

**USA Computing Olympiad**  
**USACO 2024/2025 - Concurso Diciembre 2024, Nivel Bronce****1. Problema 1: Roundabout Rounding (Redondeo aproximado)**

¡Bessie la vaca está de vuelta en el colegio! Ha comenzado a hacer su tarea de matemáticas en la que se le pide redondear enteros positivos a potencias de 10.

Para redondear un entero positivo  $a$  al  $10^b$  más cercano, donde  $b$  es un entero positivo, Bessie primero ubica el  $b$ -ésimo dígito desde la derecha. Sea  $x$  este dígito.

Si  $x \geq 5$ , Bessie añade  $10^b$  a  $a$ .

Luego, Bessie desde el  $b$ -ésimo dígito desde la derecha, pone todos los dígitos de la derecha incluyendo el mismo en 0.

Por ejemplo, si Bessie quisiera redondear 456 al  $10^2$  (centena) más cercano, Bessie primero ubicaría el 2do dígito desde la derecha que es 5. Esto significa  $x = 5$ . Entonces, dado que  $x \geq 5$ , Bessie añade 100 a  $a$ . Finalmente, Bessie pone todos los dígitos en  $a$  a la derecha de  $e$  incluyendo el 2do dígito desde la derecha en 0, resultando en 500.

Sin embargo, si Bessie tuviera que redondear 446 al  $10^2$  más cercano, terminaría con 400.

Después de revisar la tarea de Bessie, Elsie piensa que ha inventado un nuevo tipo de redondeo: redondeo en cadena. Para redondear en cadena al  $10^b$  más cercano, Elsie primero redondeará al  $10^1$  más cercano, luego al  $10^2$  más cercano, y así sucesivamente hasta el  $10^b$  más cercano.

Bessie piensa que Elsie está equivocada, pero está muy ocupada con la tarea de matemáticas para confirmar sus sospechas. Ella te pide contar cuántos enteros  $x$  de al menos 2 y a lo sumo  $N$  ( $1 \leq N \leq 10^9$ ) existen de tal manera que redondear  $x$  al  $10^P$  más cercano es diferente de redondear en cadena al  $10^P$  más cercano, donde  $P$  es el entero más pequeño tal que  $10^P \geq x$ .

**FORMATO DE ENTRADA:**

Debes responder múltiples casos de prueba.

La primera línea de entrada contiene un solo entero  $T$  ( $1 \leq T \leq 10^5$ ) denotando el número de casos de prueba.  $T$  casos de prueba siguen.

La primera y única línea de entrada en cada caso de prueba contiene un solo entero  $N$ . Todos los  $N$  dentro del mismo archivo de entrada están garantizados para ser distintos.

**FORMATO DE SALIDA:**

Imprime  $T$  líneas, la  $i$ -ésima línea conteniendo la respuesta al  $i$ -ésimo caso de prueba. Cada línea debe ser un entero denotando cuántos enteros de al menos 2 y a lo sumo  $N$  existen que son diferentes cuando se usan los dos métodos de redondeo.

**EJEMPLO DE ENTRADA:**

```
4
1
100
4567
3366
```

**EJEMPLO DE SALIDA:**

```
0
5
183
60
```

Considera el segundo caso de prueba la muestra. 48 debería ser contado porque 48 redondeado en cadena al  $10^2$  más cercano es 100 ( $48 \rightarrow 50 \rightarrow 100$ ), pero 48 redondeado al  $10^2$  más cercano es 0.

En el tercer caso de prueba, dos enteros contados son 48 y 480. 48 redondea en cadena a 100 en lugar de a 0 y 480 redondea en cadena a 1000 en lugar de 0. Sin embargo, 67 no es contado ya que redondea en cadena a 100 que es 67 redondeado al  $10^2$  más cercano.

**PUNTUACIÓN:**

- Entradas 2-4:  $N \leq 10^3$
- Entradas 5-7:  $N \leq 10^6$
- Entradas 8-13: Sin restricciones adicionales.

Créditos del problema: Weiming Zhou

## 2. Problema 2: Farmer John's Cheese block(El bloque de queso del granjero John)

El granjero John tiene un bloque de queso en forma de cubo. Se encuentra en el plano de coordenadas tridimensional, extendiéndose desde  $(0, 0, 0)$  hasta  $(N, N, N)$  ( $2 \leq N \leq 1000$ ). El granjero John realizará una serie de  $Q$  ( $1 \leq Q \leq 2 \cdot 10^5$ ) operaciones de actualización en su bloque de queso.

Para cada operación de actualización, FJ cortará el bloque de queso de  $1 \times 1 \times 1$  que se extiende desde las coordenadas enteras  $(x, y, z)$  hasta  $(x + 1, y + 1, z + 1)$ , donde  $0 \leq x, y, z < N$ . Se garantiza que existirá un bloque de queso de  $1 \times 1 \times 1$  en la ubicación donde FJ corta. Dado que FJ está jugando a Minecraft, la gravedad no hace que partes del queso caigan si se corta el queso de abajo.

Después de cada actualización, se debe imprimir el número de configuraciones distintas en las que FJ puede colocar un ladrillo de  $1 \times 1 \times N$  en el bloque de queso de tal manera que ninguna parte del ladrillo se superponga con el queso restante. Cada vértice del ladrillo debe tener coordenadas enteras en el rango  $[0, N]$  para los tres ejes. FJ puede rotar el ladrillo como desee.

