

**Xmas Contest 2014**

**Problems**

## Problem A: A+B Problem

Points: 100

くろうさ「足し算は甘え」  
しろうさ「……え？」  
くろうさ「掛け算と割り算は許してあげる」  
しろうさ「えっとえっと」  
くろうさ「ゼロと比較して分岐してもいいよ」  
しろうさ「引き z」  
くろうさ「というわけががんばってねっ」

### Input

この問題には入力ファイルは存在しない。

### Output (Special Judge)

1 以上 100 以下の整数 2 個の足し算を行うプログラムを以下の仕様に従って作成し提出せよ。後述のように、プログラムの行数が少ないほど高得点を得られる。

#### プログラムの動作

プログラムは、英大文字で表される 26 個の変数  $A, B, C, D, \dots, Z$  を扱うことができる。各変数は 0 以上  $2^{32}$  未満の整数を 1 個保持する。実行開始時に、変数  $A$  は整数  $A$ 、変数  $B$  は整数  $B$ 、変数  $C, D, \dots, Z$  は 0 で初期化される ( $A, B$  は実行ごとに異なる値である)。プログラムの終了時に変数  $C$  の値が  $A + B$  となっていることが目標である。

プログラムはいくつかの行からなり、上から下へ実行される。各行は以下の 4 種類のいずれかである：

- $z = x * y$   
 $z$  は変数、 $x$  と  $y$  は変数または定数である。この行が実行されると、 $x$  と  $y$  の積を  $2^{32}$  で割った余りを計算し、 $z$  に代入する。
- $z = x / y$   
 $z$  は変数、 $x$  と  $y$  は変数または定数である。この行が実行されると、 $x$  を  $y$  で割った商の整数部分を計算し、 $z$  に代入する。ただし、 $y$  の値が 0 である場合は実行時エラーとなる。
- `if (x) {`  
 $x$  は変数または定数である。この行が実行されると、 $x$  が 0 でないときは次の行に実行が移り、 $x$  が 0 であるときは対応する閉じ中括弧に実行が移る (中括弧に囲まれている部分を飛ばす)。
- `}`  
閉じ中括弧である (この行の実行は何も影響がない)。

### プログラムの形式

以下の条件を満たさない提出,あるいは前述の4種類のいずれでもない行を含む提出は,“PARSE ERROR”として不正解と判定される.

- 文字コードは ASCII, 改行コードは LF または CR+LF でなければならない.
- プログラムは 999 行以下でなければならない.
- 変数, 定数, =, \*, /, if, (, ), {, } の前後には, 0 個以上任意個の空白 (半角空白または水平タブ) を入れることができる. ただし, プログラムの各行は改行文字を除いて 80 文字以下でなければならない.
- 空行があってはならない.
- 定数は 0 以上  $2^{32}$  未満の整数であり, 十進表記で書かれなければならない.
- 開き中括弧と閉じ中括弧は, 適切に対応がとれていなければならない.

### 採点

提出されたプログラムが正しい形式であるとき, プログラムは  $1 \leq A \leq 100, 1 \leq B \leq 100$  を満たす各整数の組  $(A, B)$  に対して独立に実行されテストされる.

プログラムが 0 による除算を行った場合, “RUNTIME ERROR” として不正解と判定される.

ある  $(A, B)$  に対して, 実行終了時に変数  $C$  の値が  $A + B$  でない場合, “WRONG ANSWER” として不正解と判定される.

すべての  $(A, B)$  に対して, プログラムが 0 による除算を行わずに終了し, 実行終了時に変数  $C$  の値が  $A + B$  である場合, 正解 (“ACCEPTED”) と判定される. プログラムの行数が  $p$  である場合, 正解と判定されたときの得点は  $100 - \left\lfloor \frac{1}{10} p \right\rfloor$  点となる.

“PARSE ERROR”, “RUNTIME ERROR”, “WRONG ANSWER”, “ACCEPTED” の判定およびエラーメッセージは, コンテストシステムで提出の詳細から確認できる.

### 補助

<http://hos.ac/contest/xmas2014/aplusb.html> にて, JavaScript による簡易インタプリタが利用可能である. ただし, このインタプリタはほとんどのエラーチェックを行わないことに注意せよ.

### Sample

#### Sample Output

---

```
C = A * A
if (D) {
  D = 2014 * B
  if (0) {
    C = D / 4000000000
  }
}
```

---

これは, 形式が正しいプログラムの例である.

