Ejercicios de repaso

Operaciones básicas

1. Declarar dos variables, a y b de tipo entero, e imprimir la suma, resta, multiplicación, división y resto de la división de ambas.
2. Declarar dos variables, a y b de tipo real, e imprimir la suma, resta, multiplicación y división de ambas.
3. Solicitar al usuario que ingrese el precio de un artículo y un porcentaje de descuento, luego calcular e imprimir el precio final después del descuento.
4. Solicitar al usuario un número de segundos y calcular a cuántas horas, minutos y segundos corresponde. Por ejemplo: 125000 segundos son 34h 43m y 20s

Conversiones de tipo

1. Solicitar al usuario que ingrese un número entero y luego imprimir el doble de ese número como un número real (con decimales).
2. Solicitar al usuario que ingrese un número real (con decimales) y luego dividirlo entre 2 y mostrar el resultado como un número entero, es decir, eliminando los decimales.
3. Solicitar al usuario un carácter y mostrar su código ascii equivalente.
4. Solicitar al usuario un número entre 0 y 127 y mostrar a qué carácter del código ascii corresponde.

Estructuras condicionales

1. ¿De día o de noche? Crea un programa que solicite la hora al usuario en formato 24h [0 - 23] e indique si es de día o de noche, suponiendo que se hace de día a las 7 de la mañana y que se hace de noche a las 20 horas.
2. La entrada al cine. Crea un programa que permita calcular el precio de la entrada al cine. El programa preguntará al usuario la edad, si es menor de 5 años puede entrar gratis, si tiene entre 5 y 10 años debe pagar 3 €, si tiene entre 11 y 17 años debe pagar 5 € y si es mayor de edad, 7 €.
3. La entrada al cine con descuento. Modifica el programa anterior teniendo en cuenta que los clientes que pertencen a una familia numerosa tienen derecho a un 20% de descuento y que los clientes con una discapacidad superior al 33% tienen un descuento del 50%. Suponer que los descuentos se pueden acumular.
4. Combate de Pokémons. Desarrolla un juego para 2 jugadores que permita simular el combate entre dos Pokémon. Cada Pokémon tiene un valor de ataque que es el daño que infligirá al rival y un valor de defensa que es cuando cuanto del ataque del rival será capaz de mitigar. Cada jugador dispone de 10 puntos que puede repartir entre ataque y defensa. A iniciar el juego se solicitará a cada jugador qué valor quiere destinar al ataque y el restante hasta llegar a 10, será el valor asignado a la defensa. Una vez los jugadores hayan introducido los valores, dará inicio el juego realizando cada Pokémon su ataque, el Pokémon que haya conseguido hacer más daño será el ganador.

Strings y Bucles

1. En ASCII por favor. Crea un programa que solicite al usuario un texto y lo muestre en su equivalente ASCII.
2. Una carrera afortunada. Crea un juego que simule una carrera entre 2 "corredores". Los corredores avanzan de acuerdo con tiradas de dados simuladas (dos dados). En cada tirada pueden recorrer de 1 a 12 metros. El primero en recorrer 100 metros gana.
3. Encriptación de contraseña. Solicita al usuario una contraseña y un número (que actuará como clave de cifrado) encripte la contraseña utilizando el cifrado César. En este cifrado el número indica las posiciones que hay que desplazar cada letra en el abecedario. Por ejemplo: para la constraseña CAFE y el número 3 (clave de cifrado) habrá que desplazar cada letra 3 posiciones en el abecedario quedando como contraseña encriptada FDIH. En caso de llegar al final del abecedario volvemos a empezar por el principio.
4. Desencriptación de contraseña. Crea un programa que solicite la contraseña encriptada con codificación César y la clave de cifrado (el número de desplazamientos) y permita desencriptarla haciendo el proceso inverso.
5. Combate de Pokémons por turnos. Modifica el juego de Pokémon realizado en el apartado anterior añadiéndole el atributo vida a los Pokémon. De esta forma los jugadores deberán repartir los 10 puntos entre 3 valores (ataque, defensa y vida). Además, al realizar un ataque y al defender se generará un valor aleatorio entre 0 y 20 que representará en porcentaje la pérdida de efectividad del ataque o la defensa. Por ejemplo: un Pokémon con los siguientes valores ataque: 3, defensa: 4 y vida: 3, si sale un valor aleatorio de 5 para el ataque y 10 para la defensa, representará que el ataque perderá un 5% de efectividad, es decir su valor real será 2.85 y la defensa perderá un 10% de efectividad, por tanto quedará como 3.6. El juego se jugará por turnos y ganará el Pokémon que antes consiga reducir la vida del rival a 0.

