GO

RACKET

Ejercicios que no sirven

Ya están ingresados en el csv

1. Crear un programa que permita calcular el volumen de un cilindro, una esfera y un cono. Las ecuaciones de las respectivas figuras son:

Cilindro = 2 \* h Esfera = 3  Cono= 2 \*

1. Calcular el número más grande de 5 números dados.
2. Realizar las operaciones básicas con números de tipo entero y números de tipo flotante.
3. En un teatro cada cliente paga $10.000 por entrada y cada función le cuesta al teatro $300.000 por la atención prestada. Por cada cliente que entre el teatro debe pagar un costo de $2.000 por aseo. Desarrollar un programa que reciba el número de clientes de una función y devuelva el valor de las ganancias obtenidas.
4. En un supermercado se ofrecen descuentos por el total del valor en cada uno de los siguientes productos: carnes con 10%, frutas con el 5%, aseo con el 7%, dulces con el 9%.
5. Una persona desea llevar $10.550 en dulces, $50.000 en carne y $35.000 en productos de aseo. Se necesita calcular el descuento total por cada tipo de producto, posteriormente entregar el valor total a pagar con descuento y sin descuento.
6. Verificar si dos números ingresados son iguales.
7. Verificar si un número es menor a 10.
8. Hacer un programa que determine verdadero o falso si un número es mayor o igual que 10 y menor que 20 o mayor que 30.
9. Dado un número de 3 cifras, descomponerlo en sus unidades fundamentales y decir la cantidad correspondiente, ejemplo: el número 631 tiene una unidad, 3 decenas y 6 centésimas.
10. Del ejercicio anterior, descomponer un numero dado e invertirlo, por ejemplo, 582 pasaría a ser 285.
11. Del capítulo “*2. Estructuras de un programa en GO*”, desarrollar los ejercicios propuestos en modo de funciones.
12. Crear un programa que determine el mayor de 3, 4 y 5 números
13. Se necesita crear un programa que reciba la fecha de un año e indique si este es o no bisiesto.
14. Crear un programa que, dado un intervalo de números entero, verifique si la suma de los elementos comprendidos en el intervalo da como resultado un numero primo.
15. Construir un programa que reciba un numero de n dígitos y devuelva (según el número de dígitos) las unidades utilizadas con su cantidad. Por ejemplo:
16. Se recibe 356, se debe retornar como respuesta:

* 6 unidades
* 5 decenas
* 3 centenas

1. Construir una función que reciba un número entero y cuente el número de veces que se repite un dígito (si existe).
2. Crear una función que permita llenar un vector de n digitas (por teclado) y calcule cual es el mayor de todos los elementos.
3. Crear una función que permita ordenar un vector de mayor a menor y menor a mayor.
4. Construir un programa que, dados dos vectores, devuelva dos nuevos vectores que junten los números pares e impares de cada uno.
5. Crear un programa capaz de recibir una lista de datos y calcular la media, mediana, moda y desviación estándar del conjunto de datos.
6. Construir una función que, dadas dos matrices, calcule la suma y multiplicación de sus elementos. (Tener en cuenta propiedades de las matrices)
7. Construir una función en donde dada una matriz, calcule su matriz transformada. (Tener en cuenta propiedades de las matrices)
8. Construir un programa que pueda recrear el juego *Tic-tac-toe*.Serán dos jugadores posibles y gana quien alcance a completar 3 casillas en línea o diagonal, por ejemplo:

|  |  |  |
| --- | --- | --- |
| X | O | O |
| X | O | X |
| O | O | X |

1. Un cliente ordena una cierta cantidad de brochas de cerda, rodillos y sellador; las brochas de cerda tienen un 20% de descuento y los rodillos un 15% de descuento. Los datos que se tienen por cada tipo de articulo son: La cantidad pedida y el precio unitario. Además, si se paga de contado todo tiene un descuento del 7%. Elaborar un programa que permita visualizar el precio total de una cierta orden de n cantidad productos, tanto para pago de contado como pago a crédito.
2. Realice un programa que calcule el sueldo que le corresponde al trabajador de una empresa que paga 60.000 dólares anuales. El programa debe realizar los cálculos en función de los siguientes criterios
   1. Si lleva más de 10 años en la empresa se le aplica un aumento del 10%.
   2. Si lleva menos de 10 años, pero más que 5 se le aplica un aumento del 7%.
   3. Si lleva menos de 5 años, pero más que 3 se le aplica un aumento del 5%.
   4. Si lleva menos de 3 años se le aplica un aumento del 3%.
3. En la tienda MercaFacil, el impuesto pagado por la compra de articulos se calcula de la siguiente manera:
   * + Los primeros $30.000 pesos no cobran impuesto
     + Los siguientes $30.000 pesos tienen un 30% más de impuesto y el resto un 40% más de impuesto.
     + Si el costo de la venta total es mayor a $85.000 pesos, entonces solo se cobra el 50% sobre el total de la venta.

Generar un programa que, ingresando el valor total de la compra, permita calcular el resultado total de la venta luego de aplicar impuestos.

1. El jefe del departamento de construcción de la constructora **FERIESPACIOS** desea desarrollar un programa para sus empleados que calcule el sueldo total para cada uno según el número de horas trabajadas. Para ello se tienen en cuenta los siguientes criterios:
   * + Sí el número de horas trabajadas es mayor a 40, el excedente de las 40 horas se paga al doble del valor de la hora. En caso contrario se paga la hora al valor normal.
     + Si las horas exceden las 50 horas trabajadas el excedente se paga al triple del valor de la hora y se les descontará un impuesto del 12% del total del sueldo ganado.

Realizar un programa que, pidiendo el nombre del trabajador y el número de horas trabajadas muestre en pantalla el nombre del trabajador y el cálculo del sueldo total ganado. Parta del hecho que la hora trabajada tiene un valor de $8.500 pesos.

1. En un curso de programación II se hacen cuatro pruebas de las cuales el profesor obtiene 4 notas, cada una de las cuales está entre 0 y 5. En vista de las muy buenas notas que obtuvieron los estudiantes del grupo 35, el profesor decide que la nota final no será el promedio aritmético de las cuatro notas, sino que hará lo siguiente con las notas de cada uno de sus estudiantes:
   * + Eliminará la menor de las 4 notas
     + La mayor nota tendrá un porcentaje del 50%
     + Cada una de las dos notas restantes tendrá un porcentaje del 25%

De esta forma, la nota final de un estudiante que obtuvo las notas 3.0, 2.0, 4,5 y 3,2, será 3.8. Haga un programa que lea las 4 notas de un estudiante y calcule y escriba su nota definitiva.

1. Una compañía inmobiliaria requiere calcular fácilmente el precio de unos lotes que ha ofrecido para la venta. Para ello la inmobiliaria ofrece lotes desde 70 metros cuadrados hasta lotes de 900 metros cuadrados. El plan de descuentos es el siguiente: Si el lote es mayor a 400 metros cuadrados el descuento será del 25%, si tiene entre 400 y 700 metros cuadrados el descuento será del 17 % y para los lotes con área entre 700 y 900 metros el descuento es del 10 %. Para efectos del cálculo del área, se conoce el largo, el ancho de cada lote y el precio por metro cuadrado.
2. Cree un programa que permita calcular el numero de la secuencia Fibonacci correspondiente a un número que indique su posición, por ejemplo: si se ingresa el número 6 , el resultado deberá ser 8 ya que:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Posición | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Sucesión de Fibonacci | 0 | 1 | 1 | 2 | 3 | 5 | 8 |