## Problem B: Bad Sort?

Points: 20 + 20 + 60

うさぎは効率の良いソートアルゴリズムを講義で習ったが、冬休みになったので詳細を忘れてしまった。うさぎががんばって思い出したアルゴリズムは次のようなものであった。以下に示される関数  $\text{Sort}$  は、入力として受け取った数列  $x_1, \dots, x_n$  を非減少列になるように並べ替えた数列を返す。

$\text{Sort}(x_1, \dots, x_n)$ :

$n = 0$  ならば空の数列を返す。

$p$  を  $1, \dots, n$  からランダムに選ぶ。どの値も確率  $\frac{1}{n}$  で選ばれる。

空の配列  $A, B$  を用意する。

$i = 1, \dots, n$  に対して、順に以下を行う：

$i = p$  のとき、何もしない。

そうでないとき、

$x_i < x_p$  ならば、 $A$  の末尾に  $x_i$  を加える。

そうでなければ、 $B$  の末尾に  $x_i$  を加える。

$\text{Sort}(A), (x_p), \text{Sort}(B)$  をこの順に連結して返す。

果たしてこれは効率が良いものであろうか。アルゴリズムの性能を見積もるため、うさぎは関数  $\text{Sort}$  の呼び出しが発生する度に時間がかかると考え、入力の長さが  $n$  のときのその時間を  $n$  単位であると考えた。例えば、数列  $(2, 3, 1)$  をソートするために  $\text{Sort}(2, 3, 1), \text{Sort}(2, 1), \text{Sort}(), \text{Sort}(2), \text{Sort}(), \text{Sort}(), \text{Sort}()$  という呼び出しが発生したとすると、時間は  $3 + 2 + 0 + 1 + 0 + 0 + 0 = 6$  かかったと考える。

入力として数列  $(X_1, \dots, X_N)$  が与えられたとき、かかる時間の期待値を求めたい。ただし、誤差などの問題を避けるため、後述の出力形式に対応したプログラムを作成せよ。

### Input

入力の 1 行目にはテストケースの個数  $T$  が書かれている。その後次の形式で各テストケースが与えられる。

$N$

$X_1 \cdots X_N$

$N$  および  $X_1, \dots, X_N$  はすべて整数である。

### Output

$P = 10007$  とする。各テストケースに対して、求める期待値を既約分数で  $\frac{a}{b}$  と表したとき、 $a - br$  が  $P$  で割り切れるような  $0$  以上  $P$  未満の整数  $r$  を 1 行に出力せよ。制約によって、出力すべき  $r$  がちょうど 1 個存在することが証明できる。

## Constraints

すべての入力において, 以下の制約が満たされる.

- $1 \leq X_i \leq N$  ( $1 \leq i \leq N$ ).

さらに, 各入力において, 以下の制約が満たされる.

### B1.in (20 点)

- $T = 10$ .
- $1 \leq N \leq 10$ .

### B2.in (20 点)

- $T = 20$ .
- $1 \leq N \leq 200$ .

### B3.in (60 点)

- $T = 40$ .
- $1 \leq N \leq 5\,000$ .

## Sample

Sample Input	Sample Output
5	3
2	6677
1 2	8018
3	8018
2 3 1	8018
5	
1 2 3 4 5	
5	
5 4 2 1 3	
5	
4 3 2 5 1	

それぞれのテストケースに対して, かかる時間の期待値を既約分数で表すと,  $\frac{3}{1}, \frac{17}{3}, \frac{62}{5}, \frac{62}{5}, \frac{62}{5}$  となる.

## Problem C: Count Your Ways

Points: 20 + 80

うさぎの国には、 $1, \dots, N$  の番号がついた  $N$  個の街がある。  $M$  本の一方通行の道があり、 $i$  番目の道は始点が街  $A_i$ 、終点が街  $B_i$  である。うさぎは長旅が好きである。ある街から出発し、道をちょうど 2014122419 回使い、出発した街へ戻ってくる旅程は何通りあるだろうか。2 個の旅程は、使う道の列が一致するとき同じと考える (通る街が同じでも、異なる道を使っていれば異なる旅程と考える)。答えが非常に大きくなる予感を抱えたうさぎは、旅程の個数を 2014122419 で割った余りを求めることにした。

