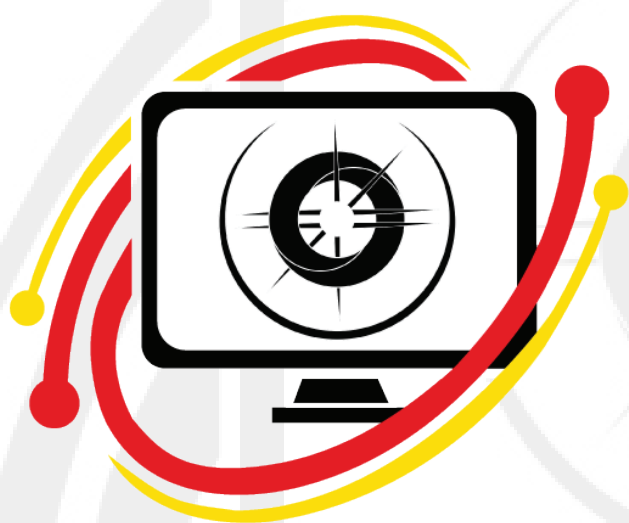


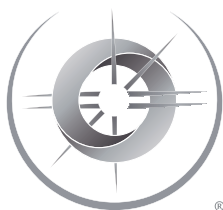
**I Olimpiada aragonesa de informática (2021-2022)**



