LENGUAJE C

PYTHON

JAVASCRIPT

JAVA

# Ejercicios Básicos

## Ejercicio 1

Escribir un programa que muestre en pantalla ¡Hola Mundo!.

## Ejercicio 2

Escribir un programa que guarde en una variable ¡Hola Mundo! Y muestre esa variable por pantalla.

## Ejercicio 3

Se ingresan las notas de dos evaluaciones de un alumno. Determinar la nota promedio.

## Ejercicio 4

Escribir un programa que pregunte al usuario la cantidad de horas que trabajó en la jornada y el coste por hora. Mostrar por pantalla la paga que le corresponde en el día.

## Ejercicio 5

Escribir un programa que pregunte el nombre del usuario en consola y después de que el usuario lo introduzca muestre por pantalla la cadena ¡Hola <nombre>, donde <nombre> es el nombre que el usuario haya introducido.

## Ejercicio 6

Escribir un programa que muestre por pantalla el resultado de la siguiente operación aritmética ((3+22)/5)²

## Ejercicio 7

Confeccionar un programa que pueda ingresar 2 números enteros y que calcule e informe mensajes aclaratorios la suma, el producto, el cociente y el resto.

## Ejercicio 8

Un usuario quiere abrir una caja de ahorro en un banco, el cual ofrece un 5% de interés anual. Esta remuneración se añade al balance final de la cuenta, para luego ser cobrada por el usuario cumpliendo el plazo. Escribir un programa que permita ingresar al usuario el monto de la cuenta y calcule la cantidad de ahorros tras el primer año, segundo año y tercer año.

## Ejercicio 9

Confeccionar un programa que ingrese una medida en ‘pies’ y la exhiba convertida a yardas, pulgadas, centímetros y metros.

NOTA: 1 pie = 12 pulgadas, 1 yarda = 3 pies, 1 pulgada = 2.54cms.

## Ejercicio 10

Una panadería vende pan a $300 el kg y medialunas a $30 cada una. Debido a que el pan que queda es de ayer, se le aplicará un descuento del 40%. Escribir un programa que solicite cuantos kilos de pan y cuantas medialunas se quieren comprar y muestre por pantalla el monto a pagar.

## Ejercicio 11

Escribir un programa que pida al usuario su peso (en Kg.) y estatura (en Metros). Calcule el índice de masa corporal, guárdelo en una variable, y muestre por la pantalla “Tu índice de masa corporal es <imc> donde <imc> es el índice de masa corporal calculado redondeado con dos decimales.

NOTA: Fórmula: peso (kg) / [estatura (m)]2

## Ejercicio 12

Escribir un programa que pida al usuario dos números enteros y muestre por pantalla <n> entre <m> da un cociente <c> y un resto <r> donde <n> y <m> son los número introducidos por el usuario, <c> el cociente y <r> el resto de la división entera respectivamente.

## Ejercicio 13

Una juguetería tiene mucho éxito en dos de sus productos: payasos y muñecas. Suele hacer venta por correo y la empresa de logística les cobra por peso de cada paquete así que deben calcular el peso de los payasos y muñecas que saldrán en cada paquete a demanda. Cada payaso pesa 112 g y cada muñeca 75 g. Escribir un programa que lea el número de payasos y muñecas vendidos en el último pedido y calcule el peso total del paquete que será enviado.

## Ejercicio 14

Conociendo la cantidad de tarros de pintura que existe en un depósito de una pintureria y sabiendo que el 50% son tarros de 1Lt, el 30% tarros de 4Lts, y el resto de tarros de 20Lts. Determinar la cantidad de tarros de 1Lt, 4Lts y 20Lts.

## Ejercicio 15

El precio para un vuelo es de $8800 en clase turista y se aplica un incremento del 30% en primera clase. Se ingresan la cantidad de pasajes vendidos de clase turista y de primera clase. Obtener la recaudación total del vuelo.

## Ejercicio 16

Confeccionar un programa que ingrese n valor expresado en Kibibyte (KiB) y lo informe expresado en: TiB, GiB, MiB, con leyendas aclaratorias.

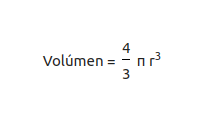
NOTA: 1Mib = 1024KiB, 1GiB = 1024Mib, 1TiB = 1024GiB.

## Ejercicio 17

Confeccionar un programa que solicite el ingreso del valor radio (r) de un circulo y con dicho valor calcule la superficie del círculo, la circunferencia (perímetro) y el volumen de la esfera.

Superficie = π r2

Perímetro = 2π r



## Ejercicio 18

Escribir un programa que calcule cuántos litros de combustible consumió un automóvil. El

usuario ingresará una cantidad de litros de combustible cargados en la estación y una

cantidad de kilómetros recorridos, después, el programa calculará el consumo (km/lt) y se lo

mostrará al usuario.

## Ejercicio 19

Escriba un programa que permita al usuario ingresar el valor de dos variables numéricas de

tipo entero. Posteriormente, el programa debe intercambiar los valores de ambas variables y

mostrar el resultado final por pantalla.

Por ejemplo, si el usuario ingresa los valores num1 = 9 y num2 = 3, la salida a del programa

deberá mostrar: num1 = 3 y num2 = 9

Ayuda: Para intercambiar los valores de dos variables se debe utilizar una variable auxiliar.

## Ejercicio 20

Ingrese un número de tres cifras y muestra la unidad, decena y la centena.

Por ejemplo, si el número ingresado por pantalla es 123 el programa debe mostrar:

CENTENA: 1

DECENA: 2

UNIDAD: 3

## Ejercicio 21

Un colegio desea saber qué porcentaje de niños y qué porcentaje de niñas hay en el curso

actual. Diseñar un algoritmo para este propósito. Recuerda que para calcular el porcentaje

puedes hacer una regla de 3 simple. El programa debe solicitar al usuario que ingrese la

cantidad total de niños, y la cantidad total de niñas que hay en el curso.

## Ejercicio 22

Solicitar al usuario que ingrese la base y altura de un rectángulo, y calcular y mostrar por

pantalla el área y perímetro del mismo

area = base \* altura

perimetro = 2 \* altura + 2 \* base.

## Ejercicio 23

Escribir un programa que calcule el volumen de un cilindro. Para ello se deberá solicitar al

usuario que ingrese el radio y la altura. Mostrar el resultado por pantalla.

volumen = π \* radio2 \* altura

## Ejercicio 24

A partir de una conocida cantidad de días que el usuario ingresa a través del teclado, escriba

un programa para convertir los días en horas, en minutos y en segundos. Por ejemplo

1 día = 24 horas = 1440 minutos = 86400 segundos

## Ejercicio 25

Crear un programa que solicite al usuario que ingrese el precio de un producto al inicio del

año, y el precio del mismo producto al finalizar el año. El programa debe calcular cuál fue el

porcentaje de aumento que tuvo ese producto en el año y mostrarlo por pantalla.

## Ejercicio 26

Calcular el cambio de monedas en dólares y euros al ingresar cierta cantidad de dinero en

pesos.

## Ejercicio 27

Hacer un programa que calcule el salario de un empleado, que descuenta el 20% de su

salario bruto.

## Ejercicio 28

Hacer un programa que ingrese por teclado un número total de segundos y que luego

pueda mostrar la cantidad de horas, minutos y segundos que existen en el valor

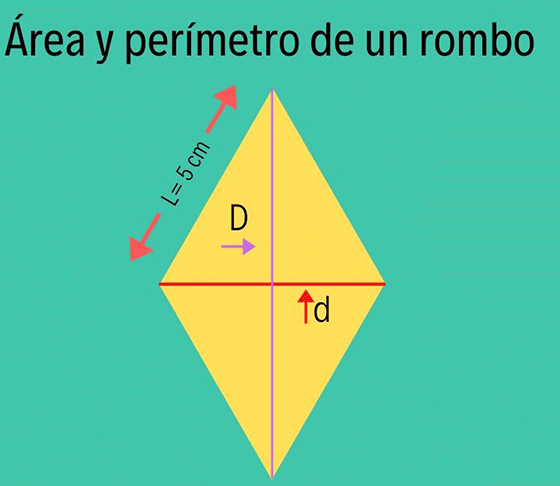
ingresado.

## Ejercicio 29

Mostrar el área y perímetro de un rombo.

Perímetro = 4L

Área = (Dxd)/2



## Ejercicio 30

Mostrar el área y perímetro de un hexágono.

Perímetro = 6. LADO = 6L

Area = (perímetro/apotema)/2

Apotema = raíz cuadrada(lado^2 + (lado/2)^2)

## Ejercicio 31

Mostrar el área y perímetro de un paralelogramo.

## Ejercicio 32

Escribir un programa que convierta un valor dado en grados Fahrenheit a grados Celsius.

NOTA: 1 °F es el equivalente de una diferencia de temperatura de 0,556 °C.

## Ejercicio 33

Un vendedor recibe un sueldo base más un 10% extra por comisión de sus ventas, el

vendedor desea saber cuánto dinero obtendrá por concepto de comisiones por las tres

ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo

base y comisiones.

## Ejercicio 34

Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea

saber cuánto deberá pagar finalmente por su compra.

## Ejercicio 35

Un alumno desea saber cuál será su calificación final en la materia de Algoritmos. Dicha

calificación se compone de los siguientes porcentajes:

a. 55% del promedio de sus tres calificaciones parciales.

b. 30% de la calificación del examen final.

c. 15% de la calificación de un trabajo final.

## Ejercicio 36

Pide al usuario dos números y muestra la “distancia” entre ellos (el valor absoluto de su

diferencia, de modo que el resultado sea siempre positivo).

## Ejercicio 37

Realizar un algoritmo que lea un número y que muestre su raíz cuadrada y su raíz cúbica.

PSeInt no tiene ninguna función predefinida que permita calcular la raíz cúbica, ¿Cómo se

puede calcular?

## Ejercicio 38

Dado un número de dos cifras, diseñe un algoritmo que permita obtener el número

invertido. Ejemplo, si se introduce 23 que muestre 32.

## Ejercicio 39

Un ciclista parte de una ciudad A a las HH horas, MM minutos y SS segundos. El tiempo de

viaje hasta llegar a otra ciudad B es de T segundos. Escribir un algoritmo que determine la

hora de llegada a la ciudad B.

# Ejercicios condicionales IF

## Ejercicio 40

Escribir un programa que pregunte al usuario su edad y muestre por pantalla si es mayor de edad o no.

## Ejercicio 41

Escribir un programa que almacene la cadena de caracteres contraseña en una variable, y luego que pregunte al usuario por la contraseña e imprima por pantalla si la contraseña introducida por el usuario coincide con la guardada en la variable sin tener en cuenta mayúsculas y minúsculas.

## Ejercicio 42

Escribir un programa que pida al usuario dos números y muestre por pantalla su división. Si el divisor es cero, el programa debe mostrar error.

## Ejercicio 43

Escribir un programa que pida al usuario un número entero y muestre por pantalla si es par o impar.

## Ejercicio 44

Una pizzería vende empanadas por unidad o por docena. La docena cuesta $300, pero si se compra individualmente, se cobra $30 la unidad. Si se compran más empanadas que no se agrupen en docenas, las adicionales se cobran por unidad. Indicar el precio total a abonar.

## Ejercicio 45

Una farmacia vende algunos artículos sin descuentos y a otros con descuentos del 20%. Confeccionar un programa que recibiendo el precio original y un código que indica si es o no descuento, informe el precio final   
NOTA: 0 aplica descuento y 1 no aplica descuento.

## Ejercicio 46

Para tributar un determinado impuesto se debe ser mayor de 16 años y tener unos ingresos iguales o superiores a $1000 mensuales. Escribir un programa que pregunte al usuario su edad y sus ingresos mensuales y muestre por pantalla si el usuario tiene que tributar o no.

## Ejercicio 47

Los alumnos de un curso se han dividido en dos grupos, A y B de acuerdo al sexo y el nombre. El grupo A está formado por mujeres con nombre anterior a la M y los hombres con nombre anterior a la N, y el grupo B por el resto.

Escribir un programa que pregunte al usuario su nombre y sexo, y muestre por pantalla el grupo que le corresponde.