### Input

入力の 1 行目にはテストケースの個数  $T$  が書かれている。その後次の形式で各テストケースが与えられる。

$$\begin{array}{l} N \ M \\ A_1 \ B_1 \\ \vdots \\ A_M \ B_M \end{array}$$

$N, M$  および  $A_1, \dots, A_M, B_1, \dots, B_M$  はすべて整数である。

### Output

各テストケースに対して、条件を満たす旅程の個数を 2014122419 で割った余りを 1 行に出力せよ。

### Constraints

すべての入力において、以下の制約が満たされる。

- $1 \leq A_i \leq N, 1 \leq B_i \leq N (1 \leq i \leq M)$ .

さらに、各入力において、以下の制約が満たされる。

#### C1.in (20 点)

- $T = 10$ .
- $1 \leq N \leq 100$ .
- $1 \leq M \leq 100$ .

#### C2.in (80 点)

- $T = 20$ .
- $1 \leq N \leq 10\,000$ .
- $1 \leq M \leq 10\,000$ .

## Sample

Sample Input	Sample Output
5	0
2 1	0
1 2	0
2 2	0
1 2	0
1 2	
3 3	
1 2	
2 3	
3 1	
3 6	
1 2	
1 3	
2 1	
2 3	
3 1	
3 2	
5 8	
1 2	
2 1	
2 1	
2 1	
1 3	
3 5	
3 5	
5 1	

5 番目のテストケースにおいては、例えば、街 1 から出発し、5 番目、6 番目、8 番目の道を使い、その後 1 番目の道と 3 番目の道を交互に 1007061208 回ずつ使って街 1 に戻ってくる、という旅程が条件を満たす。

## Problem D: Distinguished Furry Animals

Points: 30 + 70

「うさみみの館」は、現在建設中の謎の施設である。館は 1 から  $n$  までの番号のついた  $n$  個の部屋をもち、それぞれの部屋の床にはうさぎの絵が描かれている。部屋  $i$  のうさぎの絵の左耳の部分に乗ると、部屋  $l_i$  に描かれたうさぎのしっぽの上にテレポートする。同様に、部屋  $i$  のうさぎの絵の右耳の部分に乗ると、部屋  $r_i$  に描かれたうさぎのしっぽの上にテレポートする。

こんな不思議な館であるが、部屋の個数  $n$  とテレポートの設定方法  $l_1, \dots, l_n, r_1, \dots, r_n$  を具体的にどのような値にするかは未定である。ただし、 $n$  が正の整数であることと、 $1 \leq i \leq n$  に対し  $1 \leq l_i \leq n, 1 \leq r_i \leq n$  であることは決まっている。

うさぎたちは、館の完成記念パーティーに、ねこときつねを招くことにした。まずは館の御披露目として、ねこときつねにテレポートを体験してもらうというイベントがある。ねこときつねはそれぞれ部屋 1 からスタートし、耳に乗ってテレポートすることを何回か繰り返す。

実は、ねこもきつねも耳への乗り方を既に決めていているらしい。耳への乗り方は文字 L, R からなる文字列で表される。文字列の長さはテレポートの回数を表し、 $i$  文字目が L であれば  $i$  回目に左耳に乗ること、 $i$  文字目が R であれば  $i$  回目に右耳に乗ることを表す。ねこの計画は文字列  $C$  で、きつねの計画は文字列  $F$  で表される。例えば  $C$  が RRL であれば、ねこは、部屋 1 からスタートし、

1. 右耳に乗って部屋  $r_1$  にテレポートし、
2. 右耳に乗って部屋  $r_{r_1}$  にテレポートし、
3. 左耳に乗って部屋  $l_{r_{r_1}}$  にテレポートし、

最終的に部屋  $l_{r_{r_1}}$  に辿り着く。

さて、面白いテレポートの設定にするため、ねこときつねが最終的に辿り着く部屋が異なるようにしたい。そのために必要な館の部屋の個数  $n$  の最小値はどのくらいだろうか。

### Input

入力の 1 行目にはテストケースの個数  $T$  が書かれている。その後次の形式で各テストケースが与えられる。

$$C F$$

$C, F$  は文字 L, R からなる文字列である。

### Output

各テストケースに対して、ねこときつねが最終的に辿り着く部屋が異なるようにできるような  $n$  の最小値を 1 行に出力せよ。そのような  $n$  が存在しない場合には代わりに  $-1$  を出力せよ。



## Constraints

各入力において、以下の制約が満たされる。  $|w|$  は文字列  $w$  の長さを表す。

### D1.in (30 点)

- $T = 100$ .
- $C, F$  は文字 L のみからなる。
- $1 \leq |C| \leq 1000, 1 \leq |F| \leq 1000$ .