# **Olimpiada Aragonesa de Informática**

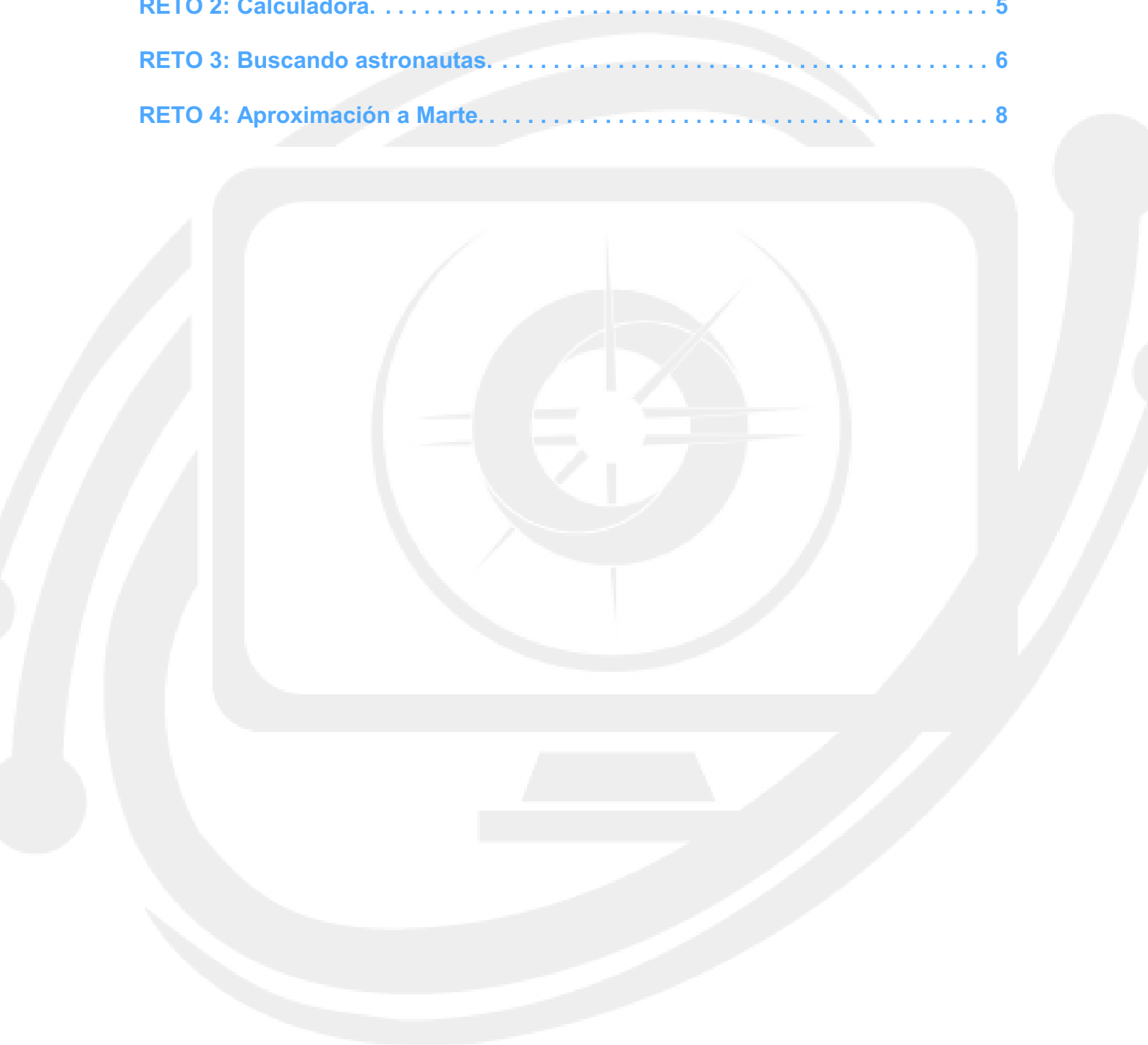
**ALGORITMIA**  
**Bachiller**

**Organiza**



Colegio Profesional  
de Ingenieros Técnicos  
en Informática de Aragón

INTRODUCCIÓN.....	3
RETO 1: el código.....	4
RETO 2: Calculadora. ....	5
RETO 3: Buscando astronautas. ....	6
RETO 4: Aproximación a Marte.....	8



# INTRODUCCIÓN

La Agencia Espacial Española planea realizar una misión al planeta Marte con el objetivo de ser los primeros humanos en llegar allí. Es por ello que se desea reclutar a las mentes más agudas para que les ayude a resolver una serie de problemas que les ha surgido en distintas fases del proyecto. Si estáis leyendo este documento estáis de enhorabuena, porque habéis sido preseleccionados para esta misión.

A lo largo de esta prueba os iremos enunciando una serie de problemas que deberéis resolver usando uno de los siguientes lenguajes de programación:

- C
- C++
- Java
- Python 3

Además de un lenguaje de programación, necesitareis vuestro ingenio del que estamos seguros que tenéis de sobra.

Antes de comenzar permitidnos unos consejos:

- 1. ESTO NO ES UN EXAMEN**, vuestras notas no se van a ver afectadas por lo que hagáis aquí.
- 2. NO HAY UN ORDEN PARA RESOLVER LOS RETOS**, si no encontráis la forma de solucionar un reto en **ESE** momento, no os frustréis, no os obcequéis, pasad al siguiente y cuando hayáis resuelto los demás, volved a ese.
- 3. LEED ATENTAMENTE LOS ENUNCIADOS DEL RETO**, para resolver el problema tenéis que saber que es lo que se pide, como os van a pasar los datos y en que formato se pide la salida.
- 4. ¿CÓMO SE GANA?**, resolviendo el mayor número de retos posibles. Y entre los que resuelvan un problema, el que haga el programa **MÁS** óptimo.
- 5.** No es necesario que siempre uséis el mismo lenguaje de programación, podéis resolver cada reto con el lenguaje que deseéis.

## RETO 1: el código.

Se os va a facilitar una información súper secreta procedente de la Agencia Espacial Española (AEE), se trata de unos códigos que facilitan la ignición del cohete. Para ello se deberá extrayendo la letra correcta de cada palabra obtendrás el código de ignición del cohete.

Para ello deberéis escribir un programa un que elimine el enésimo carácter de una cadena de entrada.

**Entrada:** La primera línea de entrada contiene un único entero  $N$  ( $1 \leq N \leq 1000$ ) que es el número de conjuntos de datos que siguen.

Cada conjunto de datos consta de una única línea de entrada que contiene  $M$ , un espacio y una cadena formada solo por letras mayúsculas y espacios.  $M$  será menor o igual que la longitud de la cadena. Se garantiza que la longitud de la cadena sea menor o igual a 80.

**Salida:** El programa debe devolver la cadena con el caracter eliminado.

Ejemplo:

ENTRADA	SALIDA ESPERADA
4 Estación	Estcion
3 Satélite	Saelite
5 Gravedad	Gravead
1 Internacional	Nternacional
2 Casco	Csco

**El programa no debería tardar más de un segundo en resolver el problema.**

## RETO 2: Calculadora.

Los astronautas de la misión a Marte se han dado cuenta que se han olvidado todas las calculadoras en la tierra y ya es demasiado tarde como para volver atrás, porque ya han despegado y están junto a la luna.

En sí mismo no es un problema muy grave pero los numerosos cálculos que tienen que hacer a lo largo del día les lleva mucho tiempo y necesitan una solución que les facilite el día a día.

