

Normas de la entrega de prácticas

- Las prácticas deben compilar, si no se corregirán.
- Los ejercicios marcados como “obligatorios” son ejercicios que tienen que funcionar perfectamente
- Es importante mantener un mínimo de pulcritud a la hora de escribir y elegir nombre de variables y funciones.
- Si se opta por el primer modelo de práctica, ES IMPRESCINDIBLE que los ejercicios se encuentren ordenados según el enunciado. Además, Hay que copiar el enunciado antes de cada función, con comentarios.

Ejercicios de funciones

Nota máxima 10

Escribir una función que calcule el área de un círculo y otra que calcule el volumen de un cilindro usando la primera función.

(obligatorio) Realiza una función que se le pase dos jugadas de “Piedra, papel, tijera, lagarto o Spock” y devuelva quién gana de los dos. Si gana el primer jugador, devuelve un -1, si gana el segundo devuelve un 1, si quedan empates, un 0. Se tiene que usar un enum.

Desarrolla una función que devuelva el resultado de una ecuación de primer grado.

(obligatorio) Desarrolla una función que devuelva el resultado de una ecuación de segundo grado.

Los tres lados a, b y c de un triángulo deben satisfacer la desigualdad triangular: cada uno de los lados no puede ser más largo que la suma de los otros dos. Escribe un programa que reciba como entrada los tres lados de un triángulo (son reales), e indique: si acaso el triángulo es inválido; y si no lo es, qué tipo de triángulo es (un enum).

Escribir una función que calcule el máximo común divisor de dos números.

Escribir una función que calcule el mínimo común múltiplo de dos números.

Escriba un programa que devuelva un string con los números naturales menores o iguales que un número n determinado y que no sean múltiplos ni de 3 ni de 7.

(obligatorio) Escriba un programa que devuelva un string con todas las combinaciones posibles al momento de lanzar tres dados de 6 caras. (1, 1, 1) (1, 1, 2) (1, 1, 3), ...

Escribe una función que se le pase un número y devuelve un string con ese mismo número separado por guiones. Por ejemplo 234 → “2-3-4”

(obligatorio) Es posible expresar 100 como la suma de tres cubos, cada uno de los cuales puede ser negativo o positivo. Sólo se conocen tres maneras de hacerlo. Una de ellas es la siguiente: $1870^3 - 1797^3 - 903^3 = 100$. Desarrolla un programa que busque las otras combinaciones para llegar al 100.

(obligatorio) Multiplicación rusa. El método de multiplicación rusa consiste en multiplicar sucesivamente por 2 el multiplicando y dividir por 2 el multiplicador hasta que el multiplicador tome el valor 1. Luego, se suman todos los multiplicandos correspondientes a los multiplicadores impares.

Dicha suma es el producto de los dos números. La siguiente tabla muestra el cálculo realizado para multiplicar 37 por 12, cuyo resultado final es $12 + 48 + 384 = 444$.

Mullicador	Multiplicando	Multiplicador impar	Suma
37	12	sí	12
18	24	no	
9	48	sí	60
4	96	no	
2	192	no	
1	384	sí	444

Crea una función, que reciba como parámetro un texto y lo escriba centrado en pantalla (suponiendo una anchura de 80 columnas; pista: deberás escribir $40 - longitud/2$ espacios antes del texto). Además subraya el mensaje utilizando el carácter `=`.

(obligatorio) Crea una función que quite los espacios por delante y por detrás de un string. Se considera un espacio: un espacio, un tabulador, un salto de línea y retorno de carro. La función recibe un string y dos booleanos.

(obligatorio) Crea un programa que calcula una aproximación de PI mediante la expresión:
 $\pi/4 = 1/1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 (\dots)$ A esta función se le pasará un entero con el número de iteraciones a realizar. Por ejemplo, si se le pasa un 4, el programa calculará:
 $p = 4 * (1/1 - 1/3 + 1/5 - 1/7)$

Escribir una función que reciba un número entero positivo y devuelva su factorial. Hay que hacer esta función de 2 formas, una iterativa y otra recursiva.

Escribir una función que reciba un número entero positivo y devuelva su sumatorio. Hay que hacer esta función de 2 formas, una iterativa y otra recursiva.

(obligatorio) Según Sheldon, el mejor número es el 73.

