



## 入室管理 (Management)

JOI 社では入室管理を行うため、入出者の名前をアルファベットで記録した。しかし、大文字と小文字が混在してしまい管理記録は読みづらいものとなってしまった。そこで、入室管理記録を読みやすくするため、記録されている名前を全て小文字に変更することにした。

### 課題

入室管理記録を読みやすくするため、記録されている名前を全て小文字に変更するプログラムを作成せよ。ただし、入室記録には同じ名前の人が複数出てくることもある。

### 制限

$1 \leq N \leq 1000$  名前の数

$S_i$   $i$  人目の入室者の名前を表す半角英字 (半角大文字のアルファベット A,B,C,...,Z または半角小文字のアルファベット a,b,c,...,z) のみからなる 1 文字以上 20 文字以下の文字列

### 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。
- 続く  $N$  行には、1 行につき 1 回の入室管理記録について記述している。これらの行のうちの  $i$  行目は  $i$  人目の入室者の名前を表す文字列  $S_i$  が書かれている。

### 出力

出力は標準出力に行うこと。出力の  $i$  行目には  $i$  人目の入室者の名前を重複を含め全て小文字で出力せよ。

### 入出力例 1

入力例	出力例
3	watanabe
WatanabE	ito
IT0	yamamoto
YamaMoto	



## 入出力例 2

入力例	出力例
4	suzuki
SUZUKI	tanaka
tanaka	takahashi
tAkAhAshi	suzuki
SuZuKi	



## 合計 (Sum)

### 課題

あなたには  $N$  個の整数  $A_1, \dots, A_N$  が順番に与えられる。整数が与えられる度に、それまでに与えられた整数の合計を出力するプログラムを作成せよ。

### 制限

$1 \leq N \leq 100$  あなたに与えられる整数の個数

$0 \leq A_i \leq 10$   $i$  番目に与えられる整数

### 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、標準入力からの整数の読み込みと、標準出力への合計の書き出しを交互に行わなければならない。

$i$  番目 ( $1 \leq i \leq N$ ) の整数を読み込んだら、1 番目から  $i$  番目までの整数の合計を標準出力に書き出せ。これを  $N$  回繰り返せ。最後に  $N$  番目の整数を読み込み 1 番目から  $N$  番目までの整数の合計を出力したら、プログラムを終了せよ。1 番目から  $N$  番目までの整数の合計を出力した後は、一切の出力を行ってはならない。

### 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, a;
    scanf("%d", &N);
```



```
for (i = 1; i <= N; i++) {  
    scanf("%d", &a);    // 採点システムからの入力を読む  
    printf("%d\n", 0); // 回答する  
    fflush(stdout);    // 次の入力を読む前に標準出力をフラッシュする  
}  
return 0;  
}
```

## 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。

## 入出力例

入力例	出力例
7	
3	
	3
5	
	8
0	
	8
2	
	10
0	
	10
10	
	20
1	
	21

この入出力例において、あなたには3, 5, 0, 2, 0, 10, 1の7個の整数が順番に与えられる。1番目の整数3が与えられたら、3を出力せよ。2番目の整数5が与えられたら、 $3 + 5 = 8$ を出力せよ。3番目の整数0が与えられたら、 $3 + 5 + 0 = 8$ を出力せよ。これを繰り返し、最後に7番目の整数1が与えられたら、7個の整数の合計21を出力し、プログラムを終了せよ。



## 時間 (Time)

あなたは 2038 年問題を知っているだろうか。C 言語の一般的な実装では時刻を 1970 年 1 月 1 日からの経過秒数で表す。しかし格納する型が伝統的に符号付き 32 ビット整数であるために  $2^{31} - 1$  秒までしか表すことができない。この  $2^{31} - 1$  秒をこえるのが 2038 年の 1 月 19 日 12 時 14 分 8 秒なのである。これを解決する方法は 64 ビット整数を使うことである。

そこで、64 ビット整数で経過秒数  $S$  が与えられた時に、経過日数を小数点以下を四捨五入して整数で答えるプログラムを作成することにした。ただし、うるう秒などは無視してよく、1 日を 86400 秒として四捨五入せよ。

## 課題

64 ビット整数で経過秒数  $S$  が与えられた時に、経過日数を小数点以下を四捨五入して整数で出力するプログラムを作成せよ。

## 制限

$$1 \leq S \leq 9\,223\,372\,036\,854\,775\,807 = 2^{63} - 1 \quad \text{1970 年 1 月 1 日からの経過秒数}$$

## 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  $S$  が書かれている。

## 出力

経過日数を小数点以下を四捨五入した整数を標準出力に出力せよ。

## 入出力例 1

入力例	出力例
86400	1



### 入出力例 2

入力例	出力例
43200	1

### 入出力例 3

入力例	出力例
2147483648	24855