## Ejercicio 48

Un hombre desea saber si su sueldo es mayor al sueldo mínimo, el programa le pedirá al

usuario su sueldo actual y el sueldo mínimo. Si el sueldo es mayor al mínimo se debe

mostrar un mensaje por pantalla indicándolo.

## Ejercicio 49

Construir un código que permita ingresar un número, si el número es mayor de

500, se debe calcular y mostrar en pantalla el 18% de este.

## Ejercicio 50

Se pide ingresar una letra del alfabeto y mostrar si dicha letra es vocal o consonante.

## Ejercicio 51

Diseñe un algoritmo que lea un número de tres cifras y determine si es o no capicúa.

## Ejercicio 52

Crea una aplicación que nos pida un día de la semana y que nos diga si es un dia laboral o

no.

## Ejercicio 53

En medio de la auditoría interna, el equipo está cubriendo a un compañero de trabajo que

está de licencia. Su compañero de trabajo ha dejado un mensaje con las tareas a realizar.

“¡Hola! Muchas gracias por cubrirme. Lo principal que debes hacer es completar la hoja

de cálculo de ingresos mensuales. Puedes hacerlo buscando los últimos ingresos

publicitarios en los informes de marketing. Después de hacer todo eso, revisa mi correo

electrónico y si hay menos de 10 correos sin leer revisa mi correo de voz para ver si hay

alguna solicitud de los ejecutivos. Si hay tales solicitudes, hágalas primero a menos que

tenga una solicitud de emergencia de otro departamento. Una vez que hayas terminado

con la solicitud de cumplimiento, riegue la planta de mi escritorio después de apagar la

computadora. Ah, espera, debería haber mencionado un par de cosas: debes iniciar

sesión con usuario de administrador para ver los informes de marketing, y tendrás que

enviarme un correo electrónico de actualización justo después de que termines de

manejar las solicitudes. Bueno, gracias de nuevo. ¡Es de gran ayuda! Te debo el almuerzo

cuando regrese.” [Continua en la siguiente página]

Nuestra tarea será imprimir por pantalla la lista de tareas en el orden que corresponden

para que luego las podamos realizar. ¿Te animas a colocar las tareas en el orden correcto?

Para hacer esto, debes crear en PSeInt la cantidad de variables que creas correctas y

asignarles valor.

Por ejemplo:

•

•

cantidadEmails = 6

solicitudesEjecutivos = 3

## Ejercicio 54

Realiza un programa que sólo permita introducir los caracteres ‘S’ y ‘N’. Si el usuario

ingresa alguno de esos dos caracteres se deberá de imprimir un mensaje por pantalla que

diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”.

## Ejercicio 55

Realizar un programa que pida un número y determine si ese número es par o impar.

Mostrar en pantalla un mensaje que indique si el número es par o impar. (para que un

número sea par, se debe dividir entre dos y su resto debe ser igual a 0).

## Ejercicio 56

Realizar un programa que pida introducir solo frases o palabras de 6 caracteres. Si el

usuario ingresa una frase o palabra de 6 caracteres se deberá de imprimir un mensaje por

pantalla que diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”.

## Ejercicio 57

Realizar un programa que pida una frase o palabra y si la frase o palabra es de 4

caracteres de largo, el programa le concatenara un signo de exclamación al final, y si no

es de 4 caracteres el programa le concatenara un signo de interrogación al final. El

programa mostrará después la frase final.

## Ejercicio 58

Escriba un programa que pida 3 notas y valide si esas notas están entre 1 y 10. Si están

entre esos parámetros se debe poner en verdadero una variable de tipo lógico y si no

ponerla en falso. Al final el programa debe decir si las 3 notas son correctas usando la

variable de tipo lógico.

## Ejercicio 59

Escriba un programa que pida una frase o palabra y valide si la primera letra de esa frase

es una ‘A’. Si la primera letra es una ‘A’, se deberá de imprimir un mensaje por pantalla

que diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”.

## Ejercicio 60

Continuando el ejercicio anterior, ahora se pedirá una frase o palabra y se validara si la

primera letra de la frase es igual a la última letra de la frase. Se deberá de imprimir un

mensaje por pantalla que diga “CORRECTO”, en caso contrario, se deberá imprimir

“INCORRECTO”.

## Ejercicio 62

Una tienda ofrece para los meses de septiembre, octubre y noviembre un descuento del

10% sobre el total de la compra que realiza un cliente. Solicitar al usuario que ingrese un

mes y el importe de la compra. El programa debe calcular cuál es el monto total que se

debe cobrar al cliente e imprimirlo por pantalla.

## Ejercicio 63

Solicitar al usuario que ingrese dos números enteros y determinar si ambos son pares o

impares. Mostrar en pantalla un mensaje que indique “Ambos números son pares” siempre

y cuando cumplan con la condición. En caso contrario se deberá imprimir el siguiente

mensaje “Los números no son pares, o uno de ellos no es par”.

## Ejercicio 64

La empresa “Te llevo a todos lados” está destinada al alquiler de autos y tiene un sistema

de tarifa que consiste en cobrar el alquiler por hora. Si el cliente devuelve el auto dentro

de las 2 horas de uso el valor que corresponde pagar es de $400 pesos y la nafta va de

regalo. Cuando el cliente regresa a la empresa pasadas las 2 horas, se ingresan la

cantidad de litros de nafta gastados y el tiempo transcurrido en horas. Luego, se le cobra

40 pesos por litro de nafta gastado, y la hora se fracciona en minutos, cobrando un total

de $5,20 el minuto de uso. Realice un programa que permita registrar esa información y el

total a pagar por el cliente.

## Ejercicio 65

Construir un programa que simule un menú de opciones para realizar las cuatro

operaciones aritméticas básicas (suma, resta, multiplicación y división) con dos valores

numéricos enteros. El usuario, además, debe especificar la operación con el primer

carácter de la operación que desea realizar: ‘S' o ‘s’ para la suma, ‘R’ o ‘r’ para la resta, ‘M’

o ‘m’ para la multiplicación y ‘D’ o ‘d’ para la división.

## Ejercicio 66

Realizar un programa que, dado un número entero, visualice en pantalla si es par o impar.

En caso de que el valor ingresado sea 0, se debe mostrar “el número no es par ni impar”.

## Ejercicio 67

Escriba un programa para obtener el grado de eficiencia de un operario de una fábrica de

tornillos, de acuerdo a las siguientes dos condiciones que se le imponen para un período

de prueba:

•Producir menos de 200 tornillos defectuosos.

•Producir más de 10000 tornillos sin defectos.

•El grado de eficiencia se determina de la siguiente manera:

•Si no cumple ninguna de las condiciones, grado 5.

•Si sólo cumple la primera condición, grado 6.

•Si sólo cumple la segunda condición, grado 7.

•Si cumple las dos condiciones, grado 8

Nota: para trabajar este ejercicio de manera prolija, ir probando cada inciso que pide el

ejercicio. No hacer todos al mismo tiempo y después probar.

## Ejercicios 68

Solicitar al usuario que ingrese un valor entre 1 y 7. EL programa debe mostrar por pantalla

un mensaje que indique a qué día de la semana corresponde. Considere que el número 1

corresponde al día “Lunes”, y así sucesivamente.

## Ejercicio 69

Leer tres números que denoten una fecha (día, mes, año) y comprobar que sea una fecha

válida. Si la fecha no es válida escribir un mensaje de error por pantalla. Si la fecha es

válida se debe imprimir la fecha cambiando el número que representa el mes por su

nombre. Por ejemplo: si se introduce 1 2 2006, se deberá imprimir “1 de febrero de 2006”.

## Ejercicio 70

Hacer un algoritmo que lea un número por el teclado y determine si tiene tres dígitos.

## Ejercicio 71

Si se compran menos de cinco llantas el precio es de $3000 cada una, si se compran

entre 5 y 10 el precio es de $2500, y si se compran más de 10 el precio es $2000.

Obtener la cantidad de dinero que una persona tiene que pagar por cada una de las

llantas que compra, y el monto total que tiene que pagar por el total de la compra.

## Ejercicio 72

Realice un programa que, dado un año, nos diga si es bisiesto o no. Un año es bisiesto

bajo las siguientes condiciones: Un año divisible por 4 es bisiesto y no debe ser divisible

por 100. Si un año es divisible por 100 y además es divisible por 400, también resulta

bisiesto.

## Ejercicio 73

Una verdulería ofrece las manzanas con descuento según la siguiente tabla:

Nº DE KILOS COMPRADOS % DESCUENTO

0–2 0%

2.01 – 5 10%

5.01 – 10 15%

10.01 en adelante 20%

Determinar cuánto pagará una persona que compre manzanas en esa verdulería.

## Ejercicio 74

El promedio de los trabajos prácticos de un curso se calcula en base a cuatro notas de las

cuales se elimina la nota menor y se promedian las tres notas más altas. Escriba un

programa que determine cuál es la nota eliminada y el promedio de los trabajos prácticos

de un estudiante.

## Ejercicio 75

Una empresa tiene personal de distintas áreas con distintas condiciones de contratación y

formas de pago. El departamento de contabilidad necesita calcular los sueldos semanales

(lunes a viernes) en base a las 3 modalidades de sueldo:

a) comisión

b) salario fijo + comisión, y

c) salario fijo

a) Para la modalidad salario por comisión se debe ingresar el monto total de las ventas

realizadas en la semana, y el 40% de ese monto total corresponde al salario del

empleado.

b) Para la condición de salario fijo + comisión, se debe ingresar el valor que se paga por

hora, la cantidad de horas trabajadas semanalmente y el monto total de las ventas en

esa semana. En este tipo de contrato las horas extras no están contempladas y se fija

como máximo 40 horas por semana. La comisión por las ventas se calcula como 25%

del valor de venta total.

c) Finalmente, para la modalidad de salario fijo se debe ingresar el valor que se paga por

hora y la cantidad de horas trabajadas en la semana. En el caso de exceder las 40

horas semanales, las horas extras se deben pagar con un extra del 50% del valor de la

hora. Realizar un menú de opciones para poder elegir el tipo de contrato que tiene un

empleado.

## Ejercicio 76

Los tramos impositivos para la declaración de la renta en un determinado país son los siguientes.

|  |  |
| --- | --- |
| RENTA | TIPO IMPOSITIVO |
| Menos de $10.000 | 5% |
| Entre $10.000 y $20.000 | 15% |
| Entre $20.000 y $35.000 | 20% |
| Entre $35.000 y $60.000 | 30% |
| Más de $60.000 | 45% |

Escribir un programa que pregunte al usuario su renta anual y muestre por pantalla el tipo de impositivo que le corresponde

## Ejercicio 77

En una determinada empresa, sus empleados son evaluados al final de cada año. Los puntos que pueden obtener en la evaluación comienzan en 0.0 y pueden ir aumentando, traduciéndose en mejores beneficios. Los puntos que pueden conseguir los empleados pueden ser 0.0, 0.4, 0.6 o más, pero no valores intermedios entre las cifras mencionadas. A continuación se muestra una tabla con los niveles correspondientes a cada puntuación. La cantidad de dinero conseguida en cada nivel es de 2.400€ multiplicada por la puntuación del nivel.

| Nivel | Puntuación |
| --- | --- |
| Inaceptable | 0.0 |
| Aceptable | 0.4 |
| Meritorio | 0.6 o más |

Escribir un programa que lea la puntuación del usuario e indique su nivel de rendimiento, así como la cantidad de dinero que recibirá el usuario.

## Ejercicio 78

Escribir un programa para una empresa que tiene salas de juegos para todas las edades y quiere calcular de forma automática el precio que debe cobrar a sus clientes por entrar. El programa debe preguntar al usuario la edad del cliente y mostrar el precio de la entrada. Si el cliente es menor de 4 años puede entrar gratis, si tiene entre 4 y 18 años debe pagar 5€ y si es mayor de 18 años, 10€.

## Ejercicio79

La pizzería Bella Napoli ofrece pizzas vegetarianas y no vegetarianas a sus clientes. Los ingredientes para cada tipo de pizza aparecen a continuación.

* Ingredientes vegetarianos: Pimiento y tofu.
* Ingredientes no vegetarianos: Peperoni, Jamón y Salmón.