1. Se necesita escribir un algoritmo para convertir un valor de temperatura de la escala Celsius a otras escalas de temperatura (Farenheit y Kelvin). El algoritmo debe solicitar al usuario un valor de temperatura, y la escala a la cual se quiere convertir dicho valor y debe reportar como resultado el valor de temperatura en la nueva escala.
2. Crear un programa que reciba una serie de números y calcule la sumatorio de todos estos.
3. Escribir una función que calcule cuantos números pares hay comprendidos entre dos números límites (sin incluirlos).
4. Crear un programa que calcule si un número es primo o no.
5. Crear un programa que simule el comportamiento de un reloj digital, escribiendo en formato de horas, minutos y segundos, iniciando desde las 00:00:00 horas hasta las 23:59:59 horas
6. Hacer un programa que pida un número y muestre en pantalla su tabla de multiplicar.
7. Generar los primeros diez números perfectos.

**Nota:** Un Número perfecto es aquel que es igual a la suma de sus divisores, excluyéndose el propio número. Por ejemplo, el número 28 presenta 5 divisores menores y distintos de 28, que son: 1, 2, 4, 7 y 14. Al sumarlos da como resultado 28

1. Según historias cuentan que el inventor del juego de ajedrez, el rey que le solicitó hacerlo le preguntó cómo quería que este le pagase, a lo cual el inventor le contesto que ubicara un grano de trigo en el primer cuadro del tablero, en el segundo ubicara el doble del primero, en el tercero el doble del segundo y así sucesivamente hasta completar los 64 cuadrados.

Se debe crear un programa que muestre un tablero (Puede ser sin líneas) en donde se indique el número de granos de trigo debió colocar el rey por cada cuadro.

Ejemplo:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 12 | 11 | 10 | 9 | 8 | 7 |
| 13 | 14 | 15 | 16 | 17 | 18 |
| 24 | 23 | 22 | 21 | 20 | 19 |
| 25 | 26 | 27 | 28 | 29 | 30 |
| 36 | 35 | 34 | 33 | 32 | 31 |
| 37 | 38 | 39 | 40 | 41 | 42 |

Nota: El ejemplo es ilustrativo, se debe basar en el tablero de ajedrez.

41. Realizar las siguientes figuras para *n* niveles:

|  |  |
| --- | --- |
| \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  #2 | \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  #3 |
| \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \*  #4 | \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \*  #5 |
| |  |  |  |  | | --- | --- | --- | --- | | \* |  |  |  | | \* | \* |  |  | | \* | \* | \* |  | | \* | \* | \* | \* | | \* | \* | \* |  | | \* | \* |  |  | | \* |  |  |  |   #6 | |  |  |  |  | | --- | --- | --- | --- | |  |  |  | \* | |  |  | \* | \* | |  | \* | \* | \* | | \* | \* | \* | \* | |  | \* | \* | \* | |  |  | \* | \* | |  |  |  | \* |   #7 |

1. Crear el siguiente cuadrado:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  | \* |  |  | |  | \* | \* | \* |  | | \* | \* | \* | \* | \* | |  | \* | \* | \* |  | |  |  | \* |  |  |   #6 |

1. Crear el siguiente conjunto de figuras:

|  |
| --- |
| - - - = +  - - = + +  - = + + +  = + + + +  #1 |

|  |  |
| --- | --- |
| A A  B B  C C  C C  B B  A A  A A  B B  C C  C C  B B  AA  #2 | \* \* \* \* \*  \* \* \* \*  \* \* \*  \* \*  \*  \* \*  \* \* \*  \* \* \* \*  \* \* \* \* \*  #3 |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | \* |  |  |  |  | \* | | \* | \* |  |  | \* | \* | | \* | \* | \* | \* | \* | \* | | \* | \* | \* | \* | \* | \* | | \* | \* |  |  | \* | \* | | \* |  |  |  |  | \* |   #4 | \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \*  \* \* \* \* \*  #5 |

1. Leer número e imprimir una pirámide de dígitos:

|  |
| --- |
| 1  121  12321  1234321  123454321  #6 |

1. Leer un número positivo e imprimir las sumas de números enteros positivos consecutivos que den el número introducido. Por ejemplo:

50 = 8 9 + 10 + II + 12

50 = II + 12 + 13 + 14

1. Hacer una función recursiva que reciba una lista y retornar la suma de sus elementos.
2. Dada una lista compuesta por números enteros, retornar la sumatoria de todos los números pares.
3. Hacer una función recursiva que reciba una lista de enteros positivos y devolver el mayor valor contenido en la lista (o-1) si está vacía.
4. Hacer un programa que reciba una lista compuesta de números enteros e indicar si está ordenada ascendentemente.
5. Dada una lista desordenada de números enteros, ordenar dicha lista de mayor a menor (no utilizar la función sort). Se deben implementar los algoritmos de ordenamientos.
6. Dada una lista compuesta por cadenas, construir una nueva lista formada por los elementos que no estén repetidos. Ejemplo: (list “hola” “mundo” “mundo”) la nueva lista será (list “hola”).
7. Hacer una función que reciba una lista de números y un dato numérico, si este número no se encuentra en la lista, ingresarlo a ella en la parte final.
8. Hacer un programa que reciba nombres (cadenas de caracteres) y devuelva el que tenga mayor longitud.
9. Dada una lista compuesta por tres puntos ((list 1 2),[[1,2]…] es un punto donde x = 1; y = 2), retornar verdadero si dichos puntos forman un triángulo equilátero, y retornar falso en caso contrario.
10. Dada una lista compuesta por cadenas, retornar una nueva lista compuesta por las cadenas, pero sin sus vocales. Ejemplo: (list “hola” “mundo”) retornará (list “hl” “mnd”).
11. Calcular la serie de Fibonacci implementando los modelos de pilas y colas
12. Usando las colas, calcular la sumatoria de n números.
13. Realizar la multiplicación de números sin usar las funciones prediseñadas. Utilice las colas para la recursión.
14. Hacer un programa que calcule los números primos de una lista y guarde los datos en una pila. También debe introducir un número a buscar en la pila, para luego mostrar en pantalla la información almacenada hasta llegar al dato pedido.
15. Ordenar un vector de números enteros teniendo en cuenta el funcionamiento de las pilas.
16. Usando las pruebas de escritorio, describir el funcionamiento de selección de un elemento usando las pilas y las colas.
17. Crear una función que reciba un árbol binario y agregue a este un nuevo nodo.
18. Crear una función que reciba un árbol binario y un dato. Recorrer el árbol y eliminar el dato ingresado
19. Crear una función que reciba un árbol binario con números enteros. Usar los tres tipos de recorrido (pre-orden, in-orden, post-orden) y mostrar en pantalla el resultado.
20. Crear una función que reciba un árbol y lo organice de menor a mayor.
21. Definir una función que reciba un árbol de números enteros. Esta deberá retornar una nueva estructura en la que cada nodo después de la raíz tendrá una lista de tres elementos, el primero y el segundo serán los hijos de ese nodo y el tercero será la suma de los anteriores. Si el nodo es una hoja, en lugar de una lista se debe ingresar un cero.
22. Crear una función que determine cuantas raíces no nulas tiene un árbol binario. Una raíz no nula es aquella que tiene por lo menos un hijo (ya sea de izquierda o derecha).
23. Elaborar una función que reciba en tiempo de ejecución tres coordenadas del plano cartesiano, cada punto será una estructura. Al final deberá retornar una lista con los 3 puntos organizados mayor a menor a partir del concepto de distancia al origen.
24. Elaborar una función que reciba como parámetro una estructura llamada persona, esta devolverá una estructura derivada llamada trabajador. Deberá pedir en tiempo de ejecución el NIT y el número de teléfono.
25. Realizar un programa que reciba como argumentos una lista de estructuras (personas), la cual contara también con un campo nombre; el programa debe retornar nuevamente una lista de estructuras ordenadas alfabéticamente (solo por iniciales).
26. Realizar un programa que reciba una lista de estructuras de tipo fecha, y que retorne una lista ordenada con las fechas anteriores a posteriores. Si algún elemento no existe o no es del tipo fecha, retornar una lista vacía.
27. Realizar un programa que arroje un menú con las siguientes opciones:

-Equipo ganador

-Historial de partidos

Con ello deberá realizar funciones que reciban como parámetros estructuras de tipo partido, con campos equipo y goles; para la primera opción se tendrá que mostrar el equipo ganador del total de partidos jugados; en la segunda deberá mostrar los partidos ordenados descendentemente.

