



II CodeBoost

ETSI Sistemas Informáticos - 2025

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

Cuadernillo de problemas

19 de noviembre de 2025



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

Índice

A El club de los programadores	3
B Diseño de la bandera de CodeBoost	5
C Silo-gística Vertical	7
D La Batalla de Gayos Palíndromos	9

Autores de los problemas:

- Guillermo Iglesias Hernández (Universidad Politécnica de Madrid)
- Raúl Lara Cabrera (Universidad Politécnica de Madrid)
- Joaquin Gayoso Cabada (Universidad Politécnica de Madrid)
- Fernando Ortega Requena (Universidad Politécnica de Madrid)

Tiempo: 1 segundo

● A

El club de los programadores

En una pequeña ciudad costera, escondido en el segundo piso de una biblioteca vieja, se reunía cada viernes por la tarde el Club de Programadores CodeBoost. No era un club cualquiera: allí, los teclados sonaban como tambores de guerra y el olor a café se mezclaba con la emoción de las ideas nuevas.

Pero lo más curioso de aquel grupo no era lo que programaban, sino cómo se saludaban. Tenían una tradición tan extraña como divertida, nacida, según contaban, de un error en un viejo programa que un día imprimió algo inesperado.

Cuando los miembros iban llegando, uno a uno, el resto los saludaba siguiendo un código secreto que solo los iniciados entendían:

Cada persona tenía un número, según el orden en que se unía a la reunión. Cuando alguien decía el número, los demás respondían. Si el número era **múltiplo de 3**, todos gritaban con entusiasmo “*Code!*”, levantando las tazas de café. Si era **múltiplo de 5**, aplaudían al ritmo y respondían “*Boost!*”, como si estuvieran recargando energía. Pero si el número era **múltiplo de ambos**, la sala entera se levantaba y coreaba “*CodeBoost!*”, el saludo más honorable de todos, reservado solo para los verdaderos programadores legendarios.

Decían que aquel ritual había nacido una noche en que los primeros miembros del club, agotados y medio dormidos, decidieron convertir un simple error de sintaxis en una tradición eterna.

Desde entonces, ningún viernes pasaba sin que los muros de la biblioteca temblaran con sus saludos:

“*Uno.*”

“*Dos.*”

“*¡Code!*”

“*Cuatro.*”

“*¡Boost!*”

Y así, una y otra vez, hasta que la risa y el código se mezclaban en una sola melodía.



Entrada

La entrada del programa estará compuesta por varias líneas, cada una con un número entero que representa la cantidad de miembros N del Club de Programadores CodeBoost.

Salida

Como salida, el programa deberá mostrar todos los saludos correspondientes al número de miembros indicado en la entrada.

Entrada de ejemplo

```
15
4
2
```

Salida de ejemplo

```
1
2
Code
4
Boost
Code
7
8
Code
Boost
11
Code
13
14
CodeBoost
1
2
Code
4
1
2
```

Límites

- $1 \leq N \leq 100$

Tiempo: 1 segundos



Diseño de la bandera de CodeBoost

En dicha ciudad costera, donde el mar y el código se entrelazaban, los programadores de CodeBoost se enfrentaban a un curioso dilema: necesitaban una bandera que los representara.

Ninguno tenía talento para el diseño, pero eso no los detuvo. Decidieron transformar su debilidad en oportunidad y crear un sistema que generara banderas mediante código.

El estilo sería sencillo, casi matemático: una forma cuadrada en el centro, sólida y constante, y un triángulo de tamaño variable al fondo, símbolo del impulso creativo que los guiaba. Así, entre algoritmos y formas geométricas, los programadores hallaron su identidad.

Entrada

La entrada del programa estará compuesta por varias líneas, cada una con dos números (n_1 y n_2) y dos caracteres que representan los diseños de banderas del Club de Programadores CodeBoost. Cada número representa el tamaño de la geometría y cada carácter el símbolo que rellena dicha forma. El orden serán primero las características del cuadrado y, segundo, las del triángulo.

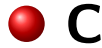
Salida

Como salida, el programa deberá mostrar las banderas resultado de las propuestas indicadas en la entrada.

Entrada de ejemplo

```
10 . 10 *
15 X 7 |
5 o 0 A
7 ? 4 $
```


Tiempo: 0.5 segundos



Silo-gística Vertical

Eres un *porter* (repartidor) en el Silo. Tu trabajo es llevar paquetes por la gran escalera espiral que conecta los niveles. El Silo es una estructura puramente vertical, donde los niveles están apilados en orden numérico. Por lo tanto, el número de pisos que debes caminar entre un nivel A y un nivel B es simplemente $||A - B||$. Hoy empiezas tu jornada en el comedor del **nivel** P . Tienes N paquetes que entregar. Afortunadamente, caben todos en tu carrito. Esto significa que puedes llevarlos todos a la vez.



