



ADN

Un grupo de científicos está estudiando un tipo especial de cadenas de ADN. Una cadena de ADN se representa como una cadena de caracteres A, C, G o T. Los científicos solamente están interesados en cadenas de ADN que tienen un número **ímpar** de ocurrencias de A.

Los científicos quieren identificar y clasificar **parejas ordenadas** de cadenas de ADN (Z_1, Z_2) , donde Z_1 y Z_2 son cadenas de ADN de longitud n con un número ímpar de As. Nótese que el orden de las cadenas importa: si $Z_1 \neq Z_2$, la pareja (Z_1, Z_2) es diferente a la (Z_2, Z_1) . También es posible que $Z_1 = Z_2$.

Para identificar las parejas de ADN de longitud n , los científicos asignarán a cada pareja un identificador consistente en una **tupla ordenada** (X, Y_1, Y_2, Y_3) , donde X, Y_1, Y_2, Y_3 son cadenas binarias de longitud n . X es el llamado subidentificador primario y Y_1, Y_2 e Y_3 son los subidentificadores secundarios: por ello, los científicos imponen que X debe ser **estRICTAMENTE MAYOR** que el máximo de Y_1, Y_2 e Y_3 cuando las cadenas en binario se interpretan como números escritos en binario. Aquí también importa el orden de los subidentificadores.

Los científicos quieren diseñar una correspondencia entre identificadores y parejas de forma que a cada identificador le corresponda una pareja y a cada pareja le corresponda un identificador. Ayúdale implementando un programa que pase de identificadores a cadenas de ADN y de cadenas de ADN a identificadores.

Eres libre de establecer la correspondencia entre los identificadores de longitud n y las cadenas de ADN de longitud n como quieras, pero la correspondencia debe ser biunívoca: si tu programa imprime una determinada pareja de cadenas de ADN al ser dado un identificador como entrada, entonces debe imprimir ese mismo identificador al ser dado esa pareja de cadenas de ADN como entrada. Y al revés: si tu programa imprime un determinado identificador al ser dado una pareja de cadenas de ADN como entrada, entonces debe imprimir esa misma pareja al ser dado ese identificador como entrada.

Entrada y salida

La entrada comienza con una línea con un entero n y un bit b . Si $b = 0$, sigue una línea con cuatro cadenas binarias de longitud n : X, Y_1, Y_2 e Y_3 . Si $b = 1$, sigue una línea con dos cadenas de ADN de longitud n : Z_1 y Z_2 .

Debes imprimir una línea. Si $b = 0$, la línea debe contener dos cadenas de ADN de longitud n separadas por un espacio: Z_1 y Z_2 . Si $b = 1$, la línea debe contener cuatro cadenas binarias de longitud n separadas por espacios: X, Y_1, Y_2 , e Y_3 .

Este es un problema de ejecución múltiple. Por cada caso de prueba, tu programa se ejecutará dos veces seguidas de forma independiente.

Si en la primera ejecución se te pregunta por un identificador y das una pareja de cadenas de ADN, en la segunda ejecución se te preguntará por esa pareja de cadenas y deberás devolver el mismo identificador original (y al revés cuando se te pregunta primero por las cadenas de ADN).

Por razones técnicas, no puedes suponer que toda la entrada se encuentra disponible inmediatamente al inicio de la ejecución; por tanto, no puedes usar métodos que lean toda la entrada de una sola vez.

Ejemplo

Entrada:



2 0
11 01 10 10

Salida:

AC GA

Entrada:

2 1
AC GA

Salida:

11 01 10 10

Entrada:

1 1
A A

Salida:

1 0 0 0

En el primer caso como ejemplo se decide emparejar el identificador (11, 01, 10, 10) con la pareja (AC, GA).

En el segundo caso se pregunta la pareja (AC, GA), por lo que se debe devolver el mismo identificador que en el primer caso.

En el tercer caso, nótese que cuando $n = 1$ existe una única pareja válida y un único identificador válido.

Restricciones

- $1 \leq n \leq 2 \cdot 10^6$.
- Las tuplas (X, Y_1, Y_2, Y_3) , tanto las que se te proporcionan como las que debes imprimir, deben satisfacer $X > \max(Y_1, Y_2, Y_3)$ interpretando las cadenas binarias como números escritos en binario.
- Las tuplas (Z_1, Z_2) , tanto las que se te proporcionan como las que debes imprimir, deben satisfacer que Z_1 y Z_2 son cadenas de ADN con un número impar de ocurrencias de A.

Subtareas

1. (11 puntos) $n \leq 2$.
2. (30 puntos) $n \leq 15$.
3. (9 puntos) Los casos con $b = 0$ satisfacerán que $X = 2^n - 1$ en binario; los casos con $b = 1$ corresponderán a lo que imprima tu programa como respuesta a los casos con $b = 0$.
4. (10 puntos) Los casos con $b = 1$ satisfacerán que las cadenas no contienen ninguna T; los casos con $b = 0$ corresponderán a lo que imprima tu programa como respuesta a los casos con $b = 1$.
5. (40 puntos) Sin restricciones adicionales.