

Torneo Argentino de Programación

17 de septiembre de 2016

- Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
- Escuela de Tecnologías de la Información y las Comunicaciones
Universidad Nacional de Chilecito
- Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional del Rosario
- Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
- Facultad de Ciencias Exactas y Tecnología
Universidad Nacional de Tucumán
- Facultad de Informática
Universidad Nacional de La Plata
- Facultad de Informática
Universidad Nacional del Comahue
- Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba
- Facultad Regional Resistencia
Universidad Tecnológica Nacional

Sesión de Práctica

Este conjunto contiene 2 problemas; las páginas están numeradas de 1 a 2.

Información General

Salvo indicación en contrario, lo siguiente vale para todos los problemas.

Entrada

1. La entrada se debe leer de la entrada estándar (standard input).
2. La entrada contiene un único caso de prueba, el cual se describe utilizando una cantidad de líneas que depende del problema. No hay otros datos en la entrada.
3. Cuando una línea de datos contiene varios valores, éstos se separan utilizando exactamente *un* espacio entre ellos. Ningún otro espacio aparece en la entrada. No hay líneas en blanco.
4. No hay letras con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ñ, é, È, ô, Ü, ç, etcétera).
5. Todas las líneas, incluyendo la última, tienen la marca usual de fin de línea.

Salida

1. La salida se debe escribir en la salida estándar (standard output).
2. El resultado del caso de prueba debe aparecer en la salida utilizando una cantidad de líneas que depende del problema. No debe haber otros datos en la salida.
3. Cuando una línea de resultados contiene varios valores, éstos se deben separar utilizando exactamente *un* espacio entre ellos. Ningún otro espacio debe aparecer en la salida. No debe haber líneas en blanco.
4. No debe haber letras con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ñ, é, È, ô, Ü, ç, etcétera).
5. Todas las líneas, incluyendo la última, deben tener la marca usual de fin de línea.
6. Para escribir números reales, redondearlos al racional más cercano con la cantidad de dígitos luego del punto decimal que se especifica en el enunciado. El caso de prueba es tal que no va a haber empates en el redondeo.

Tiempo límite

1. El tiempo límite informado corresponde a la entrada descrita en el enunciado, y *no* a múltiples instancias de la misma.

Problema A — Al escape del escape

AUTOR: PABLO ARIEL HEIBER - UNIVERSIDAD DE BUENOS AIRES

Un protocolo de comunicaciones es un conjunto de reglas para transmitir información en un sistema de comunicaciones. El trabajo de Alicia es escribir programas para implementar partes de dichos protocolos. Muchas veces es necesario transmitir secuencias de campos, y para saber dónde termina un campo y empieza el siguiente se intercala un separador entre cada par de campos consecutivos. Usar un separador simple como espacio, coma, o punto y coma, tiene el inconveniente de que a veces los campos a transmitir contienen esos mismos caracteres. La solución estándar para estos casos es insertar un carácter de “escape” justo antes de cada separador que aparece dentro de un campo, de modo de poder distinguirlo de un separador verdadero. Alicia cree que esta solución va a incrementar mucho la longitud de los datos a transmitir, por lo que decidió usar un separador lo suficientemente complejo como para que nunca esté contenido en los datos. Así piensa escapar de la ineficiente alternativa de escapar los separadores.

Para elegir el separador ideal, Alicia recopiló un *log*, que no es otra cosa que una cadena muy larga de caracteres representativa de los datos que su protocolo va a manejar. Luego de pensar un rato en el asunto, Alicia llegó a la conclusión de que cualquier cadena de caracteres no vacía que no aparezca dentro del log sería un separador aceptable para utilizar en su protocolo. Pero como a ella le interesa minimizar la longitud de los datos a transmitir, quiso saber la longitud mínima que un separador aceptable puede tener. De inmediato se puso a escribir un programa para calcular tal longitud, y lo está probando para el caso particular en el que tanto el log como los separadores aceptables contienen únicamente dígitos binarios (“0” y “1”). ¿Pueden ustedes anticipar los resultados?

Entrada

Una única línea que contiene el log, que es una cadena no vacía de a lo sumo 10^5 dígitos binarios.

Salida

Imprimir en la salida una línea conteniendo un entero que representa la longitud mínima de un separador aceptable para el log dado.

Entrada de ejemplo	Salida para la entrada de ejemplo
011101001	3

Entrada de ejemplo	Salida para la entrada de ejemplo
100010110011101	4

Entrada de ejemplo	Salida para la entrada de ejemplo
11111	1

Problema B — Boca de urna

AUTOR: LEOPOLDO TARAVILSE - UNIVERSIDAD DE BUENOS AIRES

Justo en este momento se están desarrollando las elecciones presidenciales en Nlogonia. Para que un candidato gane en primera vuelta debe obtener más votos que cada uno de los otros candidatos. Pero con eso no alcanza: además, debe obtener al menos el 45 % de todos los votos, o al menos el 40 % de todos los votos y al menos un 10 % más de votos que cada uno de los otros candidatos. Si ningún candidato gana en primera vuelta, se realiza una nueva elección en segunda vuelta.

Benicio es un periodista político de Nlogonia que siempre quiere tener la primicia. Por eso recolectó información de las encuestas de boca de urna, y quiere saber si de acuerdo a esos datos algún candidato gana en primera vuelta o, por el contrario, hay segunda vuelta. Benicio necesita decidir esto con urgencia antes de que alguien le saque la primicia. ¿Pueden ayudarlo?

Entrada

La primera línea contiene un entero N que indica la cantidad de candidatos ($2 \leq N \leq 10$). La segunda línea contiene N enteros V_i que representan las cantidades de votos obtenidos por cada uno de los candidatos ($0 \leq V_i \leq 1000$ para $i = 1, 2, \dots, N$). Al menos un candidato obtuvo al menos un voto y no hay dos candidatos con la misma cantidad de votos.

Salida

Imprimir en la salida una línea conteniendo un entero que representa si hay o no ganador en primera vuelta. Si hay ganador en primera vuelta el número debe ser un 1; caso contrario (es decir, si hay segunda vuelta) el número debe ser un 2.

Entrada de ejemplo	Salida para la entrada de ejemplo
2 60 40	1

Entrada de ejemplo	Salida para la entrada de ejemplo
3 16 28 21	1

Entrada de ejemplo	Salida para la entrada de ejemplo
3 42 23 35	2

Entrada de ejemplo	Salida para la entrada de ejemplo
3 297 302 401	2