Es por eso que la Agencia Espacial Española os encarga la misión de programar una calculadora que siga los siguientes parámetros:

- La fórmula del cálculo se introducirá por la entrada estándar.
- Se aceptarán las 4 operaciones básicas  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $( )$ ,  $^$  y la variable  $X$ .
- Hay que tener en cuenta el orden de las operaciones, recordemos que los paréntesis agrupan operaciones, que la potencia ( $^$ ), la multiplicación ( $*$ ) y la división ( $/$ ) tienen preferencia frente a la suma ( $+$ ) y a la resta ( $-$ ).
- No se introducirán valores cuyo cálculo de como resultado una división por 0.
- El resultado es un número real que se imprimirá por la salida estándar.
- El valor de  $X$ , generar un rango de valores de  $X_{min}$  a  $X_{max}$

Ejemplo:

**ENTRADA**

$X^2 * 5 / 6 * 8 + 3$  5 6

**SALIDA ESPERADA**

169,666666666666  
243

## RETO 3: Buscando astronautas.

A la hora de escoger quienes son los mejores astronautas para realizar la misión la agencia espacial Española se han encontrado con demasiados candidatos y tienen que poder seleccionar a los mejores de una forma justa.

Para ello, se va a someter a los astronautas a ciertas pruebas en un juego. Para ello se dispone de una hilera de  $N$  cubetas para almacenar los premios obtenidos por  $N$  candidatos, numerados de 1 a  $N$ , en un cierto juego. Antes de empezar el juego, cada cubeta contiene un entero 0.

La información para incrementar los valores de las cubetas, cada vez que se adjudica un nuevo premio múltiple durante el juego, es la siguiente:

- Se define un premio múltiple fijando un subrango de cubetas de la hilera mediante dos índices  $\{i, j\}$ , con  $i < j$ . Por ejemplo,  $i = 3, j = 6$ .
- Se reparte el premio múltiple incrementando en 1 el valor de la cubeta  $i$ , en 2 el de la cubeta  $i + 1$ , en 3 el de la cubeta  $i + 2$ , etcétera, hasta llegar a incrementar el valor de la cubeta  $j$ .

Por ejemplo, si  $N = 8$  se parte de una hilera de 8 cubetas que inicialmente contiene ceros:  $\langle 0, 0, 0, 0, 0, 0, 0, 0 \rangle$ . Si se adjudica el premio múltiple  $\{4, 7\}$  se incrementan las cubetas 4, 5, 6 y 7, quedando la siguiente hilera:  $\langle 0, 0, 0, 1, 2, 3, 4, 0 \rangle$ . Si a continuación se adjudica otro premio múltiple,  $\{5, 8\}$ , se incrementan las cubetas 5, 6, 7 y 8, y la hilera queda como  $\langle 0, 0, 0, 1, 3, 5, 7, 4 \rangle$ .

Una vez que se han adjudicado una secuencia de premios múltiples a la hilera de cubetas, se necesita responder preguntas como la siguiente:

Dado un subrango de cubetas de la hilera definido mediante dos índices  $[m, n]$  con  $m < n$ , por ejemplo  $m = 5$  y  $n = 7$ , se debe obtener como respuesta la suma de los valores de todas las cubetas comprendidas en el subrango que va desde 5 hasta 7 módulo 10000 (es decir, el resto que se obtiene al dividir –división entera– esa suma de valores por 10000).

**Problema:** Dado el valor de N y una secuencia de P premios múltiples definidos por parejas {i, j}, escribir un programa que responda a Q preguntas.

**Entrada:** De la primera línea de la entrada estándar deben leerse los valores de N, P y Q, que son números enteros positivos (tales que:  $1 \leq N \leq 1000000000$ ,  $1 \leq P \leq 1000$ ,  $1 \leq Q \leq 1000$ ), separados por un espacio. N es el número de cubetas de la hilera inicializadas con 0, P es el número de premios múltiples adjudicados a las cubetas y Q es el número de preguntas a realizar sobre la hilera de cubetas después de haber adjudicado todos los premios. Las siguientes P líneas contienen dos números enteros positivos (separados por espacio) cada una, representando los subrangos de cada premio múltiple. Y las siguientes Q líneas contienen dos números enteros positivos (separados por espacio) cada una, representando los subrangos de cada pregunta.

**Salida:** Se debe escribir en la salida estándar Q líneas (tantas como preguntas), y cada una de ellas debe contener un número entero igual a la respuesta a la correspondiente pregunta (no olvides que las respuestas a las preguntas son módulo 10000).

