

#### Indicaciones específicas:

- Esta evaluación contiene 6 páginas (incluyendo esta página) con 5 preguntas. El total de puntos son 34.
  - El tiempo límite para la evaluación es 100 minutos.
  - **Resolver solo 3 preguntas de las 5.**
  - Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
    - p1.cpp
    - p2.cpp
    - p3.cpp
    - p4.cpp
    - p5.cpp
  - Deberás subir estos archivos directamente a [www.gradescope.com](http://www.gradescope.com), uno en cada ejercicio. También puedes crear un .zip
- 

#### Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	6	
2	6	
3	6	
4	8	
5	8	
Total:	34	

1. (6 points) **Problema A****Producto unitario**

Van a recibir  $n$  números  $a_1, a_2, \dots, a_n$ . Con una moneda pueden realizar la siguiente operación:

- Seleccionar uno de estos números y sumarle o restarle 1.

En particular, podemos realizar esta operación a un mismo número cualquier cantidad (posiblemente 0) de veces.

Queremos alterar la lista de números de modo que al final el producto de todos ellos sea igual a 1, en otras palabras, que  $a_1 \cdot a_2 \cdot \dots \cdot a_n = 1$ .

Por ejemplo, para  $n = 3$  y números  $[1, -3, 0]$ , podemos obtener un producto igual a 1 con 3 monedas: agregar 1 al segundo elemento, agregar 1 al segundo elemento de nuevo, y restar 1 al tercer elemento. El arreglo se convierte en  $[1, -1, -1]$  y  $1 \cdot (-1) \cdot (-1) = 1$ .

¿Cuál es el mínimo costo que tenemos que pagar para realizar esto?

**Input** La primera línea contiene un entero positivo  $n$  ( $1 \leq n \leq 10^5$ ) - el número de elementos.

La segunda línea contiene  $n$  enteros  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ ) - los números

**Output** El output será un número - el mínimo número de monedas que uno tendría que pagar para realizar que el número sea 1.

2. (6 points) **Problema B****Parcial de hoy XD**

Tienen un set de problemas que consisten en  $n$  problemas. La dificultad del  $i$ -ésimo problema es  $a_i$ . Es garantizado que todas las dificultades son diferentes entre sí y que son dadas en orden creciente.

Napa tiene que crear un contest que consiste en algunos problemas de un problemset. En otras palabras *el contest que tiene que crear debe ser un subconjunto de problemas (no necesariamente consecutivos) del set original de problemas*. Solo hay una condición que debe cumplirse. Para cada problema que no es el más difícil del set escogido, debe haber en este set escogido otro problema que sea más difícil que este problema, pero no más que el doble de dificultad del problema. En otras palabras. sea  $a_{i_1}, a_{i_2}, \dots, a_{i_k}$  las dificultades en orden creciente del set escogido. Para cada  $j < p$  debe cumplirse que  $a_{i_j} \leq a_{i_{j+1}}$ . Esto significa que un escogido válido puede consistir de solo 1 problema

De todos los sets escogidos válidos que uno puede escoger del set original, Napa quiere escoger aquel con mayor cantidad de problemas. Tu task es encontrar este número de problemas.

**Input** La primera línea del input contiene un entero  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) - el número de problemas.

La segunda línea del input contiene  $n$  enteros  $a_1, a_2 \dots a_n$  ( $1 \leq a_i \leq 10^9$ ) - las dificultades de los problemas. **Es garantizado que las dificultades de los problemas son diferentes entre sí y son dados en orden creciente.**

**Output** Printear un único entero - el máximo número de problemas en el contest que puede escoger Napa de modo que se cumplan las condiciones del problema.

3. (6 points) **Problema C****Botones**

Vasya ha encontrado un artefacto extraño. En el panel frontal del artefacto se encuentran un boton rojo y un azul, y en la pantalla aparece un número entero positivo. Cada vez que presiona el boton rojo, el artefacto duplica el número en pantalla, mientras que cada vez que presiona el boton azul, el artefacto subtrae un elemento de la pantalla. Si en algún momento el número deja de ser positivo, el artefacto deja de funcionar. La pantalla tiene la capacidad de mostrar enteros positivos suficientemente grandes. Al inicio la pantalla muestra el número  $n$ .

**Input** La primera y única línea del input contiene dos enteros positivos  $n$  y  $m$  ( $1 \leq n, m \leq 10^4$ ), separados por un espacio.

**Output** Imprimir un único entero - la mínima cantidad de veces que uno necesita presionar el boton para obtener de llegar desde  $n$  a  $m$  apretando una serie de botones sin malograr el artefacto.

4. (8 points) **Problema D****Art Museum**

Una galería de arte tiene  $2N$  cuartos. La galería se puede representar como  $N$  filas de 2 cuartos lado a lado. Las puertas conectan cuartos adyacentes (que comparten un lado). Un trabajador informa que debe cerrar  $k$  de esos cuartos debido a recorte de personal. Los visitantes pueden ingresar por uno de los dos cuartos en la primera fila, y salir por uno de los cuartos de la última fila. Y caminar por 2 cuartos vecinos. Esto quiere decir que los cuartos que deben ser cerrados no deben impedir el tránsito desde la entrada a la salida. Además, el gerente ha calculado cuánto valor tiene cada uno de estos cuartos al público general, así que quiere cerrar  $k$  de los salones de modo que los cuartos restantes generen la mayor cantidad de valor para el público sin que se bloquee el tránsito entre la entrada y salida.

**Input** La primera y única línea del input contiene dos enteros positivos  $n$  y  $m$  ( $1 \leq n, m \leq 10^4$ ), separados por un espacio.

**Output** Imprimir un único entero - la mínima cantidad de veces que uno necesita presionar el botón para obtener de llegar desde  $n$  a  $m$  apretando una serie de botones sin malograr el artefacto.

5. (8 points) **Problema E****Menú**

Luego del examen de Competitiva los estudiantes han ido a un restaurante para ordenar su almuerzo y desahogar sus penas de lo difícil que fue el examen. Cada orden les toma casi tanto tiempo como resolver un problema, así que quizá puedas ayudarlos. Tu tarea es encontrar la lista de almuerzos que fueron pedidos en el restaurante, conociendo el costo total.

**Input** El input empieza con una línea conteniendo un entero  $1 \leq n \leq 100$ , el número de items en el menú. La siguiente línea contiene  $n$  enteros positivos que representan el costo de cada item del menú. Los items costarán a lo mucho 1000 soles.

Esto es seguido por una línea conteniendo  $1 \leq m \leq 1000$ , que representa la pedidos, y una línea con la  $m$  ordenes. Cada orden representa es un entero  $1 \leq s \leq 30000$ , el total de costos de cada orden en soles.

**Output** Para cada orden en el input, imprimir una línea de la siguiente manera: Si hay un único orden que puede sumar dicho total, imprimir una línea que menciona los números de los items que son necesarios para alcanzar dicho total, en orden creciente. El primer plato del menú tiene número 1, el segundo número 2, etc.

Si no existe un orden que da una suma pedida, imprimir en la línea *Impossible*. Si hay más de una forma de obtener dicha suma, imprimir *Ambiguous*.