Escribir un programa que pregunte al usuario si quiere una pizza vegetariana o no, y en función de su respuesta le muestre un menú con los ingredientes disponibles para que elija. Solo se puede eligir un ingrediente además de la mozzarella y el tomate que están en todas la pizzas. Al final se debe mostrar por pantalla si la pizza elegida es vegetariana o no y todos los ingredientes que lleva.

# Ejercicios de bucle FOR

Ejercicio

Mostrar por pantalla los números pares comprendidos entre 100 y 500.

Ejercicio 77

Escribir un programa que pida al usuario una palabra y la muestre por pantalla 10 veces.

Ejercicio 78

Escribir un programa que pregunte al usuario su edad y muestre por pantalla todos los años que ha cumplido (desde 1 hasta su edad).

Ejercicio 79

Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla todos los números impares desde 1 hasta ese número separados por coma.

Ejercicio

Confeccionar un programa para calcular el valor de la siguiente suma: 100 + 95 + 90 + ... + 50.

Ejercicio

Se ingresan 50 números enteros. Determinar el promedio de los números pares.

Ejercicio

Se ingresan 100 letras. Determinar la cantidad de vocales ingresadas.

Ejercicio

Confeccionar un programa para calcular la suma de los primeros N números naturales. El valor de N lo solicita por teclado el programa.

Ejercicio

Ingresar N y N Números naturales. Determinar e informar:

a.  La sumatoria de los valores múltiplos de 3.

b.  La cantidad de valores múltiplos de 5.

c.  La sumatoria de los valores que se ingresan en orden par.

Ejercicio

Dadas 20 notas y legajos de alumnos de un curso. Determinar:

a.  Cantidad de alumnos aplazados.

b.  Cuando la nota sea mayor a 7, mostrar el mensaje: “El alumno con legajo xxx ha Promocionado”.

Ejercicio

Dadas las edades y estaturas de 45 alumnos de un curso. Determinar:

a.  Edad promedio.

b.  Estatura promedio.

c.  Cantidad de alumnos mayores de 10 años.

d.  Cantidad de alumnos que miden menos de 1.40 cm.

Ejercicio

Confeccionar un programa que exhiba por pantalla una lista, a dos columnas, con los primeros 15 números impares en la primera y los 15 primeros pares en la segunda, incluyendo los títulos.

Ejercicio

Confeccionar un programa que: solicite el ingreso de un valor entero N < 12 y luego una lista de N números reales sobre la cual debe calcular:

a.  el promedio de los positivos.

b.  el promedio de los negativos.

c.  la cantidad de ceros.

Si el valor ingresado N es mayor que 12 informar 'VALOR EXCEDIDO” y si es menor o igual a 0 informar “CANTIDAD INVALIDA”.

Ejercicio

Ingresar 18 valores de temperatura distintos de cero. Se pide determinar e informar cuantas ternas (tres valores seguidos) de valores positivos y cuantas de negativos hay.

Ejercicio

Calcular el factorial de un número ingresado por teclado. El factorial se calcula como el producto de todos los enteros positivos desde 1 hasta el número. En matemática el factorial se expresa con el símbolo !. Por ejemplo, el factorial de 5 es 120 ya que 5! = 1 x 2 x 3 x 4 x 5 =120

Ejercicio

Realizar un programa que:

a.  Muestre todos los números primos entre 1 y 100. (Un nro. es primo cuando es divisible solamente por 1 y por sí mismo).

b.  Contar y mostrar la cantidad de primos encontrados.

Ejercicio

Confeccionar un programa que determine e informe los números perfectos comprendidos entre 1 y 9000. Un número es perfecto cuando es igual a la suma de sus divisores positivos menores que él.

Ejercicios

Se realizó un concurso de tiro al blanco. Existen 5 participantes y cada uno de ellos efectúa 3 disparos, registrándose las coordenadas X-Y de cada disparo. Determinar:

a.   Cuantos disparos se efectuaron en cada cuadrante por cada participante.

b.   Cuantos disparos se efectuaron en total en el centro.

NOTA: no considere disparos sobre los ejes.

Ejercicio

Se ingresan DNI y la nota promedio de 10 alumnos. Determinar el DNI del alumno de mayor nota promedio (considerar único).

Ejercicio

Ingresar N artículos (códigos) y sus correspondientes precios. Indicar el código del artículo más caro y el precio del más barato.

NOTA: todos los artículos tienen precios distintos.

Ejercicio

Realizar un programa que muestre por pantalla las tablas de multiplicar del 1 al 9 de la siguiente forma:

1  2  3  4  5  6  7  8  9

2  4  6  8 10 12 14 16 18

3  6  9 12 15 18 21 24 27

4  8 12 16 20 24 28 32 36

5 10 15 20 25 30 35 40 45

6 12 18 24 30 36 42 48 54

7 14 21 28 35 42 49 56 63

8 16 24 32 40 48 56 64 72

9 18 27 36 45 54 63 72 81

Ejercicio

Ingresar un número entero mayor a 0. Debe dibujar un triángulo rectángulo con \* con tantas filas como el número indicado. En cada fila se va incrementando la cantidad de asteriscos. Por ejemplo si se ingresa el número 6 debe mostrar en pantalla:

 \*

 \*\*

 \*\*\*

 \*\*\*\*

 \*\*\*\*\*

 \*\*\*\*\*\*

Ejercicio

Ingresar un número entero mayor a 0. Debe dibujar una pirámide con \* con tantas filas como el número indicado. En cada fila se va incrementando la cantidad de asteriscos de 2 en 2. Por ejemplo si se ingresa el número 5 debe mostrar en pantalla:

       \*

    \*\*\*

     \*\*\*\*\*

    \*\*\*\*\*\*\*

   \*\*\*\*\*\*\*\*\*

Ejercicio

Tomando como base el ejercicio anterior ingresar un número entero mayor a 0 y dibujar un rombo con \*. Por ejemplo si se ingresa el número 5 dibuja la pirámide de 5 filas y luego completa el rombo al ir decrementando la cantidad de asteriscos.

            \*

          \*\*\*

        \*\*\*\*\*

      \*\*\*\*\*\*\*

    \*\*\*\*\*\*\*\*\*

      \*\*\*\*\*\*\*

        \*\*\*\*\*

         \*\*\*

          \*

Ejercicio 80

Se ingresa un número entero de 3 cifras. Descomponerlo en una unidad, decena y centena.

Ejercicio1

Se ingresa un número entero que representa una fecha con formato “ddmmaa”. Se pide transformarlo a un número con formato “aammdd”.

Ejercicio 82

Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla la cuenta atrás desde ese número hasta cero separados por comas.

Ejercicio 84

Escribir un programa que muestre por pantalla la tabla de multiplicar del 1 al 10.

Ejercicio 85

Escribir un programa que pida al usuario un número entero y muestre por pantalla un triángulo rectángulo como el de más abajo.

1

3 1

5 3 1

7 5 3 1

9 7 5 3 1

Ejercicio 86

Escribir un programa que almacene la cadena de caracteres contraseña en una variable, pregunte al usuario por la contraseña hasta que introduzca la correcta.

Ejercicio 87

Escribir un programa que pida al usuario un núm ero entero y muestre por pantalla si es un número primo o no.

Ejercicio 88

Escribir un programa que pida al usuario una palabra y luego muestre por pantalla una a una las letras de la palabra introducida empezando desde la última.

Ejercicio 89

Escribir un programa en el que se pregunte al usuario por una frase y una letra, y muestre por pantalla el número de veces que aparece la letra en la frase.

Ejercicio 90

Escribir un programa que solicite el ingreso de una palabra y luego imprima la cantidad de letras que tiene con el siguiente formato:

PALABRA tiene n letras.

Ejercicio 91

Escribir un programa que lea un entero positivo, n, introducido por el usuario y después muestre en pantalla la suma de todos los enteros desde 1 hasta n. La suma de los n primeros enteros positivos puede ser calculada de la siguiente forma:

suma: (n(n+1)/2)

Ejercicio 92

Confeccionar un programa que solicite el ingreso de un número entero positivo de 4 cifras y pueda calcular e informar la suma de sus dígitos gasta llegar a una sola cifra. EJEMPLO: 2561 → 5

Ejercicio 96

Para un número N menor de 100. Mostrar la suma de los cuadrados de los números que están separados entre si cuatro posiciones.

Ejercicio 97

Imprimir 10 veces la serie de números de 1 a 10.

Ejercicio 98

Para un número N, imprimir su tabla de multiplicar.

Ejercicio 99

Identificar si la suma de los dígitos de un número es par o impar.

Ejercicio 100

Solicitar un número e imprimir los dígitos pares de este.

Ejercicio 109

Escribir un programa que calcule el cuadrado de los 9 primeros números naturales e

imprima por pantalla el número seguido de su cuadrado. Ejemplo: “2 elevado al cuadrado

es igual a 4”, y así sucesivamente.

Ejercicio 110

Realizar un programa que pida una frase y el programa deberá mostrar la frase con un

espacio entre cada letra. La frase se mostrará así: H o l a. Nota: recordar el funcionamiento

de la función Subcadena().

NOTA:. En PseInt, si queremos escribir sin que haya saltos de línea, al final de la operación

“escribir” escribimos “sin saltar”. Por ejemplo:

Escribir sin saltar “Hola, “

Escribir sin saltar “cómo estás?”

Imprimirá por pantalla: Hola, cómo estás?

Ejercicio 111

Un docente de Programación tiene un listado de 3 notas registradas por cada uno de sus

N estudiantes. La nota final se compone de un trabajo práctico Integrador (35%), una

Exposición (25%) y un Parcial (40%). El docente requiere los siguientes informes claves de

sus estudiantes:

§Nota promedio final de los estudiantes que reprobaron el curso. Un estudiante

reprueba el curso si tiene una nota final inferior a 6.5

§Porcentaje de alumnos que tienen una nota de integrador mayor a 7.5.

§La mayor nota obtenida en las exposiciones.

§Total de estudiantes que obtuvieron en el Parcial entre 4.0 y 7.5.

El programa pedirá la cantidad de alumnos que tiene el docente y en cada alumno pedirá

las 3 notas y calculará todos informes claves que requiere el docente.

Ejercicio 112

Realizar un programa que muestre la cantidad de números que son múltiplos de 2 o de 3

comprendidos entre 1 y 100.

Ejercicio 113

Escribir un programa que calcule la suma de los N primeros números naturales. El valor de

N se leerá por teclado.

Ejercicio 114

Siguiendo el ejercicio 20 de los ejercicios principales, ahora deberemos hacer lo mismo

pero que la cadena se muestre al revés. Por ejemplo, si tenemos la cadena: Hola,

deberemos mostrar a l o H.

Ejercicio 115

Una compañía de seguros tiene contratados a n vendedores. Cada vendedor realiza

múltiples ventas a la semana. La política de pagos de la compañía es que cada vendedor

recibe un sueldo base más un 10% extra por comisiones de sus ventas. El gerente de la

compañía desea saber, por un lado, cuánto dinero deberá pagar en la semana a cada

vendedor por concepto de comisiones de las ventas realizadas, y por otro lado, cuánto

deberá pagar a cada vendedor como sueldo total (sueldo base + comisiones). Para cada

vendedor ingresar cuanto es su sueldo base, cuantas ventas realizó y cuanto cobró por

cada venta.

Ejercicio 116

Realizar un programa que lea un número entero (tamaño del lado) y a partir de él cree un

cuadrado de asteriscos de ese tamaño. Los asteriscos sólo se verán en el borde del

cuadrado, no en el interior. Por ejemplo, si se ingresa el número 4 se debe mostrar:

\*\*\*\*\*\*\*

\* \*

\* \*

\* \*

\* \*

\*\*\*\*\*\*\*

Ejercicio 117

La función factorial se aplica a números enteros positivos. El factorial de un número entero

positivo (!n) es igual al producto de los enteros positivos desde 1 hasta n:

n! = 1 \* 2 \* 3 \* 4 \* 5 \* (n-1) \* n

Escriba un programa que calcule las factoriales de todos los números enteros desde el 1

hasta el 5. El programa deberá mostrar la siguiente salida:

!1 = 1

!2 = 1\*2 = 2

...