Métodos

1. Factores primos. Desarrolla un programa que lea un número (controlando que sea positivo), y visualice en pantalla todos sus factores primos en la forma "A exp B " siendo A el factor primo y B el exponente.

   Descomposición factorial del nº 8

   2 15 3

   2 8 = 2 exp 3 5 5 15 = 3 exp 1 x 5 exp 1

   2 1

   1
2. La serie de Fibonacci. Desarrolla un método que muestre los N primeros términos de la serie de Fibonacci, y la suma de estos términos. La serie de Fibonacci está formada por una secuencia de números en la que el primer y segundo término son el 0 y el 1, y a partir del segundo, cada término es suma de los dos anteriores.
3. Un número perfecto. Desarrolla un método que determine si un número es perfecto o no. El método debe comprobar que el número recibido como parámetro sea positivo. Un número es

perfecto cuando la suma de sus divisores más una unidad es igual al número.

6 = 2 + 3 + 1 Perfecto

10 = 2 + 5 + 1 No perfecto

4. Una cuenta extraña. Desarrolla un programa que dada una serie de números mayores de 0, cuente y sume por separado, los pares, los impares, los primos y los perfectos. Teniendo en cuenta que un número puede cumplir a la vez varias de las condiciones anteriores, (2 es par y primo). El cero finaliza la serie y un número negativo no se considera.
5. Números amigos. Dados dos números decir si son o no, amigos.

   Dos números son amigos si cada uno de ellos es igual a la suma de los divisores del otro.

   284 divisores 1+2+4+71+142 = 220

   220 divisores 1+2+4+5+10+11+20+22+44+55+110 = 284
6. Parejas de amigos. Desarrolla un programa que muestre todas las parejas de números amigos menores o iguales que m, siendo m un número introducido por teclado.

Arrays

1. Swap. Desarrolla un método que reciba como parámetros un array de enteros, y dos valores numéricos que indicarán 2 posiciones del array e intercambie los elementos del array que ocupan dichas posiciones.
2. La carrera de caracoles. Tenemos los siguientes datos de una carrera de caracoles:

   CaracolA = {2, 3, 4, 4, 5, 3, 3, 4, 3, 4, 4}

   CaracolB = {3, 5, 3, 3, 1, 3, 2, 3, 4, 3, 3}

   Los arrays representan las aceleraciones medias (en m/s) de los caracoles en cada segundo. Sabiendo que la distancia a recorrer de la carrera era de 100 metros ¿Podrías determinar qué caracol ganó la carrera y en qué instante de tiempo?
3. El reproductor musical. Crea un programa que permita mostrar las notas musicales de una canción. Para ello debes utilizar 2 arrays: el primer array representará las notas y el segundo array la duración de cada nota. Para simplificar el ejercicio vamos a implementar una única octava, donde estarán representadas las siguientes notas: DO, DO#, RE, RE#, MI, FA, FA#, SOL, SOL#, LA, LA#, SI

   La reproducción consistirá en mostrar en orden cada nota y su duración.

   Para comprobar el funcionamiento, genera un array con 50 notas aleatorias y el otro array con 50 duraciones. Las duraciones podrán variar entre 1 y 4.
4. Cifrado con sustituciones y desplazamientos. Tomando como base el ejercicio de cifrado con codificación César, crea un sistema de cifrado que luego aplique sustituciones basadas en un array de carácteres. El número de la clave será el desplazamiento sobre las posiciones del array.