個の単語のリストをコンピュータに記録していたのだが、誤って並べ替えてしまった。そこで、単語のリストから「5しりとり」の様子を復元したい。

課題

単語のリストが与えられたとき、「5しりとり」の様子を復元するプログラムを作成せよ。

制限

$1 \leq N \leq 500\,000$ 単語の個数

入力

標準入力から以下の入力を読み込め。

- 1行目には整数 N が書かれており、単語の個数を表す。
- 続く N 行には各単語が1行に1つずつ書かれている。各単語は数字ちょうど10個からなる文字列で表される。単語は辞書に掲載されている順に入力に書かれており、すべての単語は異なる。



出力

与えられた N 個の単語を用いた「5 しりとり」が不可能である場合、**impossible** と 1 行に出力せよ。
「5 しりとり」が可能である場合、用いられる N 個の単語を 1 行 1 つずつ出力せよ。可能な「5 しりとり」が複数考えられるときは、以下の条件を満たすものを選べ。

- 1 番目の単語が辞書に最も早く掲載されているもの。
- 以上で定まらない場合は、2 番目の単語が辞書に最も早く掲載されているもの。
- \vdots
- 以上で定まらない場合は、 N 番目の単語が辞書に最も早く掲載されているもの。

採点基準

採点用データのうち、配点の 65% 分については、現れる文字は **00** から **19** までの 20 種類に限られる。
採点用データのうち、配点の 65% 分については、 $N \leq 1000$ を満たす。
採点用データのうち、配点の 40% 分については、これら 2 つの条件の両方を満たす。
採点用データのうち、配点の 90% 分については、これら 2 つの条件の少なくとも一方を満たす。

入出力の例

入力例 1	出力例 1
5	0000010201
0000010201	0102030403
0102030403	0308090002
0104050603	0206070801
0206070801	0104050603
0308090002	

この例では、以下の 2 通りの「5 しりとり」が考えられる。

- 0000010201 → 0102030403 → 0308090002 → 0206070801 → 0104050603
- 0000010201 → 0104050603 → 0308090002 → 0206070801 → 0102030403

1 番目の単語は同じであり、2 番目の単語が辞書に掲載されている順番を比較して、前者を出力する。



入力例 2	出力例 2
4 9600000098 9700000099 9800000099 9900000098	impossible

この例では、可能な「5 しりとり」は存在しない。

入力例 3	出力例 3
12 0114090401 0214051905 0304141219 0510031717 0703050011 1102190101 1108040907 1110090702 1313071203 1707120711 1902090011 1909121313	1909121313 1313071203 0304141219 1902090011 1108040907 0703050011 1110090702 0214051905 0510031717 1707120711 1102190101 0114090401