### **FORMATO DE ENTRADA (la entrada llega desde la terminal / stdin):**

La primera línea contiene  $N$  y  $Q$ .

Las siguientes  $Q$  líneas contienen  $x, y, z$ , las coordenadas a cortar.

### **FORMATO DE SALIDA (imprimir salida a la terminal / stdout):**

Después de cada operación de actualización, imprimir un entero, el número de configuraciones.

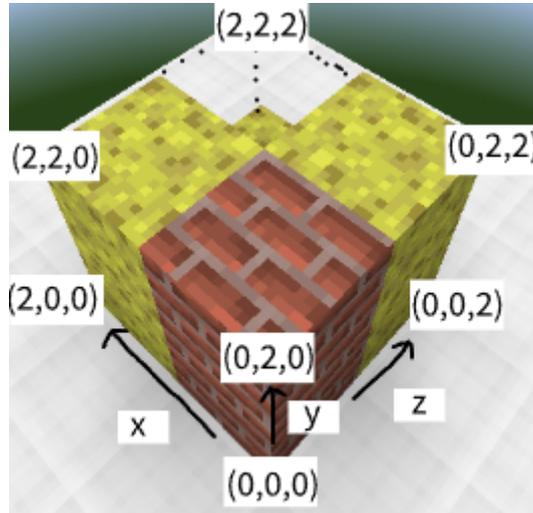
### **EJEMPLO DE ENTRADA:**

```
2 5
0 0 0
1 1 1
0 1 0
1 0 0
1 1 0
```

### **EJEMPLO DE SALIDA:**

```
0
0
1
2
5
```

Después de las primeras tres actualizaciones, el ladrillo de  $1 \times 2 \times 1$  que abarca  $[0, 1] \times [0, 2] \times [0, 1]$  no se superpone con el queso restante, por lo que contribuye a la respuesta.

**PUNTUACIÓN:**

- Entradas 2-4:  $N \leq 10^3$
- Entradas 5-7:  $N \leq 10^6$
- Entradas 8-13: Sin restricciones adicionales.

Créditos del problema: Weiming Zhou

### 3. Problema 3: It's moo in' time (Es hora de mugir)

El granjero John está tratando de describir su concurso favorito de USACO a Elsie, pero ella tiene problemas para entender por qué le gusta tanto. Él dice: "Mi parte favorita del concurso fue cuando Bessie dijo 'Es hora de Mugir' y mugió en todo el concurso."

Elsie todavía no entiende, así que el granjero John descarga el concurso como un archivo de texto e intenta explicar lo que quiere decir. El concurso se define como una cadena de letras minúsculas de longitud  $N$  ( $3 \leq N \leq 20,000$ ). Un "moo" se define generalmente como la subcadena  $c_i c_j c_j$  donde algún carácter  $c_i$  seguido directamente por 2 ocurrencias de algún carácter  $c_j$  donde  $c_i \neq c_j$ . Según el granjero John, Bessie muge mucho, así que si algún moo aparece al menos  $F$  ( $1 \leq F \leq N$ ) veces en el concurso, podría ser de Bessie.

Sin embargo, la descarga del granjero John podría haberse corrompido, y el archivo de texto podría tener hasta un carácter que difiere del archivo original. Imprime todos los posibles "moo" que Bessie podría haber hecho teniendo en cuenta el posible error, ordenados en orden alfabético.

#### **FORMATO DE ENTRADA (la entrada llega desde la terminal / stdin):**

La primera línea contiene  $N$  y  $F$ , representando la longitud de la cadena y el umbral de frecuencia para un "moo" de Bessie.

La segunda línea contiene una cadena de letras minúsculas de longitud  $N$ , representando el concurso.

#### **FORMATO DE SALIDA (imprimir salida a la terminal / stdout):**

Imprime el número de posibles "moo" que Bessie hace, seguido de una lista ordenada lexicográficamente de los "moo". Cada "moo" debe aparecer en una línea separada.

#### **EJEMPLO DE ENTRADA:**

```
10 2  
zzmoozzmoo
```

#### **EJEMPLO DE SALIDA:**

```
1  
moo
```

En este caso, ningún cambio de carácter afecta la respuesta. El único "moo" que Bessie hizo fue "moo".

**EJEMPLO DE ENTRADA:**

```
17 2
momoobaaaaaaqqqcqq
```

**EJEMPLO DE SALIDA:**

```
3
aaq
baa
cqq
```

En este caso, la `a` en la posición 8 (índice cero) podría haberse corrompido de una `b` que habría resultado en "baa"siendo un "moo"que Bessie hizo dos veces. Alternativamente, la `q` en la posición 11 podría haberse corrompido de una `c` que habría resultado en `cqc` siendo un posible "moo"que Bessie hizo. `aaq` puede hacerse intercambiando la `c` con una `a`.

**EJEMPLO DE ENTRADA:**

```
3 1
ooo
```

**EJEMPLO DE SALIDA:**

```
25
aoo
boo
coo
doo
eoo
foo
goo
hoo
ioo
joo
koo
loo
moo
noo
poo
qoo
roo
soo
too
uoo
voo
woo
xoo
yoo
zoo
```

**PUNTUACIÓN:**

- Entradas 4-8:  $N \leq 100$
- Entradas 9-13: Sin restricciones adicionales.

Créditos del problema: Suhas Nagar