!5 = 1\*2\*3\*4\*5 = 120

ejercicio

Escribir un programa que pregunte al usuario una cantidad a invertir, el interés anual y el número de años, y muestre por pantalla el capital obtenido en la inversión cada año que dura la inversión.

Ejercicio

# Ejercicios ciclo WHILE

Ejercicio

Escribir un programa que muestre el eco de todo lo que el usuario introduzca hasta que el usuario escriba “salir” que terminará.

Ejercicio 118

Escriba un programa que valide si una nota está entre 0 y 10, sino está entre 0 y 10 la nota

se pedirá de nuevo hasta que la nota sea correcta.

Ejercicio 119

Escriba un programa en el cual se ingrese un valor límite positivo, y a continuación solicite

números al usuario hasta que la suma de los números introducidos supere el límite inicial.

Ejercicio 120

Dada una secuencia de números ingresados por teclado que finaliza con un –1, por

ejemplo: 5,3,0,2,4,4,0,0,2,3,6,0,……,-1; realizar un programa que calcule el promedio de los

números ingresados. Suponemos que el usuario no insertará número negativos.

Ejercicio 121

Escriba un programa en el cual se ingrese un número y mientras ese número sea mayor

de 10, se pedirá el número de nuevo.

Ejercicio

 Confeccionar un programa que calcule e informe los valores de las potencias de 2 cuyo resultado sean menores que 600. (1 2 4 8 16 …. )

Ejercicio

Escribir las sentencias que permitan el ingreso correcto del valor numérico de un día y un mes determinado. Por algún error volver a solicitar.

Ejercicio

Confeccionar un programa que pueda ingresar una secuencia de caracteres hasta que se digite un \*. Se desea obtener e informar cuantos caracteres ingresados son numéricos y cuantos alfabéticos. Estos últimos tienen que mostrarse separados por mayúsculas y minúsculas. También informar la cantidad de otros símbolos que no sean ni letras ni números.

Ejercicio 122

Escriba un programa que solicite dos números enteros (mínimo y máximo). A continuación,

se debe pedir al usuario que ingrese números enteros situados entre el máximo y mínimo.

Cada vez que un número se encuentre entre ese intervalo, se sumara uno a una variable.

El programa terminará cuando se escriba un número que no pertenezca a ese intervalo, y

al finalizar se debe mostrar por pantalla la cantidad de números ingresados dentro del

intervalo.

Ejercicio 123

Escriba un programa que solicite al usuario números decimales mientras que el usuario

escriba números mayores al primero que se ingresó. Por ejemplo: si el usuario ingresa

como primer número un 3.1, y luego ingresa un 4, el programa debe solicitar un tercer

número. El programa continuará solicitando valores sucesivamente mientras los valores

ingresados sean mayores que 3.1, caso contrario, el programa finaliza.

Ejercicio 124

Calcular las calificaciones de un grupo de alumnos. La nota final de cada alumno se

calcula según el siguiente criterio: la parte práctica vale el 10%; la parte de problemas vale

el 50% y la parte teórica el 40%. El programa leerá el nombre del alumno, las tres notas

obtenidas, mostrará el resultado por pantalla, y a continuación volverá a pedir los datos del

siguiente alumno hasta que el nombre sea una cadena vacía. Las notas deben estar

comprendidas entre 0 y 10, y si no están dentro de ese rango no se imprimirá el promedio

y se mostrará un mensaje de error.

Ejercicio 125

Escribir un programa que calcule cuántos dígitos tiene un número entero positivo sin

convertirlo a cadena (pista: se puede hacer dividiendo varias veces entre 10).

Ejercicio 126

Teniendo en cuenta que la clave es “eureka”, escribir un programa que nos pida ingresar

una clave. Sólo se cuenta con 3 intentos para acertar, si fallamos los 3 intentos se deberá

mostrar un mensaje indicándonos que hemos agotado esos 3 intentos. Si acertamos la

clave se deberá mostrar un mensaje que indique que se ha ingresado al sistema

correctamente.

Ejercicio 127

Escribir un programa que lea números enteros hasta teclear 0 (cero). Al finalizar el

programa se debe mostrar el máximo número ingresado, el mínimo, y el promedio de

todos ellos.

Para poder lograr, por ejemplo, el máximo inicializaremos una variable en cero llamada

numeroMaximo. Luego iremos comparando cada número que se ingresa con esta variable.

Si es mayor reemplazaremos el valor de numeroMaximo. Por ejemplo, si 5 > 0 entonces el

máximo entre estos números será 5. Si luego ingreso el número 2, se evalúa 2 > 5 lo que

resultará falso y por lo tanto el valor 5 de numeroMaximo no se reemplaza. Una lógica

similar tendrá el mínimo.

Ejercicio 128

Realizar un programa que solicite al usuario su código de usuario (un número entero

mayor que cero) y su contraseña numérica (otro número entero positivo). El programa no le

debe permitir continuar hasta que introduzca como código 1024 y como contraseña 4567.

El programa finaliza cuando ingresa los datos correctos.

Ejercicio 129

Se debe realizar un programa que:

1º) Pida por teclado un número (entero positivo).

2º) Pregunte al usuario si desea introducir o no otro número.

3º) Repita los pasos 1º y 2º mientras que el usuario no responda n/N (no).

4º) Muestre por pantalla la suma de los números introducidos por el usuario.

Ejercicio 130

Hacer un algoritmo para calcular la media de los números pares e impares, sólo se

ingresará diez números.

Ejercicio 131

Se pide escribir un programa que calcule la suma de los N primeros números pares. Es

decir, si ingresamos el número 5 como valor de N, el algoritmo nos debe realizar la suma

de los siguientes valores: 2+4+6+8+10.

Ejercicio 132

Programar un juego donde la computadora elige un número al azar entre 1 y 10, y a

continuación el jugador tiene que adivinarlo. La estructura del programa es la siguiente:

1º) El programa elige al azar un número n entre 1 y 10.

2º) El usuario ingresa un número x.

3º) Si x no es el número exacto, el programa indica si n es más grande o más pequeño que

el número ingresado.

4º) Repetimos desde 2) hasta que x sea igual a n.

El programa tiene que imprimir los mensajes adecuados para informarle al usuario qué

hacer y qué pasó hasta que adivine el número.

# CADENAS, STRINGS

Ejercicio 93

Escribir un programa que pregunte el nombre del usuario en la consola y un número entero e imprima por pantalla en líneas distintas el nombre del usuario tantas veces como el número introducido.

Ejercicio 94

Escribir un programa que que pregunte el nombre completo del usuario en la consola y después muestre por pantalla el nombre completo del usuario tres veces, una con todas las letras minúsculas, otro con todas las letras mayúsculas y otra con la primera letra del nombre y de los apellidos en mayúsculas. El usuario puede introducir su nombre combinando mayúsculas y minúsculas como quiera.

Ejercicio 95

Escribir un programa que pregunte el nombre del usuario en la consola y después de que el usuario lo introduzca, muestre por pantalla <NOMBRE> tiene <n> letras, donde <NOMBRE> es el nombre de usuario en mayúsculas y <n> es el número de letras que tiene el nombre.

Ejercicio 101

Los teléfonos de una empresa tienen el siguiente formato prefijo-número-extensión donde el prefijo es el código del país +34, y la extensión tiene dos dígitos (por ejemplo: +34-913724710-56).

Escribir un programa que pregunte por un número de teléfono con este formato y muestre por pantalla el número de teléfono sin el prefijo y la extensión.

Ejercicio 102

Escribir un programa que pida al usuario que introduzca una frase en la consola y muestre por pantalla la frase invertida.

Ejercicio 103

Escribir un programa que pida al usuario que introduzca una frase en la consola y una vocal, y después muestre por pantalla la misma frase pero con la vocal introducida en mayúscula.

Ejercicio 104

Escribir un programa que pregunte el correo electrónico del usuario en la consola y muestre por pantalla otro correo electrónico con el mismo nombre (la parte delantera de la arroba @) pero con dominio ceu.es

Ejercicio 105

Escribir un programa que pregunte por consola el precio de un producto en euros con dos decimales y muestre por pantalla números de euros y número de céntimos del precio introducido.

Ejercicio 106

Escribir un programa que pregunte al usuario la fecha de su nacimiento en formato d/mm/aa y muestra por pantalla, el día, el mes, el año. Adaptar el programa anterior para que también funcione cuando el día o el mes se introduzca con un solo carácter.

Ejercicio 107

Escribir un programa que pregunte por consola por los productos de una cesta de la compra, separados por comas, y muestre por pantalla cada uno de los productos en una línea distinta.

Ejercicio 108

Escribir un programa que pregunte el nombre de un producto, su precio, y un número de unidades y muestre por pantalla una cadena con el nombre del producto seguido de su precio unitario con 6 digitos enteros y 2 decimales, el número de unidades

Ejercicio

Ingresar una frase de hasta 500 caracteres y contar cuántas palabras contiene dicha frase.

 a.  Considerar que las palabras están separadas por un único espacio.

 b.  Considerar que las palabras pueden estar separadas por más de un espacio consecutivo.

Ejercicio

Ingresar dos Strings, uno con el nombre y el otro con el apellido de una persona. Utilizando las funciones de la biblioteca string.h generar un tercer string formado por: NOMBRE, APELLIDO (nombre coma espacio y apellido) y mostrarlo.

Ejercicio

Ingresar nombre y DNI de los alumnos de un curso. Como máximo el curso puede tener 50 alumnos. La carga finaliza con un alumno de nombre FIN. Luego de cargar los alumnos se pide:

 a.  Ingresar nombres de a uno en uno y buscarlos. Si el nombre está en el curso mostrar su DNI y sino informar que no está. Seguir ingresando nombres hasta que se ingrese un nombre igual a NOBUSCARMAS.

 b.  Mostrar el listado de alumnos ordenado alfabéticamente de menor a mayor.

Ejercicio

Se ingresan código y precio unitario de los productos que vende un negocio. No se sabe la cantidad exacta de productos, pero sí se sabe que son menos de 50. El código es alfanumérico de 3 caracteres y la carga de los datos de productos termina con un código igual al “FIN”. Luego se registran las ventas del día y por cada venta se ingresa el código de producto y cantidad de unidades vendidas terminando con una cantidad igual a 0. Se solicita:

 a.  Calcular la recaudación total del día y el producto del cual se vendió menor cantidad de unidades.  
 b.  Mostrar el listado de productos con su precio ordenado en forma alfabética por código de producto.

Ejercicio

Una empresa de alquiler de autos tiene una flota de 30 autos de alta gama, identificados por su número de patente, cargado en la memoria principal en un vector de 30 posiciones. Al comenzar el procesamiento de los alquileres, se ingresa la fecha y la cotización del dólar de ese día. A continuación, se ingresan los siguientes datos correspondiente a cada alquiler realizado en el día:

 •  Patente del auto (alfanumérico, de 6 caracteres)  
 •  Cantidad de días de alquiler (entero, mayor que 0)  
 •  Precio diario del alquiles en dólares (real, mayor que 0)

Para finalizar la carga del día, se ingresa una patente de auto igual a “FINDIA”  
Determinar e informar:

 a.  El porcentaje de autos alquilados durante el día.  
 b.  Realizar el informe con el formato siguiente:

ALQUILER DE AUTOS DEL DIA: DD-MM-AAAA  
COTIZACION DEL DOLAR: $ XX,XX

NRO DE AUTO    DIAS DE ALQUILER      PRECIO DE ALQUILER    FECHA DEDEVOLUCION DL AUTO  
    XXXXX                 XX                        XXXX,XX;               XX/XX/XX  
    XXXXX                 XX                       XXXX,XX;               XX/XX/XX  
    TOTAL RECAUDADO X ALQUILER (EN PESOS)           $XXXX,XX   
    TOTAL RECUADADO X ALQUILER (EN DOLARES)        U$SXXXX,XX 

Ejercicio

Realizar el juego del ahorcado. Primero se debe ingresar la palabra a adivinar de hasta 10 caracteres. Luego se muestra por cada letra un guion bajo para que el jugador sepa la cantidad de letras a adivinar. S e irá ingresando una a una las letras y si estas se encuentran en la palabra las deberá ir mostrando en el lugar correspondiente. Por cada letra que no se encuentre en la palabra perderá una vida. El jugador dispondrá de 5 vidas para intentar ganar el juego.

