



I CodeBoost

ETSI Sistemas Informáticos - 2024

<http://blogs.upm.es/codeboost>

Cuadernillo de problemas

23 de mayo de 2025



Proyecto de innovación educativa CodeBoost (IE25.6108) financiado por la Universidad Politécnica de Madrid

Índice

| | |
|---|---|
| A La Guerra Mortal entre el Ejército Verde y el Ejército Azul | 3 |
| B El Desafío de los Centinelas Primos | 5 |
| C El Código Omega | 7 |
| D Pirámide de Gosling | 9 |

Autores de los problemas:

- Raúl Lara Cabrera (Universidad Politécnica de Madrid)
- Fernando Ortega Requena (Universidad Politécnica de Madrid)
- Joaquín Gayoso Cabada (Universidad Politécnica de Madrid)
- Jorge Dueñas Lerín (Universidad Politécnica de Madrid)

Tiempo: 3 segundos



La Guerra Mortal entre el Ejército Verde y el Ejército Azul

En el Planeta de la Guerra se está librando una batalla épica entre los dos grandes ejércitos rivales: el **ejército verde** y el **ejército azul**. Cada soldado ha sido asignado a la batalla y luchará hasta que uno de los ejércitos desaparezca por completo: ¡ha comenzado la **Guerra Mortal**!



Cada soldado tiene una medida de poder, que describe su capacidad de combate. Cuando dos soldados se enfrentan uno contra uno, sobrevive el que tiene más poder y el otro muere. Sin embargo, el poder del soldado sobreviviente se reduce exactamente en la cantidad del poder del soldado que murió. Por ejemplo, si un soldado verde con poder 50 se enfrenta a uno azul con poder 40, el azul muere y el verde sobrevive con poder 10, ya que $50 - 40 = 10$. Si dos soldados tienen el mismo poder, ambos mueren.

En la guerra entre los dos ejércitos hay un número determinado de campos de batalla. Cada ejército asigna un soldado a cada campo, comenzando por los más poderosos. Por ejemplo, si un ejército tiene 5 soldados con poderes 50, 50, 40, 40, 30, y hay 3 campos de batalla, entonces un soldado con poder 50 va al campo 1, otro con poder 50 al campo 2, y uno con poder 40 al campo 3. El otro ejército hará lo mismo.

La Guerra Mortal se desarrolla por rondas. En cada ronda, los mejores soldados de cada ejército se enfrentan en los campos de batalla. Todas las batallas ocurren simultáneamente. Algunos soldados saldrán victoriosos (pero debilitados) y otros morirán. Los que sobrevivan volverán a sus respectivos ejércitos y se preparará una nueva ronda con los mejores soldados disponibles. Si en algún momento uno de los ejércitos no tiene suficientes soldados para cubrir todos los campos de batalla, entonces solo habrá lucha en los campos donde ambos ejércitos hayan enviado soldados.

La guerra termina cuando uno de los ejércitos se queda sin soldados o cuando ambos desaparecen al mismo tiempo. Por ejemplo, en una guerra con 2 campos de batalla, si el ejército verde tiene soldados con poder 20 y 10, y el azul tiene soldados con poder 10, 10, 15, entonces en la primera ronda habrá:

Campo 1: 20 vs 15 → gana el verde con poder 5

Campo 2: 10 vs 10 → ambos mueren

El ejército verde queda con un soldado de poder 5, y el azul con uno de poder 10 (que no peleó). En la siguiente ronda, solo hay un combate: 5 vs 10. Gana el azul, y el verde queda eliminado. ¡Victoria para el ejército azul!

Pero en la batalla real... ¿quién ganará?

Entrada

La primera línea contiene un número N , que indica la cantidad de casos de prueba. Cada caso de prueba comienza con una línea que contiene tres enteros separados por espacios: B , SG y SB , que representan respectivamente:

- el número de campos de batalla disponibles,
- la cantidad de soldados en el ejército verde,
- la cantidad de soldados en el ejército azul.

Luego siguen SG líneas, cada una con un entero P que indica el poder de un soldado del ejército verde, y luego SB líneas con los poderes de los soldados del ejército azul. Los soldados no están ordenados de ninguna manera específica.

Salida

Para cada caso de prueba, debes imprimir:

- verde y azul murieron si ambos ejércitos desaparecieron en la misma ronda
- verde gana si el ejército verde ganó, seguido de una línea por cada soldado sobreviviente (en orden descendente)
- azul gana si el ejército azul ganó, seguido de una línea por cada soldado sobreviviente (en orden descendente)

Debe haber **una línea en blanco** entre casos de prueba.

Entrada de ejemplo

```
3
5 1 1
10
10
2 2 3
20
10
10
10
15
3 5 5
50
40
30
40
50
50
30
30
20
60
```

Salida de ejemplo

```
verde y azul murieron

azul gana
5

verde gana
10
10
```

Límites

- $1 \leq N, P \leq 100$
- $1 \leq B, SG, SB \leq 100000$

Tiempo: 2 segundos



El Desafío de los Centinelas Primos

Iago, el célebre criptógrafo de la Orden de los Números Silentes, ha recibido un mensaje cifrado proveniente de una antigua red de sabios matemáticos. El contenido del mensaje está custodiado por una clave numérica: una serie de “centinelas primos”, números indivisibles que se alzan como guardianes del conocimiento secreto.