1. A partir del ejercicio No. 3 expuesto en “Archivos”, realice un programa que reciba como parámetros el nombre del archivo, un carácter a buscar y otro el cual será el sustituto. El programa deberá de cambiar n todo el archivo el carácter buscado por su reemplazo.
2. Hacer una función que reciba un archivo y calcule el número de líneas que lo componen.
3. Hacer un programa que reciba un archivo con un texto almacenado. Las palabras que se encuentren en la posición par serán guardadas en un archivo llamado “par.txt”, de igual forma, las impares irán en un archivo “impares.txt”. El documento original elimina.
4. Hacer un programa que desencripte o reconstruya el mensaje a partir de los dos archivos generados, “par.txt” e “impar.txt”.
5. Hacer una función que reciba un archivo con texto. A cada posición que sea un numero primo, se le deberá de sumar 2 al valor del carácter. Al final deberá guardar el nuevo texto en un nuevo documento.
6. Hacer un programa el cual al leer un archivo de texto, pueda contar el número de palabras que componen al archivo.
7. A través de una estructura *persona*, crear un archivo que guarde 5 registros de la estructura persona, una línea por registro. Se deben usar las características mas notables de una persona

|  |  |  |
| --- | --- | --- |
| cambiar los caracteres de las posiciones que sean un numero primo de un archivo de texto sumandoles dos unidades a su valor corresponidente en ASCII | hacer | expresion |
| tener un funcion que lea la informacion de un archivo de texto | condicion-info | expresion |
| identificar y cambiar los caracteres del texto que sean una posicion con numero primo aumentados en dos unidades de su equivalente en ASCII | condicion-info | expresion |
| realizar y guardar los cambios en un nuevo documento | condicion-info | expresion |
| lectura del archivo dada por funciones dedicadas | entrada | expresion |
| nuevo documento de texto creado con los cambios solicitados | salida | expresion |
| - | - | - |
| separar en dos archivos de texto llamados par e impar las palabras que se encuentren en posiciones pares e impares respectivamente de otro documento leido previamente | hacer | expresion |
| separar en un archivo las palabras pares | condicion-info | expresion |
| separar en un archivo las palabras impares | condicion-info | expresion |
| lectura del archivo dada por funciones dedicadas | entrada | expresion |
| los dos documentos par e impar y el documento reconstruido ambas categorias de tipo texto | salida | expresion |
| eliminar el archivo usado para lectura de texto | condicion-info | expresion |
| tambien se debe crear alguna funcion o nuevo programa que permita recrear el archivo que fue originalmente leido y luego eliminado | hacer | expresion |
| separar en un archivo la reconstruccion del archivo original | condicion-info | expresion |
| - | - | - |
| desarrollar una funcion que lea un archivo y calcule el numero de lineas que lo componen | hacer | expresion |
| archivos que contengan texto y que el formato sea texto | condicion-info | expresion |
| lectura del archivo dada por funciones dedicadas | entrada | expresion |
| un numero entero que representa el numero de lineas que componen al texto del archivo | salida | expresion |
| - | - | - |
| realizar un programa que permita visualizar un menu de dos opciones para un equipo ganador y un historial de partidos | hacer | expresion |
| estas opciones se calcularan segun ciertas condiciones | hacer | expresion |
| usar estructuras para el concepto de equipos de futbol | condicion-info | expresion |
| la estructura equipo debe tener las caracteristicas mas notorias de un equipo de futbol de la vida real | condicion-info | expresion |
| poder registrar equipos de futbol | condicion-info | expresion |