Complemento:

• Ir completando el dibujo del muñeco del ahorcado cada vez que se comete un error.

# Integrador 1

Necesitamos crear un sistema para una máquina de reciclaje de botellas automática. Dicha máquina nos pagará dinero por la cantidad de plástico reciclado. Tenemos que ingresar nuestro usuario y contraseña para que se nos cargue el saldo por sistema a nuestra cuenta.

• Condición simple anidada: validaremos que el usuario sea “Albus\_D”, luego si esto es verdadero, validaremos si la contraseña es “caramelosDeLimon”. Si la contraseña es correcta haremos que una variable llamada Login sea verdadera.

• Bucle Mientras: Este bloque de validación de la contraseña lo encerraremos en un bucle Mientras para darle al usuario sólo 3 intentos para poner la contraseña.

• Bucle Hacer Mientras(Repetir): Una vez que el login sea verdadero, accederemos al menú de opciones:

o Ingresar botellas

o Consultar saldo

o Salir

• Ingresar Botellas: Primero preguntaremos cuántas botellas se va a ingresar al sistema. Una vez que tenemos el número vamos a usar un bucle para, a fin de ir ingresando cada botella. En cada ciclo del bucle se debe generar un número aleatorio entre 100 y 3000 gr, que va a ser el peso de las botellas a reciclar (simulando que el usuario está ingresando botellas en la máquina). Una vez generado, según el peso del material, usaremos un condicional múltiple para asignarle un valor monetario:

o Si es menos de 500 gr, corresponden $50

o Si es entre 501 gr y 1500 gr, corresponden $125

o Si es más de 1501 gr, corresponden $200

Hecho esto, el programa debe informar al usuario por pantalla el valor que se le ofrece. Si el usuario acepta, lo acreditamos a su saldo, sino se debe devolver el material (sólo mostrar en pantalla “Devolviendo material”). Para esto usaremos un condicional doble.

• Consultar saldo: revisaremos el valor monetario que tiene asignada la variable “saldo”.

• Tanto al terminar “Ingresar Botellas” como “Consultar Saldo” debe volver al menú

principal.

# FUNCIONES

Ejercicio  
Realizar un procedimiento que permita intercambiar el valor de dos variables de tipo entero. La variable A, debe terminar con el valor de la variable B.

Ejercicio  
Crear un procedimiento que calcule la temperatura media de un día a partir de la temperatura máxima y mínima. Crear un programa principal, que, utilizando un procedimiento, vaya pidiendo la temperatura máxima y mínima de n días y vaya mostrando la media de cada día. El programa pedirá el número de días que se van a introducir.

Ejercicio  
Realizar un procedimiento que permita realizar la división entre dos números y muestre el cociente y el resto utilizando el método de restas sucesivas. El método de división por restas sucesivas consiste en restar el dividendo con el divisor hasta obtener un resultado menor que el divisor, este resultado es el residuo, y el número de restas realizadas es el cociente. Por ejemplo: 50 / 13:

50 – 13 = 37 una resta realizada

37 – 13 = 24 dos restas realizadas

24 – 13 = 11 tres restas realizadas

dado que 11 es menor que 13, entonces: el residuo es 11 y el cociente es 3.

Ejercicio

Definir una función max() que tome como argumento dos números y devuelva el mayor de ellos. (Es cierto que python tiene una función max() incorporada, pero hacerla nosotros mismos es un muy buen ejercicio.

Ejercicio

Definir una función max\_de\_tres(), que tome tres números como argumentos y devuelva el mayor de ellos.

Ejercicio

Definir una función que calcule la longitud de una lista o una cadena dada. (Es cierto que python tiene la función len() incorporada, pero escribirla por nosotros mismos resulta un muy buen ejercicio.

Ejercicio

Escribir una función que tome un carácter y devuelva True si es una vocal, de lo contrario devuelve False.

Ejercicio

Escribir una función sum() y una función multip() que sumen y multipliquen respectivamente todos los números de una lista. Por ejemplo: sum([1,2,3,4]) debería devolver 10 y multip([1,2,3,4]) debería devolver

Ejercicio

Definir una función inversa() que calcule la inversión de una cadena. Por ejemplo la cadena "estoy probando" debería devolver la cadena "odnaborp yotse"

Ejercicio

Definir una función es\_palindromo() que reconoce palíndromos (es decir, palabras que tienen el mismo aspecto escritas invertidas), ejemplo: es\_palindromo ("radar") tendría que devolver True.

Ejercicio

Definir una función superposicion() que tome dos [listas](https://pythondiario.com/2014/04/listas-en-python.html) y devuelva True si tienen al menos 1 miembro en común o devuelva False de lo contrario. Escribir la función usando el bucle for anidado.

Ejercicio

Definir una función generar\_n\_caracteres() que tome un entero n y devuelva el caracter multiplicado por n. Por ejemplo: generar\_n\_caracteres(5, "x") debería devolver "xxxxx".

Ejercicio

Definir un histograma procedimiento() que tome una lista de números enteros e imprima un histograma en la pantalla. Ejemplo: procedimiento([4, 9, 7]) debería imprimir lo siguiente:

\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

Ejercicio

Escribir una función que muestre por pantalla el saludo ¡Hola amiga! cada vez que se la invoque.

Ejercicio

Escribir una función a la que se le pase una cadena <nombre> y muestre por pantalla el saludo ¡hola <nombre>!.

Ejercicio

Escribir una función que reciba un número entero positivo y devuelva su factorial.

Ejercicio

Escribir una función que calcule el total de una factura tras aplicarle el IVA. La función debe recibir la cantidad sin IVA y el porcentaje de IVA a aplicar, y devolver el total de la factura. Si se invoca la función sin pasarle el porcentaje de IVA, deberá aplicar un 21%.

Ejercicio

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

Ejercicio

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

Ejercicio

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

Ejercicio

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

Ejercicio

Escribir una función que calcule el máximo común divisor de dos números y otra que calcule el mínimo común múltiplo.

Ejercicio

Escribir una función que convierta un número decimal en binario y otra que convierta un número binario en decimal.

Ejercicio

Escribir un programa que reciba una cadena de caracteres y devuelva un diccionario con cada palabra que contiene y su frecuencia. Escribir otra función que reciba el diccionario generado con la función anterior y devuelva una tupla con la palabra más repetida y su frecuencia.

Ejercicio

Realizar un programa que al ingresar solicite el ingreso de dos números enteros y luego muestre por pantalla el siguiente menú (las xx de los números deben ser reemplazadas con los valores correspondientes):

Menú de Opciones  
----------------

Numero 1: xx Numero 2: xx

1) Sumar

2) Restar

3) Multiplicar

4) Dividir

5) Ingresar Nuevos Números

6) Salir

Ingrese su opción:

Al ingresar el número de la opción del 1 al 4, se realiza la operación y muestra el resultado hasta que se presione una tecla. Luego vuelve a mostrar el menú para poder realizar otra operación con los mismos números. La opción 5 pide el ingreso de dos nuevos números de trabajo mientras que la opción 6 sale y cierra el programa. El ingreso de la opción debe estar validado y en caso de ingresarse un número no válido debe solicitarlo nuevamente. Usar una función para mostrar el menú y retornar el valor elegido.

Ejercicio

Se ingresan números enteros comprendidos entre 100 y 2000 (usar función LeerYValidar). Determinar usando la función EstaDentroDelRango:

 a.  Cantidad de números ingresados entre 100 y 500

 b.  Cantidad de números pares ingresados entre 500 y 1200

 c.  Promedio de números ingresados entre 1200 y 2000

El ingreso de datos finaliza cuando se ingresa un número igual a 99.

Para realizar este programa se deben realizar las siguientes funciones:

 •  EstaDentroDelRango: que reciba 3 enteros correspondientes a un número a validar y los límites superior e inferior del rango. La función debe retornar un 1 si el número a validar se encuentra dentro del rango indicado o un 0 si no lo está.

 •  LeerYValidar: que reciba los límites superior e inferior de un rango y retorne un número que se encuentre dentro del mismo. (El ingreso de datos se realiza dentro de la función). Para validar el rango utilizar la función EstaDentroDelRango realizada en el punto anterior.

Ejercicio

Se realizó un concurso de tiro al blanco. Por cada participante se ingresa un número que lo identifica y el resultado de los disparos efectuados. El ingreso finaliza con un número de participante negativo. Cada participante efectúa 5 disparos, registrándose las coordenadas X-Y de cada disparo.

 •  No considere disparos sobre los ejes, pero sí en el centro (si es sobre los ejes las coordenadas deberán volver a ingresarse).

 •  Para determinar el cuadrante utilizar la función CUADRANTE que reciba las dos coordenadas y retorne el cuadrante al cual pertenece (1 a 4) y 0 para indicar un tiro en el centro.

Para calcular el puntaje utilizar la función PUNTAJE que reciba 5 parámetros que representan la cantidad disparos en cada cuadrante y en el centro. La función debe retornar el puntaje obtenido según la siguiente escala:

 •  Cuadrantes 1 y 2: 50 puntos

 •  Cuadrantes 3 y 4: 40 puntos

 •  Centro: 100 puntos

Determinar:

a. El puntaje obtenido por cada participante, detallando cuantos disparos realizó en cada cuadrante.  
b. Mostrar el número del participante ganador y el puntaje obtenido.  
c. Calcular y mostrar la cantidad total de disparos en el centro (de todos los participantes).

Ejercicio

Se ingresa valores positivos entre 1 y 50, finalizando el ingreso con el valor -10, para ello generar una función que valide dicho ingreso. Calcular:

  a. Cuántos números son primos.

  b. Cantidad de números pares.

  c. Promedio de los números impares.

Utilizar las siguientes funciones:

 •  Resto: recibe por parámetro el Dividendo y el Divisor; Retorna el resto

 •  EsPar: recibe por parámetro un número; Retorna 1 si es Par, 0 si es Impar; Invoca a la función Resto

 •  EsPrimo: recibe por parámetro un número; Retorna 1 si es primo, 0 si no es primo; Invoca a la función Resto. (un número es primero si solo es divisible por 1 y por sí mismo)

 •  Promedio: recibe por parámetro la suma y el contador; Retorna el promedio si contador > 0 sino Retorna 0.

Ejercicio

Confeccionar un programa que permita ingresar varias ternas de valores a punto flotante, que finalice al ingresar una terna con los valores ORDENADOS en forma creciente. Para cada terna exhibir la siguiente pantalla:

\*\* OPCIONES POSIBLES \*\*

1 – Mayor valor

2 – Promedio

3 – Suma

4 – Finalizar

\*\* Digite su opción:

Según la opción deseada debe aparecer en pantalla el resultado, sin borrar el menú. No admitir otros valores de opción. Confeccionar y utilizar una función PARA CADA UNA de las opciones y otra para detectar la condición final.

Ejercicio

Confeccionar un programa que permita determinar e informar en cuántos de los números enteros comprendidos entre 10001 y 90009 existen más cifras 3 que 4. Debe existir por lo menos, un 4 para que sea considerado el número como válido. Para resolverlo confeccionar una función llamada CIFRAS, que pueda determinar cuántas cifras “n” existen en un número de 5 cifras.

Ejercicio

 Confeccionar un programa que permita ingresar diversas ternas de valores enteros mayores que 0 y menores que 99. Finalizar el ingreso cuando se informen 3 valores iguales a 96, 97 y 98. Determinar e informar para cada terna, el mayor y el menor de los valores. Confeccionar y utilizar una función con parámetros llamada MayorMenor, que permita calcular el menor o el mayor valor de la terna según un parámetro de configuración.

Ejercicio

Realizar una función que, dado un mes y un año, calcule la cantidad de días de dicho mes. El año se utilizará en el caso de que el mes sea febrero ya que los años bisiestos tienen 29 días en lugar de 28. Un año es bisiesto cuando: (el año es divisible por 4 y NO por 100) o (el año es divisible por 400).

Ejercicio

Realizar una función que reciba 3 números enteros correspondientes al día, mes y año de una fecha y valide si la misma es correcta. En caso de que la fecha es correcta debe retornar un 1 y si es incorrecta debe retornar un 0. Para la validación usar la función del punto 8 que retorna la cantidad de días de un mes.