**Ejemplo:**

**ENTRADA**

15 3 3
9 11
6 9
2 6
1 11
6 9
2 2

**SALIDA ESPERADA**

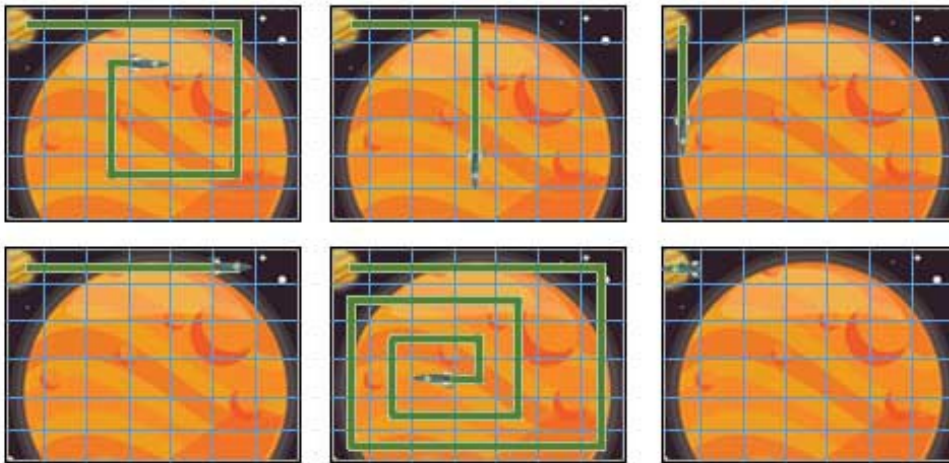
31
16
1

**Información que puede ser útil:** El sumatorio de números desde 1 hasta n es  $n * (n + 1) / 2$ .

## RETO 4: Aproximación a Marte.

Después de meses, por fin la misión ha llegado al planeta rojo, pero en el momento de iniciar la aproximación al planeta, los astronautas se han dado cuenta que un micro asteroide a inutilizado el propulsor derecho, por lo que la nave solo puede avanzar hacia adelante y girar 90 grados a la derecha (recuerda que el propulsor derecho hacia girar hacia la izquierda y viceversa). Para calcular las distintas rutas de aproximación, se dispone de una retícula rectangular con una serie de casillas organizadas en N filas y M columnas. La nave siempre empieza en la casilla superior izquierda y se mueve dando un paso hacia adelante o girando 90 grados a la derecha y dando un paso. El maniobra de aproximación puede finalizar su recorrido en cualquier momento siendo un simple punto o una simple línea recta recorridos posibles. Una cosa más, el propulsor de la nave produce una estela radioactiva que hace que sea peligroso atravesarla para los astronautas, por lo que no se puede volver a pasar nunca por una casilla por la que ya ha pasado.

Seis ejemplos de recorridos distintos trazados:



Necesitamos que escribas un programa que dados N y M, averigüe cuántos recorridos distintos puede realizar la nave en esa cuadrícula, escribiendo en la salida estándar ese número módulo mil millones (es decir, el resto que se obtiene al dividir –división entera– el número por 1000000000).



**Entrada:** Se debe leer desde la entrada estándar los valores de dos enteros N y M (tales que  $1 \leq N \leq 1000$ ,  $1 \leq M \leq 1000$ ) separados por un espacio.

**Salida:** Se debe escribir en la salida estándar el número total (módulo mil millones) de recorridos posibles que puede realizar por la nave.

Ejemplos:

ENTRADA	SALIDA ESPERADA
2 3	9
ENTRADA	SALIDA ESPERADA
5 5	251



# I OLIMPIADA

## ARAGONESA DE INFORMÁTICA

- COLEGIO PROFESIONAL DE INGENIEROS TÉCNICOS  
EN INFORMÁTICA DE ARAGÓN -

INSCRIPCIONES EN:  
[www.olimpiadaaragonesainformatica.es](http://www.olimpiadaaragonesainformatica.es)



Organiza:



Colegio Profesional  
de Ingenieros Técnicos  
en Informática de Aragón

Patrocina:



Colabora:



Escuela de  
Ingeniería y Arquitectura  
**Universidad Zaragoza**

universidad  
**SANJORGE**  
GRUPO SANVALERO



ESCUELA DE  
ARQUITECTURA  
Y TECNOLOGÍA



Escuela Universitaria  
Politécnica - Teruel  
**Universidad Zaragoza**



Escuela Universitaria  
Politécnica - La Alfranca  
Centro adscrito  
**Universidad Zaragoza**



**GOBIERNO  
DE ARAGON**