

Problema 1: El problema del parque (sección 1)

Imagina que estás trabajando en un proyecto de conservación de un gran parque nacional que está dividido en secciones para realizar un manejo más eficaz de los recursos naturales. El parque tiene una extensión total que, por simplicidad, consideraremos aquí como una medida cuantitativa discreta, por ejemplo, 1000 hectáreas. La administración del parque ha decidido dividir esta área en zonas de conservación de igual tamaño para poder asignar recursos y planificar actividades de conservación de manera más organizada. Sin embargo, debido a restricciones en el sistema de información geográfica (GIS) utilizado para la planificación, las operaciones de división y módulo están temporalmente deshabilitadas debido a una actualización del sistema. El desafío es dividir el área total del parque en zonas de igual tamaño sin utilizar los operadores de división (/) ni módulo (MOD), para determinar cuántas zonas de conservación se pueden formar y cuánta área quedaría sin asignar si no es posible una división exacta.

Este proceso es crucial para la asignación eficiente de los recursos naturales, la planificación de actividades de conservación, y para asegurar que todas las áreas del parque reciban la atención necesaria.

Problema resumido: La administración del parque ha decidido que cada zona de conservación debe abarcar 50 hectáreas. Con un total de 1000 hectáreas, necesitan un método para calcular cuántas zonas de 50 hectáreas se pueden formar y cuántas hectáreas sobrarían, todo esto sin utilizar los operadores de división o módulo.

Casos de prueba

Entrada (Hectáreas)	Salidas (Zonas completas Área remanente)	Estado
1000	20 0	<input type="checkbox"/>
1025	20 25	<input type="checkbox"/>
450	9 0	<input type="checkbox"/>
365	7 15	<input type="checkbox"/>
-450	Error! debe ser positivo	<input type="checkbox"/>

La salida debe contener en la primera línea el total de áreas completadas y en la segunda la cantidad de áreas sobrantes. En caso de una entrada negativa, envía un mensaje de error.

Problema 2: Potenciar el Kamehameha

Goku está practicando su técnica Kamehameha y necesita que le ayudes a gestionar sus niveles de energía para asegurarse de que puede lanzar la ráfaga más potente posible. Su energía está almacenada en un arreglo de células energéticas que representa el consolidado de energía diaria, cada una de las cuales contiene una cantidad determinada de energía (positiva, negativa o cero). Cada célula representa una unidad específica de energía consolidada que puede aprovechar a diario. Para prepararse para el Kamehameha, requiere saber cuál es el día recomendado para hacer su despliegue.

Tu tarea es desarrollar un programa que lea el arreglo de enteros, donde cada entero representa la energía diaria acumulada. El programa debe determinar el día que le conviene mostrar su técnica considerando el de máxima energía.

Restricciones

- El vector puede contener tanto números positivos como negativos. Los valores negativos representan áreas donde la energía de Goku se agota.
- Los días inician el domingo (0: domingo, 1: lunes, 2: martes ...6: sábado)
- No use tildes en las salidas, los días deben ser desplegarse en minúscula

Entrada

Un arreglo de 7 enteros representando la energía de cada celda por cada día.

Salida

Un único entero que representa la energía máxima que Goku dispone, en la línea siguiente el día que corresponde en palabras seguido del sufijo " de kamehameha"

Casos de prueba

Entrada	Salida	Estado
5 -3 7 2 -8 10 -1	10 viernes de kamehameha	<input type="checkbox"/>
15 -3 7 2 -8 10 -1	15 domingo de kamehameha	<input type="checkbox"/>
1 1 1	1 cualquier día de kamehameha	<input type="checkbox"/>

1		
1		
1		
1		
-1	nunca de kamehameha	<input type="checkbox"/>
-1		
-1		
-4		
-3		
-5		
-1		

Problema 3: El problema de distribución de libros escolares

Imagina que estás ayudando a una escuela a distribuir libros entre las aulas para un nuevo semestre. La escuela tiene una cantidad total fija de libros disponibles, digamos 720 libros. Lo ideal sería que cada aula recibiera la misma cantidad de libros para garantizar una distribución justa entre los estudiantes. Sin embargo, debido a una actualización reciente del software, el sistema de gestión de inventario ha desactivado temporalmente las operaciones de división (/) y módulo (MOD). La tarea es determinar cuántas aulas pueden recibir la misma cantidad de libros, dado que lo ideal sería que cada aula recibiera 30 libros. Además, si no es posible una división perfecta, calcula cuántos libros quedarían sin asignar. Esta información ayudará a la escuela a planificar cualquier compra adicional o espacio de almacenamiento necesario para los libros no utilizados.

Resumen del problema: La escuela necesita distribuir libros de manera que cada aula reciba exactamente 30 libros. Con un total de 720 libros disponibles, cree un método para determinar cuántas aulas completas se pueden abastecer y cuántos libros quedarían sin distribuir si no es posible una división exacta, todo ello sin utilizar operadores de división o módulo.

Casos de prueba

Entrada (Total libros)	Salidas (cursos completos libros sin distribuir)
720	24 0
745	24 25
150	5 0
98	3 8
-150	Error! debe ser positivo

La salida debe contener en la primera línea el total de cursos con asignación completa y en la segunda la cantidad de libros sobrantes. En caso de una entrada negativa, envía un mensaje de error.

Restricciones de soporte: Si es necesario, solicite ayuda a el/la ayudante o profesor/a. Tiene solo 3 oportunidades para cada problema.

Problema 4: Carga de Choque Trueno de Pikachu

Pikachu se está preparando para una gran batalla, y necesita que le ayudes a gestionar sus niveles de electricidad para asegurarse de que puede lanzar el Choque *Trueno* más fuerte posible. Pikachu carga su energía eléctrica en una batería por cada día de la semana, distribuidas en un arreglo de 7 posiciones. Para estar bien preparado, Pikachu requiere saber cuál es el día de máxima electricidad que puede encontrar en las baterías almacenadas para *lanzar su Golpe Trueno*.

Tu tarea es escribir un programa que:

1. Lea un arreglo de enteros donde cada entero representa la energía eléctrica cargada en una batería, una por cada día de la semana .
2. Identifique la posición y valor de la energía máxima de cualquier batería en el arreglo.
3. Muestre la energía máxima y el correspondiente día de la semana en palabras agregando el sufijo “ es un buen día para un Golpe Trueno”.

Restricciones

1. El arreglo puede contener números positivos y negativos. Los números negativos representan días de pérdidas de energía respecto de la semana anterior.
2. Los días empiezan el lunes (0 = lunes, 1 = martes, ... 6 = domingo).

Entrada

- Un array de 7 enteros que representan la energía de cada batería diariamente.

Salida

- Un único entero para la energía máxima que Pikachu puede recoger de una batería, y el día correspondiente en palabras con el sufijo “ de Golpe Trueno”.

Casos de prueba

Entrada	Salida
5 -3 7 2 -8 10 0	10 sábado es un buen día para un Golpe Trueno
15 -3 7 2 -8 10 -1	15 lunes es un buen día para un Golpe Trueno
1 1 1 1 1 1 1	1 cualquier día es un buen día para un Golpe Trueno

-2 -4 -5 -1 -1 -1 -10	nunca es un buen día para un Golpe Trueno
---	---

Restricciones de soporte: Si es necesario, solicite ayuda a el/la ayudante o profesor/a. Tiene solo 3 oportunidades para cada problema.