Ejercicio

Realizar una función que permita sumar 1 día a una fecha válida mostrando la fecha resultante. Si la fecha recibida no es válida informarlo con un mensaje (validar utilizando la función del punto 9).

Ejercicio

Realizar una función que permita sumar n días a una fecha válida mostrando la fecha resultante. Si la fecha recibida no es válida informarlo con un mensaje (validar utilizando la función del punto 9).

Ejercicio

Realizar una función que permita restar 1 día a una fecha válida mostrando la fecha resultante. Si la fecha recibida no es válida informarlo con un mensaje (validar utilizando la función del punto 9).

Ejercicio

Realizar una función que permita restar n días a una fecha válida mostrando la fecha resultante. Si la fecha recibida no es válida informarlo con un mensaje (validar utilizando la función del punto 9).

Ejercicio

Realizar un programa que ingrese la fecha actual y la fecha de nacimiento de una persona y calcule su edad. Para validar las fechas utilizar la función del punto 9. Para el cálculo realizar una función que reciba las dos fechas y retorne la edad de la persona.

Ejercicio  
Escribir un programa que procese una secuencia de caracteres ingresada por teclado y terminada en punto, y luego codifique la palabra o frase ingresada de la siguiente manera: cada vocal se reemplaza por el carácter que se indica en la tabla y el resto de los caracteres (incluyendo a las vocales acentuadas) se mantienen sin cambios.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | e | i | o | u |
| @ | # | $ | % | \* |

Realice un subprograma que reciba una secuencia de caracteres y retorne la codificación correspondiente. Utilice la estructura “según” para la transformación.

Por ejemplo, si el usuario ingresa: Ayer, lunes, salimos a las once y 10.

La salida del programa debería ser: @y#r, l\*n#s, s@l$m%s @ l@s %nc# y 10.

NOTA: investigue el uso de la función concatenar de PSeInt para armar la palabra/frase.

Ejercicio  
Crea un procedimiento “convertirEspaciado”, que reciba como argumento un texto y muestra una cadena con un espacio adicional tras cada letra.

Por ejemplo, “Hola, tú” devolverá “H o l a , t ú “. Crea un programa principal donde se use dicho procedimiento.

Ejercicio  
Realizar un subproceso que reciba una letra y muestre un mensaje si esa letra esta entre las letras “M” y “T”. Recordar que Pseint le da un valor numérico a cada letra a través del Código Ascii, lo que nos deja usar operadores relacionales con letras y cadenas.

Ejercicio  
Crear un programa que dibuje una escalera de números, donde cada línea de números cmience en uno y termine en el número de la línea. Solicitar la altura de la escalera al usuario al comenzar. Ejemplo: si se ingresa el número 3:

1  
12  
123

Ejercicio  
Realice un algoritmo que solicite al usuario una fecha y muestre por pantalla la fecha anterior.  
Para ello se deberá utilizar un procedimiento llamado diaAnterior que reciba una fecha representada a través de tres enteros dia, mes y anio, y retorne la fecha anterior. Puede asumir que dia, mes y anio representan una fecha válida. Realice pruebas de escritorio para los valores dia=5, mes=10, anio=2012 y para dia=1, mes=3, anio=2004.

Ejercicio  
Diseñar un procedimiento que reciba una frase, y el programa remueva todas las vocales repetidas. Al final el procedimiento mostrará la frase final.  
Por ejemplo:

Entrada: “Habia una vez un barco”

Salida: “Habi un vez n brco"

Se marcan en rojo las repetidas sólo para explicar la consigna. Las vocales ‘e’, ‘i’ y ‘o’ quedan al no estar repetidas.

Ejercicio  
Realizar una función que calcule la suma de dos números. En el algoritmo principal le pediremos al usuario los dos números para pasárselos a la función. Después la función calculará la suma y lo devolverá para imprimirlo en el algoritmo.

Ejercicio  
Realizar una función que valide si un número es impar o no. Si es impar la función debe devolver un verdadero, si no es impar debe devolver falso. Nota: la función no debe tener mensajes que digan si es par o no, eso debe pasar en el Algoritmo.

Ejercicio  
Crea una función EsMultiplo que reciba los dos números pasados por el usuario, validando que el primer número múltiplo del segundo y devuelva verdadero si el primer número es múltiplo del segundo, sino es múltiplo que devuelva falso.

Ejercicio  
Realizar un programa que pida al usuario una frase y una letra a buscar en esa frase. La función debe devolver la cantidad de veces que encontró la letra. Nota: recordar el uso de la función Subcadena().  
  
Ejercicio  
Realizar una función que reciba un numero ingresado por el usuario y averigüe si el número es primo o no. Un número es primo cuando es divisible sólo por 1 y por sí mismo, por ejemplo: 2, 3, 5, 7, 11, 13, 17, etc. Nota: recordar el uso del MOD.

Ejercicio  
Realizar una función que calcule y retorne la suma de todos los divisores del número n distintos de n. El valor de n debe ser ingresado por el usuario.

Ejercicio  
Diseñar una función que reciba un numero en forma de cadena y lo devuelva como numero entero. El programa podrá recibir números de hasta 3 dígitos. Nota: no poner números con decimales ni letras. Ejemplo: ingresando “100”(carácter) debe convertirse en 100(entero).

Ejercicio  
Crear una función llamada “Login”, que recibe un nombre de usuario y una contraseña y que devuelve Verdadero si el nombre de usuario es “usuario1” y si la contraseña es “asdasd”. Además, la función calculara el número de intentos que se ha usado para loguearse, tenemos solo 3 intentos, si nos quedamos sin intentos la función devolverá Falso.

Ejercicio  
Los empleados de una fábrica trabajan en dos turnos: Diurno y Nocturno. Se desea calcular el jornal diario de acuerdo con las siguientes reglas:

a) La tarifa de las horas diurnas es de $ 90

b) La tarifa de las horas nocturnas es de $ 125

c) En caso de ser feriado, la tarifa se incrementa en un 10% si el turno es diurno y en

un 15% si el turno es nocturno.

El programa debe solicitar la siguiente información al usuario: el nombre del trabajador, el día de la semana, el turno (diurno o nocturno) y la cantidad de horas trabajadas. Además, debemos preguntarle al usuario si el día de la semana (lunes, martes, miércoles, etc.) era festivo o no, para poder calcular el jornal diario. Utilice una función para realizar el cálculo.

Ejercicio  
Realizar una función que calcule la suma de los dígitos de un número.

Ejemplo: 25 = 2 + 5 = 7

Nota: Para obtener el último número de un digito de 2 cifras o más debemos pensar en el resto de una división entre 10. Recordar el uso de la función Mod y Trunc.

Ejercicio  
Realizar una función que permita obtener el término n de la sucesión de Fibonacci. La sucesión de Fibonacci es la sucesión de los siguientes números:

1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Donde cada uno de los números se calcula sumando los dos anteriores a él. Por ejemplo:

La sucesión del número 2 se calcula sumando (1+1)

Análogamente, la sucesión del número 3 es (1+2),

Y la del 5 es (2+3),

Y así sucesivamente…

La sucesión de Fibonacci se puede formalizar de acuerdo a la siguiente fórmula:

Fibonacci (n) = Fibonacci (n-1) + Fibonacci (n-2) para todo n > 1

Fibonacci (n) = 1 para todo n <= 1

Por lo tanto, si queremos calcular el término “n” debemos escribir una función que reciba como argumento el valor de “n” y que calcule la serie hasta llegar a ese valor.

Para conocer más acerca de la serie de Fibonacci consultar el siguiente link: <https://quantdare.com/numeros-de-fibonacci/>

Ejercicio  
Realizar una función que reciba un numero ingresado por el usuario y averigüe si el número es capicúa o no (Por ejemplo: 12321). Nota: recordar el uso del MOD y el Trunc. No podemos transformar el numero a cadena para realizar el ejercicio.

# VECTORES/LISTAS/TUPLAS

Ejercicio

Realizar un programa que ingrese la fecha actual y la fecha de nacimiento de una persona y calcule su edad. Para validar las fechas utilizar la función del punto 9. Para el cálculo realizar una función que reciba las dos fechas y retorne la edad de la persona.

Ejercicio

Realizar un programa que ingrese la fecha actual y la fecha de nacimiento de una persona y calcule su edad. Para validar las fechas utilizar la función del punto 9. Para el cálculo realizar una función que reciba las dos fechas y retorne la edad de la persona.

Ejercicio  
Escribir un programa que almacene en una lista los siguientes precios, 50, 75, 46, 22, 80, 65, 8, y muestre por pantalla el menor y el mayor de los precios.

Ejercicio

Escribir un programa que almacene los vectores (1,2,3) y (-1,0,2) en dos listas y muestre por pantalla su producto escalar.

Ejercicio

Ejercicio

Escribir un programa que almacene las asignaturas de un curso (por ejemplo Matemáticas, Física, Química, Historia y Lengua) en una lista, pregunte al usuario la nota que ha sacado en cada asignatura, y después las muestre por pantalla con el mensaje En <asignatura> has sacado <nota> donde <asignatura> es cada una des las asignaturas de la lista y <nota> cada una de las correspondientes notas introducidas por el usuario.

Ejercicio

Escribir un programa que pregunte al usuario los números ganadores de la lotería primitiva, los almacene en una lista y los muestre por pantalla ordenados de menor a mayor.

Ejercicio

Escribir un programa que almacene en una lista los números del 1 al 10 y los muestre por pantalla en orden inverso separados por comas.

Ejercicio

Escribir un programa que almacene las asignaturas de un curso (por ejemplo Matemáticas, Física, Química, Historia y Lengua) en una lista, pregunte al usuario la nota que ha sacado en cada asignatura y elimine de la lista las asignaturas aprobadas. Al final el programa debe mostrar por pantalla las asignaturas que el usuario tiene que repetir.

Ejercicio

Escribir una función que reciba una muestra de números en una lista y devuelva su media.

Ejercicio

Escribir una función que reciba una muestra de números en una lista y devuelva otra lista con sus cuadrados.

Ejercicio

Escribir una función que reciba una muestra de números en una lista y devuelva un diccionario con su media, varianza y desviación típica.

Ejercicio

 Ingresar un vector de 10 unidades reales mediante una función llamada CARGA. Luego mediante una segunda función llamada INVERTIR, generar un segundo vector del mismo tamaño con los elementos ubicados en el orden inverso al del vector original. Mostrar el vector invertido con una función llamada MOSTRAR.

Ejercicio

Realizar una función que reciba por parámetro dos vectores enteros de 5 elementos cada uno y genere un vector de 10 posiciones con el contenido de ambos vectores uno a continuación del otro.

Ejercicio

Realizar una función que reciba por parámetro dos vectores enteros de 5 elementos cada uno y genere un vector de 10 posiciones con el contenido de ambos vectores intercalados.

Ejercicio

Construir una función que reciba dos vectores enteros de igual tamaño y retorne un 1 si son iguales o un 0 sino lo son.

Ejercicio

Una empresa debe registrar los pedidos recibidos de cada uno de sus 10 productos a lo largo del día. Por cada pedido se recibe:

 •  Código de producto (de 1 a 10)

 •  Cantidad de unidades solicitadas

Se puede recibir más de un pedido por producto. La carga finaliza cuando se ingresa un producto con código igual a cero.

Al finalizar se debe emitir un listado con código y cantidad de unidades solicitadas de cada producto.

Ejercicio

La función max() del ejercicio 1 (primera parte) y la función max\_de\_tres() del ejercicio 2 (primera parte), solo van a funcionar para 2 o 3 números. Supongamos que tenemos mas de 3 números o no sabemos cuantos números son. Escribir una función max\_in\_list() que tome una lista de números y devuelva el mas grande.

Ejercicio

Definir una tupla con 10 edades de personas.

Imprimir la cantidad de personas con edades superiores a 20.

Puedes variar el ejercicio para que sea el usuario quien ingrese las edades.

Ejercicio

Definir una lista con un conjunto de nombres, imprimir la cantidad de comienzan con la letra a.

También se puede hacer elegir al usuario la letra a buscar.  (Un poco mas emocionante)

Ejercicio

Escribe una función llamada "elimina" que tome una lista y elimine el primer y último elemento de la lista y cree una nueva lista con los elementos que no fueron eliminados.