Para descifrar el mensaje, Iago debe conocer cuántos centinelas primos existen desde el principio del tiempo numérico (el 2) hasta un número N que recibe como parámetro. Su misión es implementar un algoritmo preciso y eficiente que calcule cuántos números primos hay hasta ese umbral N .

El tiempo apremia, y los secretos del mensaje podrían cambiar el equilibrio de las comunicaciones cifradas del mundo. ¿Estás listo para ayudar a Iago a desvelar cuántos centinelas primos custodian el umbral de la verdad?

Entrada

La entrada del programa indicará el N (incluido) hasta el que deben buscar los número primos.

Salida

Se deberá especificar cuántos números primos se han encontrado hasta el N indicado en la entrada.

Entrada de ejemplo

```
4
8
15
16
23
42
```

Salida de ejemplo

```
2
4
6
6
9
13
```

Límites

- $2 \leq N \leq 2000000$

Tiempo: 3 segundos

● C

El Código Omega

La P.I.A. (Politechnical Intelligent Agency) ha interceptado una transmisión sospechosa enviada por la I.A. Omega, una inteligencia artificial que ha cobrado conciencia propia y amenaza con conquistar el mundo empezando, como no, por Vallecas. Para desgracia de todos, el mundo vuelve a depender de los espías mas incompetentes del estado, usted, y de su capacidad de deducción y solución de problemas.

Según los archivos ultrasecretos del Súper, el mensaje contiene una **clave alfabética secreta** camuflada entre números y letras sin sentido. Esta clave está formada por secuencias de letras consecutivas (como **ababc** o **fedef**), y puede tolerar **un único salto de carácter** para continuar una secuencia.

Tu misión, agente de élite, es encontrar:

- La longitud de la **secuencia alfabética válida más larga** (ascendente y descendente en simultaneo), con un salto de carácter permitido.
- Cuántas veces aparece esa longitud máxima en el mensaje.

La clave será utilizada por el profesor para **encriptar la documentación necesaria** para detener a la I.A. Omega antes de que despliegue su plan definitivo de dominación global... dentro de pocos **minutos** (aunque conociendo al profesor, igual se autodestruye antes).



Entrada

Un conjunto de líneas con el mensaje interceptado: una cadena S de hasta 2 000 000 caracteres, que puede contener letras y números mezclados.

Salida

Parejas de números donde cada dígito esta separado por un espacio:

- La longitud máxima de una secuencia válida.
- Cuántas secuencias existen con esa longitud máxima.

Entrada de ejemplo

```
5
ab1cdef3gkz
ab
ab3ayb4a5zabcd
abycdref
yok
```

Salida de ejemplo

```
7 1 2 1 5 2 4 2 1 3
```

Notas

En `abycdref`, las secuencias más largas son `abcd`, con un salto permitido, y `cdef`. Dando lugar al resultado parcial 4 2

Límites

- Longitud de S : hasta 2 000 000 caracteres.
- Solo letras (mayúsculas o minúsculas) cuentan como parte de las secuencias.
- Las letras se comparan sin distinguir mayúsculas de minúsculas.
- Se permite un único salto en cada secuencia.

Tiempo: X segundos



Pirámide de Gosling

El 23 de mayo de 1995, *Sun Microsystems* lanzó Java al mundo. Hoy, 29 años más tarde, celebramos el primer **CodeBoost** en esa misma fecha. Para conmemorar este hito, hemos preparado un homenaje especial: una **pirámide de tazas de café**.

Según cuenta la leyenda, **James Gosling**, el creador de Java, solo podía pensar con claridad si sus tazas estaban dispuestas en una pirámide perfecta y cromáticamente equilibrada. Cada taza tenía un color (representado por un número entero), y Gosling era muy estricto con dos reglas:

- **Equilibrio perfecto:** Cada color debe aparecer exactamente $\frac{m}{k}$ veces, donde $m = \frac{n(n+1)}{2}$ es el total de tazas.
- **Radio de exclusión:** Ningún color puede repetirse en un radio de d posiciones laterales. Si una taza está en posición i , todas las posiciones desde $\max(0, i - d)$ hasta $\min(m - 1, i + d)$ (excluyendo i) deben tener colores diferentes. NOTA: Como este radio solo se aplica a las posiciones laterales, podemos tratar la pirámide como una lista lineal de m posiciones.

La pirámide se representa como lista lineal ordenada de arriba abajo y de izquierda a derecha. Por ejemplo, para $n = 3$:

```
[0]
[1] [2]
[3] [4] [5]
```

Como el radio de exclusión solo se aplica a las posiciones laterales se puede representar como una lista lineal de la siguiente manera:

```
[0] [1] [2] [3] [4] [5]
```

Ejemplo de radio ($d = 2$ en posición 2): Las posiciones 0,1,3,4 no pueden compartir color con la posición 2.

Entrada

Cada caso de prueba contiene tres enteros:

n k d

- n : Niveles de la pirámide ($1 \leq n \leq 7$)
- k : Colores disponibles ($1 \leq k \leq 3$)
- d : Radio de exclusión ($1 \leq d \leq 3$)

Salida

Para cada caso escribir el número de soluciones posibles.

Entrada de ejemplo

```
1 1 1
2 1 2
3 3 2
4 2 2
5 3 1
6 3 2
```

Salida de ejemplo

```
1
0
6
0
7188
6
```

Límites

- $1 \leq n \leq 7$
- $1 \leq k \leq 3$
- $1 \leq d \leq 3$