### D2.in (70 点)

- $T = 100000$ .
- $1 \leq |C| \leq 18, 1 \leq |F| \leq 18$ .

## Sample

Sample Input	Sample Output
6	2
L LL	3
LL LLLL	2
LRLL LLR	3
LRRL RLRL	2
RLRRL LRLRR	3
LRL LLLRL	

6 番目のテストケースにおいて、例えば、部屋の個数を  $n = 3$ 、テレポートの設定を  $l_1 = 2, l_2 = 3, l_3 = 1, r_1 = 1, r_2 = 2, r_3 = 3$  とする。ねこは、部屋 1 からスタートし、

1. 左耳に乗って部屋  $l_1 = 2$  にテレポートし、
2. 右耳に乗って部屋  $r_2 = 2$  にテレポートし、
3. 左耳に乗って部屋  $l_2 = 3$  にテレポートし、

最終的に部屋 3 に辿り着く。一方きつねは、部屋 1 からスタートし、

1. 左耳に乗って部屋  $l_1 = 2$  にテレポートし、
2. 左耳に乗って部屋  $l_2 = 3$  にテレポートし、
3. 左耳に乗って部屋  $l_3 = 1$  にテレポートし、
4. 右耳に乗って部屋  $r_1 = 1$  にテレポートし、
5. 左耳に乗って部屋  $l_1 = 2$  にテレポートし、

最終的に部屋 2 に辿り着く。このようにして、ねこときつねが最終的に辿り着く部屋が異なるようにできる。

## Problem E: Economic Plan

Points: 20 + 80

うさぎの国には  $N$  個の空港があり、1 から  $N$  までの番号がついている。現在、 $1 \leq a < b \leq N$  を満たすすべての整数の組  $(a, b)$  に対して、空港  $a$  と空港  $b$  を双方向に結ぶ航空便がある。したがって、全部で  $\frac{1}{2}N(N-1)$  便がある。

航空便が多すぎて手に負えないと判断したうさぎの国の政府は、この度、維持費節約のために航空便を一部廃止することになった。まず、利用者が極端に少ない  $M$  便があったため、それらの廃止を決定した。  $M$  便のうち  $i$  番目は空港  $A_i$  と空港  $B_i$  を結んでいた ( $1 \leq i \leq M$ )。これら  $M$  便の廃止後も、どの空港に対しても、その空港と他の空港を結ぶ航空便が  $N-1$  便のうち半分は残ることがわかっている。

うさぎの王様は、さらなる航空便の廃止を行いたい。政府の事情に詳しくない王様は、政府に対して「追加あと  $k$  便を廃止せよ」という命令を出し、具体的にどの  $k$  便を廃止するかは政府に一任することにした。ここで  $k$  は  $0 \leq k \leq \frac{1}{2}N(N-1) - M$  を満たす整数である。

王様は、どの空港からどの空港へも何本かの航空便を乗り継いで行くことができる状態を保ちたい、と考えた。政府は、残る  $\frac{1}{2}N(N-1) - M$  便のうちどの  $k$  便を廃止するかわからない。政府がどんな横暴に及んでもこの状態が必ず保たれるような  $k$  の最大値はどれくらいだろうか。

### Input

入力の 1 行目にはテストケースの個数  $T$  が書かれている。その後次の形式で各テストケースが与えられる。

$$\begin{array}{l} N \ M \\ A_1 \ B_1 \\ \vdots \\ A_M \ B_M \end{array}$$

$N, M$  および  $A_1, \dots, A_M, B_1, \dots, B_M$  はすべて整数である。

### Output

各テストケースに対して、どの空港からどの空港へも何本かの航空便を乗り継いで行くことができる状態が必ず保たれるような  $k$  の最大値を 1 行に出力せよ。そのような  $k$  が存在しない場合には代わりに  $-1$  を出力せよ。

## Constraints

すべての入力において、以下の制約が満たされる。

- $1 \leq A_i < B_i \leq N$  ( $1 \leq i \leq M$ ).
- $(A_i, B_i) \neq (A_j, B_j)$  ( $1 \leq i < j \leq M$ ).
- 各整数  $c$  ( $1 \leq c \leq N$ ) に対し、 $A_i = c$  または  $B_i = c$  となる  $i$  ( $1 \leq i \leq M$ ) の個数は  $\frac{1}{2}(N-1)$  未満である。

さらに、各入力において、以下の制約が満たされる。

### E1.in (20 点)

- $T = 10$ .
- $1 \leq N \leq 100$ .
- $1 \leq M \leq 100$ .