Luego escribe una función que se llame "media" que tome una lista y devuelva una nueva lista que contenga todos los elementos de la lista anterior menos el primero y el último.

Ejercicio

Escribe una función "ordenada" que tome una lista como parámetro y devuelva True si la lista está ordenada en orden ascendente y devuelva False en caso contrario.

Por ejemplo, ordenada([1, 2, 3]) retorna True y ordenada([b, a]) retorna False.

Ejercicio

A - Escribe una función llamada "duplicado" que tome una lista y devuelva True si tiene algún elemento duplicado. La función no debe modificar la lista.

B - Crear una función que genere una lista de 23 números aleatorios del 1 al 100 y comprobar con la función anterior si [existen elementos duplicados](https://www.pythondiario.com/2016/11/eliminar-elemento-duplicados-de-una.html). (Puedes ver el [módulo random](https://www.pythondiario.com/2013/06/modulo-random-en-python-con-ejemplos.html) como guía)

Ejercicio

Escribe una función llamada "elimina\_duplicados" que tome una lista y devuelva una nueva lista con los elementos únicos de la lista original. No tienen porque estar en el mismo orden.

Ejercicio

Escribe una función llamada "inversa" que busque todas las palabras inversas de una lista.

Ejemplo de palabras inversas: radar, oro, rajar, rallar, salas, somos, etc…

Ejercicio

Escribir una función mas\_larga() que tome una lista de palabras y devuelva la mas larga.

Ejercicio

Escribir una función filtrar\_palabras() que tome una lista de palabras y un entero n, y devuelva las palabras que tengan mas de n caracteres.

Ejercicio

Escriba una función que tome una lista de números y devuelva la suma acumulada, es decir, una nueva lista donde el primer elemento es el mismo, el segundo elemento es la suma del primero con el segundo, el tercer elemento es la suma del resultado anterior con el siguiente elemento y así sucesivamente. Por ejemplo, la suma acumulada de [1,2,3] es [1, 3, 6].

Ejercicio

Para comprobar si una palabra está en una lista se puede utilizar el operador "in", pero sería una búsqueda lenta, ya que busca a través de las palabras en orden.

Debido a que las palabras están en orden alfabético, podemos acelerar las cosas con una búsqueda de bisección (también conocida como búsqueda binaria), que es similar a lo que haces cuando buscas una palabra en el diccionario. Comenzamos por el centro y comprobamos si la palabra que buscamos está antes o después del centro. Si está antes, se busca solo en la primera mitad, si está después se busca en la otra mitad de la lista. Con esto reduciremos el tiempo de búsqueda

Escribir una función llamada "bisect" que tome una lista ordenada y una palabra como objetivo, y nos devuelva el índice en el que se encuentra en la lista, en caso de no aparecer en la lista devuelve "No se encontró la palabra".

Ejercicio

dem ejercicio anterior, pero con código de productos no correlativos de 4 dígitos. Los códigos deben solicitarse al iniciar el programa mediante la función IngresaCodigos. Luego de la carga determinar:

 a.  El / los productos del cual se solicitaron mayor cantidad de unidades.

 b.  El / los productos del cual se solicitaron menos cantidad de unidades.

Ejercicio

Se ingresan DNI, código de sector (1 a 5) y sueldo de los empleados de una empresa. Se sabe que como máximo la empresa tiene 100 empleados. Se desea:

 a.  Cargar los datos de los empleados mediante la función CargaEmpleados. La carga finaliza con un empleado con DNI 99.

 b.  Generar un vector con la cantidad de empleados en cada sector mediante la función CuentaPorSector y mostrarlo en forma de listado.

 c.  Determinar el importe total a pagar de sueldos mediante la función TotalAPagar.

 d.  Volver a mostrar el listado de cantidad de empleados por sector, pero esta vez ordenado de menor a mayor por cantidad de empleados en cada sector.

Ejercicio

Se ingresan DNI y nota de un parcial de los alumnos de un curso. El ingreso de datos finaliza con un DNI negativo. Se sabe que como máximo pueden presentarse a rendir 60 alumnos. Mostrar:

 a.  Listado de alumnos con su correspondiente DNI y la nota obtenida (en forma de listado).

 b.  La máxima nota obtenida y el DNI de los alumnos que la obtuvieron.

Ejercicio

Se debe ingresar los códigos de 15 productos de una empresa (números de 3 dígitos). Luego se ingresan las ventas realizadas durante el día. Por cada venta se ingresa código de vendedor, código de artículo y cantidad. Los vendedores son 5 y están codificados en forma correlativa de 1001 a 1005. Se puede recibir más de una venta de un mismo vendedor y artículo. El ingreso de datos finaliza con código de vendedor igual a 0. Se desea:

 a.  Mostrar un listado ordenado de mayor a menor por cantidad de unidades vendidas.

      CANT. UNIDADES          CODIGO

         XXX                        XXX

         XXX                        XXX

 b.  Indicar el/los vendedores que realizaron menor cantidad de ventas (no de unidades).

Ejercicio

Realizar los siguientes vectores con 10 valores reales. Confeccionar un programa para:

 a.  Generar el vector VA con la información ingresada por teclado

 b.  Generar el vector VB, donde cada elemento es la raíz cuadrada del respectivo elemento de VA. (para el cálculo de la raíz cuadrada utilizar la función sqrt disponible en la biblioteca math.h). Si el número es negativo poner un 0 ya que no se puede calcular la raíz.

 c.  Calcular el vector VC como VA+VB.

 d.  Copiar el vector VC en orden inverso en otro vector llamado VD.

 e.  Hallar el valor máximo del vector VA y en qué posición se halló (el mayor es único).

 f.  Hallar el valor mínimo del vector VB y en qué posiciones se hallaron (el mínimo se pude repetir).

 g.  Dado un valor ingresado por teclado, indicar en qué posición se encontró del vector VC.

 h.  Ordenar el vector VD en forma ascendente (de menor a mayo).

 i.  Generar un vector VE con aquellos valores negativos que se encuentran en el vector VA.

 j.  Mostrar el contenido de todos los vectores.

FUNCIONES:

 I.  Para el ingreso del vector AA, del punto a).

 II.  Para generar el vector CC, del punto c).

 III.  Para realizar la búsqueda del punto g).

 IV.  Para realizar el ordenamiento del punto h).

 V.  Para generar el vector VE del punto i).

 VI.  Para mostrar los valores de todos los vectores del punto j).

Ejercicio

Confeccionar un programa para:

 a.  Generar los vectores necesarios para guardar la información ingresada por teclado. Se ingresa los datos de 300 artículos diferentes. De cada artículo se ingresa los siguientes datos:

  • Código de artículo (int, entre 1 y 500)  
  • Clase de artículo (char, ‘A’ o ‘B’ o ‘C’ o ‘D’)  
  • Número de depósito donde se encuentra el artículo (int, entre 1 y 100)  
  • Stock (int, entre o y 2000)  
  • Punto de reposición (int, entre 500 y 1000)

 b.  Generar e informar el vector VB, que contenga los códigos de artículos de aquellos que hayan quedado con el saldo igual a cero.

 c.  Generar e informar el vector VC, que contengo a los códigos de artículos de aquellos que hayan quedado con saldo por debajo del punto de reposición.

 d.  Generar e informar el vector VD, que contengo los códigos de artículos de aquellos que están en los depósitos entre el 15 y 50.

 e.  Informar el/los códigos de el/los artículos que tienen el stock más alto. (puede haber varios artículos que tienen la misma cantidad de stock).

FUNCIONES:

 I. Para generar los vectores del punto a).  
 II. Para generar el vector VB.  
 III. Para generar el vector VC.  
 IV. Para generar el vector VD.

Ejercicio

Dados los legajos y sueldos de los empleados de una empresa (máximo 100 empleados). Se pide determinar el/los empleados/as de máximo sueldo. El fin de carga de sueldos está dado por un legajo igual a cero.

 a.  Los números de legajo son números correlativos de 1 a 100.

 b.  Realizar el mismo ejercicio considerando que los números de legajos son números NO correlativos de 4 cifras.

Ejercicio

Para una carrera de automóviles, se toman los tiempos de clasificación (real, mayor que 0 en segundos y centésimos de segundo) de como máximo 60 autos de carrera inscriptos para la competencia. Los autos se identifican con números correlativos del 1 al 60, quedando sin información aquellos que no participan de la presente carrera. Cuando ya no hay más tiempos de clasificación para cargar, se ingresa un auto con número negativo. Se solicita determinar:

 a.  El número de auto que clasificó en primer lugar.

 b.  El número de auto que obtuvo el peor tiempo de clasificación.

 c.  La cantidad de autos que superaron el promedio de tiempo de clasificación.

 d.  El o los números de autos que no participan de la carrera actual.

Ejercicio

Una fábrica produce 14 tipos de muebles de oficina con la mano de obra de 25 operarios especializados. Al finalizar el mes se genera una planilla donde cada registro indica:

•  Día hábil del mes de fabricación (entero, de 1 a 22)

•  Tipo de mueble que se fabricó (entero, de 1 a 14)

•  Código del operario que fabricó el mueble (entero, de 1 a 25)

En la última información presentada, se ingresó un día hábil negativo. No hay orden alguno en los datos ingresados y un mismo operario puede hacer más de un mueble (del mismo tipo o no) durante el mes. Determinar e informar:

 a.  La cantidad de cada tipo de mueble fabricado durante el mes.

 b.  El tipo de mueble del cual se ha fabricado la mayor cantidad. (puede haber más de uno)

 c.  El listado ordenado en forma ascendente por cantidad de muebles fabricados por operario (sin importar el tipo), indicando:

      Código de Operario          Cantidad de muebles fabricados

         XXX                               XXX

 d.  El día o días hábiles en los cuales no hubo producción.

Ejercicio

La empresa TODOMODA desea controlar las ventas de sus productos. La empresa vende 200 productos diferentes. La información de los productos se ingresa por teclado. Cada producto contiene:

 •  Código de producto (int, 301 y 900)

 •  Unidades en stock. (int, mayor o igual a 0)

Luego ingresar la información de las ventas diarias también por teclado. Los datos ingresados son (este lote de datos finaliza con un código de producto igual a 0):

 •  Código de producto (int, 301 y 900)  •  Cantidad de unidades vendidas (int, 1 y 150)

Se piden procesar las ventas diarias y luego determinar:

 a.  Informar el stock de cada código de producto, indicando el código de producto y el stock del mismo.

 b.  Informar el código del producto que tiene el stock más alto.

 c.  Informar el código del producto de aquellos que hayan superado el stock promedio.

FUNCIONES:

 I.  Para controlar los datos ingresados por teclado (la función debe controlar un solo dato).

 II.  Para informar el punto a).

Ejercicio

Una empresa textil desea realizar un informe sobre los sueldos de cada sección para lo cual dispone de los siguientes datos:

 •  La fecha del día del proceso, que debe ser ingresada al comienzo del programa y controlarse que sea válida. De no serlo, reingresarla hasta que lo sea.

 •  Los sueldos de cada empleado:

     i. Legajo (entero, mayor a cero)

     ii. Sección (carácter, desde ‘A’ hasta ‘H’)

     iii. Sueldo (real, mayor a cero)

Los datos de cada empleado deben validarse y en caso de que alguno sea incorrecto, debe ignorarse los tres y reingresarse nuevamente. El fin de datos es cuando aparece un legajo igual a cero. Se pide:

 a.  Emitir un informe que indique por sección, el total de sueldos de la misma, ordenado en forma descendente por total de sueldos:

                               Sueldos al: XX/XX/XXXX

                    Sección          Total de sueldos por sección

                       H                               XXX,XX

                       B                               XXX,XX

                      ...                               ...

 b.  El legajo del empleado con mayor sueldo consignando legajo, sección y sueldo.