73 es el 21er número primo. Su espejo, 37, es el 12mo número primo. 21 es el producto de multiplicar 7 por 3. En binario, 73 es un palíndromo: 1001001.

Escriba programas que le permitan responder las siguientes preguntas:

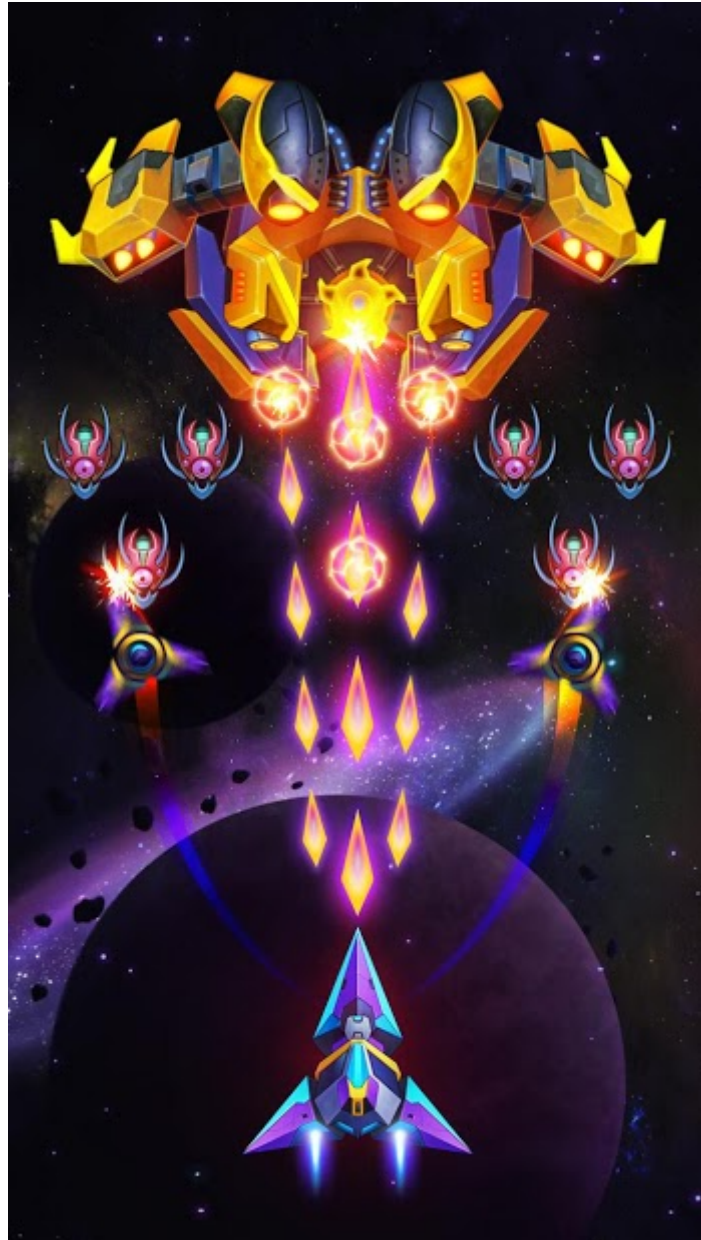
¿Existen otros valores p que sean el n -ésimo primo, tales que espejo(p) es el espejo(n)-ésimo primo?

¿Existen otros valores p que sean el n -ésimo primo, tales que n es el producto de los dígitos de p ?

¿Cuáles son los primeros diez números primos cuya representación binaria es un palíndromo?

Galaxy Invaders

Nota máxima 14



Para realizarla tienes que tener en cuenta lo siguiente:

- Hay que poner un fondo de pantalla.
- Cada objeto que sale por pantalla pertenece a una clase `GameObject`.
- Cada objeto tiene que tener un atributo que defina que tipo de objeto es.
- Hay que hacer una clase llamada `World`. Esta clase, entre otras, lleva el tamaño del campo de batalla.
- **IMPORTANTE**, no se puede cargar una imagen más de una vez.
- Cada vez que un disparo alcance un objetivo hay que crear otro objeto de tipo explosión con una animación.

- Los enemigos puedes hacerlos aparecer por bloques. Con que generes un bloque es suficiente.
- La nave protagonista no puede salir del escenario