### E2.in (80 点)

- $T = 20$ .
- $1 \leq N \leq 10\,000$ .
- $1 \leq M \leq 10\,000$ .

## Sample

Sample Input	Sample Output
2	1
4 2	2
1 3	
2 4	
6 6	
1 2	
2 3	
3 4	
4 5	
5 6	
1 6	

1 番目のテストケースにおいては、もし  $k = 2$  としてしまうと、例えば、空港 1, 4 を結ぶ便と空港 2, 3 を結ぶ便を政府が追加で廃止した場合に、空港 1 から空港 4 へ航空便を乗り継いで行くことができなくなってしまふ。

**Problem F: Fake Distribution**

Points: 30 + 70

うさぎたちの間では、クリスマスプレゼントとして乱数列を交換するのが流行である。しかし、実数の開区間  $(0, 1)$  に一様分布する乱数の生成器 (R と呼ばれる) は誰もが持っているので、もっと面白い分布をする乱数列をプレゼントするのが乙である。

しろうさは、 $s, t$  を R で独立に生成した 2 個の値とするとき、値

$$\sqrt{-\log s} \cdot \sin(2\pi t)$$

の分布が綺麗だと考えた。

くろうさは、 $u_1, u_2, u_3, u_4, u_5, u_6$  を R で独立に生成した 6 個の値とするとき、値

$$u_1 + u_2 + u_3 + u_4 + u_5 + u_6 - 3$$

の分布がかっこいいと考えた。

さて、しろうさの考えた分布とくろうさの考えた分布は似ているという噂が立っているらしい。テストケースごとに、しろうさの分布とくろうさの分布のうち一方が選ばれ、その分布に従った  $N$  個の値  $X_1, \dots, X_N$  が与えられる。果たしてどちらの分布が選ばれたか判定できるだろうか。

**Input**

入力の 1 行目にはテストケースの個数  $T$  が書かれている。その後次の形式で各テストケースが与えられる。

$$N \\ X_1 \cdots X_N$$

値の個数  $N$  は整数である。  $X_1, \dots, X_N$  は実数であり、小数点以下 10 桁まで書かれている。

**Output (Special Judge)**

各テストケースに対して、  $X_1, \dots, X_N$  がしろうさの分布のものからと判断した場合は文字列 **White** を、くろうさの分布のものからと判断した場合は文字列 **Black** を 1 行に出力せよ。

すべてのテストケースに対して **White** または **Black** と答えていて、かつ、 $\frac{4}{5}T$  個以上のテストケースに正しく答えた場合、正解と判定される。

## Constraints

すべての入力において, 以下の制約が満たされる.

- $T$  個のテストケースのうち,  $\frac{1}{2}T$  個においては  $X_1, \dots, X_N$  はしろうさの分布から生成された値であり, 残りの  $\frac{1}{2}T$  個においては  $X_1, \dots, X_N$  はくろうさの分布から生成された値である.

さらに, 各入力において, 以下の制約が満たされる.

F1.in (30 点)

- $T = 20$ .
- $N = 10\,000$ .

F2.in (70 点)

- $T = 100$ .
- $N = 2\,000$ .