Tu plan es visitar todos los N niveles de entrega en algún orden. No necesitas volver al nivel P entre entregas.

Empezarás en P , viajarás al primer nivel de entrega, luego al segundo, y así sucesivamente, hasta que entregues el último paquete en el último nivel de visita. Tu objetivo es ser eficiente. Quieres encontrar el orden de entrega que **minimice el número total de niveles** que tienes que caminar.

Por ejemplo, si empiezas en el nivel 50 y tienes que entregar en los niveles 10, 60 y 20, una posible ruta es: $50 \rightarrow 10 \rightarrow 20 \rightarrow 60$. El coste total de esta ruta es: $\text{abs}(50 - 10) + \text{abs}(10 - 20) + \text{abs}(20 - 60) = 40 + 10 + 40 = 90$ niveles.

Dados tu nivel inicial P y una lista de N niveles de entrega, calcula el número total mínimo de niveles que debes recorrer para visitar todos los destinos.

Entrada

La entrada del problema, que se leerá de la entrada estándar, está formada por dos líneas:

- La primera línea contendrá dos enteros separados por un espacio: P (tu nivel inicial) y N (el número de paquetes).
- La segunda línea contendrá N enteros separados por espacios: los niveles de destino L_1, L_2, \dots, L_N .

Salida

Como salida, el programa deberá escribir en la salida estándar un único entero: el número total mínimo de niveles recorridos.

Entrada de ejemplo

```
100 3
10 20 30
```

Salida de ejemplo

```
90
```

Límites

- $1 \leq P \leq 1000$
- $1 \leq N \leq 100$
- Todos los niveles L_i están entre 1 y 1000

Tiempo: 0,125 segundos



La Batalla de Gayos Palíndromos

Hay una tarima, dos micros y una multitud expectante. Esta no es una batalla cualquiera: es la **Batalla de Gayos Palíndromos**, el torneo más extraño del circuito de raperos de la costa digital. Aquí no se improvisan rimas ni se grita sin razón: se forjan reflejos. Cada palabra lanzada al aire busca su eco perfecto, porque solo aquellas que se leen igual al derecho y al revés son dignas de permanecer.

Nada de bases, ni coros, ni autotune; solo letras puras, desafiando a la simetría. Cada ronda es una prueba de ingenio, una lucha contra el espejo. La fuerza de cada verso se mide por el *palíndromo contiguo más largo* escondido dentro de la palabra elegida: cuanto más largo y perfecto el reflejo, más respeto se gana entre los espectadores y más vibra el público con cada eco del micrófono.



Las reglas son simples, implacables y legendarias:

- Si una palabra tiene un palíndromo de longitud mayor o igual que 4 y la otra no, gana quien lo tiene.
- Si ambas lo tienen, gana quien tenga el palíndromo más largo.
- Si empatan en longitud o ninguna alcanza 4, la ronda termina en tablas.

Al final de la batalla, solo cuenta quién ganó más rondas. Si ambos ganan las mismas, el veredicto es un empate total.

Entrada

La entrada comienza con un entero T que indica el número de batallas (casos de prueba).

Para cada batalla:

- Una línea con un entero R — el número de rondas.
- A continuación, R líneas; cada una contiene dos palabras A y B (solo letras, sin espacios), separadas por un espacio.

Se define la **longitud de combate** de una palabra como la longitud del **palíndromo contiguo más largo** que contiene. Una ronda otorga punto al primer rapero si su longitud de combate es ≥ 4 y estrictamente mayor que la del segundo; análogamente para el segundo. En caso contrario (igualdad o ambas < 4), la ronda no otorga puntos.

Salida

Por cada batalla, imprime una línea con un único entero:

- 1 si el primer rapero gana más rondas,
- 2 si el segundo rapero gana más rondas,
- 0 si empatan.

Entrada de ejemplo

```
1
3
reconocemos parladoradarín
microrotorotorista ensalasona
menemismo arenero
```

Salida de ejemplo

```
1
```

Explicación (informal): reconocemos aporta econoce (7) frente a parladoradarín con radar (5), gana el primero. En la segunda ronda, microrotorotorista contiene el palíndromo rotorotor (9) frente a ensalasona con salas (5), vuelve a ganar el primero. En la tercera, menemismo con menem, (5) supera a arenero con ene, (0 al ser palíndromo menor que 4). Marcador final: 3-0 para el primero; **salida 1**.

Límites

- $1 \leq T \leq 50$
- $1 \leq R \leq 100,000$
- Longitud de cada palabra ≥ 1 .
- Suma total de longitudes de todas las palabras en toda la entrada $\leq 300,000$.

Notas

- Se considera **palíndromo contiguo** a cualquier subcadena que se lea igual al derecho y al revés.