## Sample

Sample Input	Sample Output
2	White
3	Black
0.1207533857 -0.0740957678 -0.5254538066	
3	
0.3057030458 0.7705966469 -0.0639899533	

## Problem G: Geometry Lover

Points: 30 + 70

うさぎは幾何の問題が大好きなので、クリスマスの飾り付けで余ったゴムひもで遊んでいた。

直方体状の部屋があり、3本の非常によく伸びるゴムひもがある。うさぎはこれから、各ゴムひもの一方の端を左の壁に、もう一方の端を右の壁に固定しようとしている。ゴムひもを固定する点は相異なる。ゴムひも同士は絡み合っているかもしれない。各ゴムひもを左の壁から右の壁まで辿るとき、常に左から右へ進む(右から左へ進む部分はない)。

うさぎは、2通りのゴムひもの配置を考えた。これらを天井から見た様子がアスキーアート  $S_1, S_2$  として入力に与えられる。これら2通りは実は同一の配置かもしれない。すなわち、ゴムひもの端を固定した後、ゴムひもを伸ばしたり縮めたり絡ませたり解いたりして一方から他方を得ることができるかもしれない。たとえば、以下に示される2通りは同一の配置である。

```

--  -----  --          -----  -----
      \ /      \ /          \ /
      \      \      \      \
      / \      / \      / \
--  --  --  --  --  =  --  --  --  --  --
      \ /      \ /      \ /      \ /
      \      \      \      \
      / \      / \      / \
-----  -----          --  -----  --

```

というわけで、与えられた2通りのゴムひもの配置が同一かどうか判定してほしい。

### Input

入力の1行目にはテストケースの個数  $T$  が書かれている。その後次の形式で各テストケースが与えられる。

```

N1 N2
S1
S2

```

$N_i$  ( $i = 1, 2$ ) は整数である。 $S_i$  ( $i = 1, 2$ ) は長さ  $N_i$  の文字列9行であり、文字  $.$  (半角ピリオド),  $-$  (半角ハイフン),  $\backslash$  (半角バックスラッシュ),  $/$  (半角スラッシュ) からなる。

### Output

各テストケースに対して、与えられた2通りのゴムひもの配置が同一ならば文字列 YES を、そうでない場合は文字列 NO を1行に出力せよ。

## Constraints

すべての入力において, 以下の制約が満たされる.

- $S_1, S_2$  は, 以下の 5 種類のブロックを横に並べたものである.

-	-...-	-...-	-----	-----
.	.\./.	.\./.	.....	.....
.	..\..\	../..\	.....	.....
.	./.\.	./.\.	.....	.....
-	-...-	-...-	-...-	-...-
.	.....	.....	.\./.	.\./.
.	.....	.....	..\..\	../..\
.	.....	.....	./.\.	./.\.
-	-----	-----	-...-	-...-

文字 `.` は空白の代わりである. 右の 4 種類は, 天井から見たときのゴムひもが交差する様子を表す. 3 行目または 7 行目の 3 文字目は, 上側に見えるゴムひもの進行方向を示す.

さらに, 各入力において, 以下の制約が満たされる.

### G1.in (30 点)

- $T = 20$ .
- $1 \leq N_i \leq 50$  ( $i = 1, 2$ ).

### G2.in (70 点)

- $T = 40$ .
- $1 \leq N_i \leq 5000$  ( $i = 1, 2$ ).

Sample

Sample Input	Sample Output
2	YES
17 17	NO
<pre> --...-----...-- ..\.\/.....\.\/.. ...\.\/.....\.\/.. ..\/.\/.....\/.\/.. --...-----...-- .....\.\/..... .....\.\/..... .....\/.\/..... -----...----- -----...----- .....\.\/..... .....\.\/..... .....\/.\/..... --...-----...-- ..\.\/.....\.\/.. ...\.\/.....\.\/.. ..\/.\/.....\/.\/.. --...-----...-- </pre>	
10 20	
<pre> ----- ..... ..... ..... ----- ..... ..... ..... ----- -----...----- .....\.\/.....\.\/..... .....\/.\/.....\/.\/..... .....\/.\/.....\/.\/..... -----...----- ..... ..... ..... ----- </pre>	



## Problem H: Hack Me

Points: 10 + 90

### Input

あるテキストファイルを, ある秘密の方法で暗号化したファイルが与えられる.  
本問に限り, 入力はテキストファイルではなくバイナリファイルである (ただし実行可能ファイルではない).

### Output

元のテキストファイルを出力せよ.

### Constraints

各入力において, 以下の制約が満たされる.

#### H1.in (10 点)

- 元のテキストファイルは, Xmas Contest 2014 の A, B, C, D, E, F, G 問題のいずれかの Sample Output である.

#### H2.in (90 点)

- 元のテキストファイルは 32 行からなり, 各行には 1 個の整数のみが不要な先頭の 0 のない十進表記で書かれている. 各整数は 0 以上  $2^{32}$  未満である.