Ejercicio

En un negocio trabajan 12 vendedores. Cada vez que se realiza una venta durante el día, se emite una factura donde se indican los siguientes datos:

 • Número de Factura (entero, mayor que cero)

 • Código de vendedor (entero)

 • Importe de la venta (real, mayor que cero)

La información termina con un número de Factura igual a cero. Al principio del ingreso, se debe solicitar la fecha del día de procesamiento. Se pide informar con las leyendas aclaratorias:

 • Importe total por facturación por cada vendedor y Cantidad de facturas emitidas por vendedor, ordenando de forma descendente por importe, con el siguiente formato:

Venta del Día: XX/XX/XXXX

                Nro. de Vendedor     Importe vendido     Cant. facturas emitidas

                 XX                    XXXX,XX                          XX

                 XX                    XXXX,XX                          XX

                 ...                   .......                         ....

                 XX                    XXXX,XX                          XX

                TOTAL                 XXXXXXX,XX                        XX

Desarrollar 2 programas:

 a. Con código de vendedor numérico correlativo de 1 a 12.

 b. Con código de vendedor umérico de 3 cifras, no correlativos

Ejercicio

Se procesan los datos de los casi 90 alumnos de un turno de la materia Elementos de Programación. Por cada alumno se ingresan:

 •  DNI (entero, mayor que cero y menor que 99.999.999)

 •  Nota del Parcial 1 (entero, de 0 a 10)

 •  Nota del Parcial 2 (entero, de 0 a 10)

 •  Porcentaje de asistencia (real, mayor o igual a cero)

Para finalizar, se ingresa un DNI igual a cero.

Se pide informar con las leyendas aclaratorias y/o títulos:

 a.  Según los parciales si promocionó, aprobó, reprobó o estuvo ausente (uno o ambos parciales igual a cero).

 b.  Listar los alumnos que no cumplen con la asistencia (mayor o igual a 75%).

 c.  Informar cuántos alumnos que promocionaron NO cumplen en la asistencia.

 d.  Informar los alumnos que sacaron 10 en el parcial 2.

 e.  Informar los alumnos con menor asistencia (puede haber varios).

 f.  Al finalizar, informar el promedio total de notas de cada parcial y de asistencia.

INTEGRADOR

Vamos a programar una calculadora de materiales para construir

Primero leeremos todo el ejercicio y luego dividiremos tareas en el equipo.

El algoritmo principal sólo debe llamar al subPrograma menu()

Cada subPrograma puede descomponerse, si hiciera falta, en otros subProgramas a creatividad

del programador

El menú debe quedar de la siguiente manera:

1. Calcular muro de ladrillo

2. Calcular viga de hormigón

3. Calcular columnas de hormigón

4. Calcular contrapisos

5. Calcular techo

6. Calcular pisos

7. Calcular pintura

8. Calcular iluminación

9. Salir

subprogramas calcularSuperficie y calcularVolumen

Haremos ambos para usarlos dentro de los otros subprogramas. El usuario no puede acceder a

ellos.

2

subprograma calcularMuro

Nos debe pedir primero si el muro será de 20 o 30 cm de espesor. Luego el largo y el alto. A partir

de estos datos se debe mostrar al usuario la superficie del muro y la cantidad de materiales que

necesitaremos para construirlo.

Si el muro es de 30cm necesitaremos por metro cuadrado: 15.2 kg de cemento, 0.115 m3 de arena

y 120 ladrillos.

Si el muro es de 20cm necesitaremos por metro cuadrado: 10.9 kg de cemento, 0.09 m3 de arena

y 90 ladrillos.

subprograma calcularViga

Nos debe pedir el largo de la viga. Por metro lineal de viga se necesitarán: 9 kg de cemento, 0.02

m3 de arena, 0.02 m2 de piedra, 4 m de hierro del 8 y 3 m de hierro del 4.

Debemos mostrar al usuario la cantidad de materiales necesaria.

subprograma calcularColumna

Nos debe pedir el largo de la columna. Por metro lineal de columna se necesitarán: 7.5 kg de

cemento, 0.016 m3 de arena, 0.016 m2 de piedra, 6 m de hierro del 10 y 3 m de hierro del 4.

Debemos mostrar al usuario la cantidad de materiales necesaria.

subprograma calcularContrapisos

Nos debe pedir espesor, ancho y largo del contrapiso a calcular.

Por metro cúbico de contrapiso se necesita: 105 kg de cemento, 0.45 m3 de arena y 0.9 m3 de

piedra.

Debemos mostrar al usuario la cantidad de materiales necesaria.

subprograma calcularTecho

Nos debe pedir espesor, ancho y largo del techo a calcular.

Por metro cuadrado de techo se necesita: 33 kg de cemento, 0.072 m3 de arena, 0.072 m3 de

piedra, 7 m de hierro del 8 y 4 m de hierro del 6

Debemos mostrar al usuario la cantidad de materiales necesaria.

subprograma calcularPisos

Nos debe pedir ancho y largo del paño de piso a colocar. Teniendo esos datos se debe calcular la

superficie y añadirle un 10% extra por recortes

Mostrar el resultado en m2

subprograma calcularPintura

Nos debe pedir la superficie del muro y mostrar cuánta pintura necesitamos teniendo en cuenta

que rinde 6 m2 por litro de pintura.

subprograma calcularIluminacion

Nos debe pedir la superficie de la habitación. La iluminación la calculamos de la siguiente forma:

superficie \* 0.20. Eso nos da la cantidad mínima de superficie de iluminación natural (ventanas y

puertas de vidrio). Mostrar resultado

# MATRICES

Ejercicio

Ingresar una matriz de 3x3 por teclado y un número entero. Realizar el producto de la matriz por la constante.

Ejercicio

Ingresar una matriz de 2x3 por teclado. Calcular la matriz transpuesta.

Ejercicio

Ingresar una matriz de 5x4 por teclado. Realizar una función que genere un vector con la suma por fila de dicha matriz.

Ejercicio

Ingresar una matriz de 5x4 por teclado. Realizar una función que genere un vector con la suma por columna de dicha matriz.

Ejercicio

Escribir un programa que almacene las matrices

A=(123456)yB=(−100111)

en una lista y muestre por pantalla su producto.

Nota: Para representar matrices mediante listas usar listas anidadas, representando cada vector fila en una lista.

Ejercicio

Ejercicio

Se desea desarrollar un sistema de reservas de entradas para un cine. La sala consta de 12 filas numeradas de la 1 a la 12 y cada fila tiene 9 butacas numeradas a partir de la columna central, con las butacas impares a la derecha y las pares a la izquierda, como en el siguiente esquema:

8 6 4 2 1 3 5 7 9

Para la carga, se debe mostrar al usuario un esquema con las butacas disponibles y reservadas, marcando con la letra D las disponibles y con la letra R las reservadas.

Por cada reserva se debe solicitar la fila y número de butaca a reservar. Cada vez que se realice una reserva se deberá actualizar el esquema que muestra las butacas. Si la butaca seleccionada ya estaba ocupada se debe informar al usuario para que seleccione otra. El proceso de reserva finaliza con una fila con un número negativo. .

Al finalizar mostrar:

 a.  la cantidad de asientos disponibles y la cantidad de asientos reservados.

 b.  los números de filas que quedaron vacías.

 c.  la o las filas con mayor cantidad de espectadores.

 d.  un listado con la cantidad de personas que se sentaron en los mismos números de butacas en todo el cine, ordenado de mayor a menor. Por ejemplo:

Butaca  Cantidad

1     20

3     15

2     10

..............

Ejercicio

Se solicita modificar el programa anterior para que además de la butaca y fila ingrese la cantidad de butacas de la reserva. En dicho caso se deberá verificar si existen a partir de la butaca ingresada la cantidad suficiente de butacas disponibles consecutivas (tomar siempre hacia la derecha del seleccionado). Si están disponibles reservar todas las butacas, sino mostrar un mensaje de error.

Ejercicio

Una empresa desea registrar las ventas realizadas de cada uno de sus 20 productos a lo lardo del año. Los productos están identificados por un código numérico correlativo del 1 al 20. Como primera información se ingresa el precio unitario de cada uno de los productos. Luego por cada venta se ingresa:

 •  Código de producto

 •  Día de venta

 •  Mes de venta

 •  Cantidad de unidades vendidas

El ingreso de datos finaliza con un código de producto igual a 0. Informar:

 a.  El detalle de unidades vendidas de cada producto en cada uno de los 12 meses.

 b.  El producto con el cual se obtuvo la mayor recaudación.

 c.  El trimestre de menor recaudación del año.

Ejercicio

Un consorcio desea controlar la cobranza de las expensas en un edificio de 5 pisos donde existen 15 departamentos en cada piso. El valor de las expensas es de $ 1000. Para registrar la cobranza se ingresa el número de piso (1 a 5) y el número de departamento (correlativo de 1 a 15 en cada piso). El ingreso finaliza con un piso igual a 99.

Al finalizar mostrar el dinero total recaudado y una tabla con una X indicando aquellos departamentos deudores de la siguiente manera:

TOTAL RECAUDADO

DEPARTAMENTO DEUDORES

                              PISO        DPTO1       DPTO2     DPTO15

                              PISO 1        X

                              PISO 2                   X             X

                              PISO 3        X          X

                              PISO 4                                 X

                              PISO 5        X          X

Ejercicio

Se desea contabilizar los votos recibidos en las elecciones de un club de fútbol. Existen 10 candidatos cada uno representado por un número de lista diferente (número de 3 cifras no correlativo). La votación se realiza en 15 sedes distintas codificadas del 1 al 15. Se ingresan los votos registrados en cada una de las sedes registrando:

 •  Número de lista

 •  Número de sede

 •  Cantidad de votos

(Solo se informan candidatos que recibieron votos).

La carga de votos finaliza con un número de lista igual a 0

Mostrar:

 a.  Cantidad de votos recibidos por cada candidato en cada sede.

                      LISTA        SEDE1        SEDE2       SEDE3     SEDE15

                       873          36          78           99...         XX

                       735          XX          XX           XXX...        XX

 b.  Listado ordenado por cantidad de votos totales en formar decreciente, con el siguiente formato:

                              TOTAL DE VOTOS        PORCENTAJE       LISTA

                               800                   80%              873

                               200                   20%              735

c .  Candidatos que NO recibieron votos en la sede 5.

Ejercicio

 Ingresar por teclado valores enteros y generar una matriz de 5x3 (5 filas y 3 por columnas). Informar:

 a.  La suma de los elementos de las columnas.

 b.  Las sumas de los elementos de las filas.

 c.  La suma total de los elementos.

 d.  El valor promedio de toda la matriz.

 e.  El valor promedio por columna.

 f.  El máximo valor de la matriz y en qué posición se encuentra (el máximo es único).

 g.  El mínimo valor de cada columna y en qué posición se encuentra (el mínimo es único)

Utilizar funciones para la carga y para resolver cada uno de los puntos.

Ejercicio

 La cadena de supermercados NORTE, tiene 40 sucursales y comercializa 120 rubros de distintas mercaderías. La Casa Central recibe a diario las notas de pedido de mercaderías emitidas por cada una de las sucursales. Se ingresa por teclado las unidades pedidas (estos datos se ingresan hasta que aparece una sucursal igual a 99, los datos vienen desordenados, y puede venir más de una información de una misma sucursal y el mismo rubro):

 •  Número de sucursal (1 a 40)

 •  Código de rubro (1 a 120)

 •  Cantidad de unidades pedidas.

Informar:

 a.  La cantidad de pedidos por sucursal.

 b.  Cuál fue la sucursal que pidió la menor cantidad en total (puede haber varias sucursales que tienen el mismo valor mínimo).

FUNCIONES:

 I. Para inicializar la matriz.

 II. Para mostrar el resultado del punto b.

Ejercicio

La empresa de turismo GUADALAJARA VIAJES comercializa 10 diferentes tours a través de 15 agencias en todo el país. Diariamente cada agencia envía la cantidad de ventas efectuadas de cada tour, indicando agencia, tours y la cantidad de pasajes. Algunas agencias envían o en la cantidad cuando no existen pasajes para un tour determinado, pero no en todas. La información que se envía tiene los siguientes datos que se ingresan en forma desordenada:

 •  Código de agencia (de 1 a 15)

 •  Código de tour (1 a 10)

 •  Cantidad de pasajes (0 a 20)

Informar:

a. El total de pasajes vendidos por tour.

b. El número de agencia que obtuvo la máxima cantidad de pasajes vendidos.

FUNCIONES:

 I. Para inicializar la matriz.

 II. Para mostrar el resultado del punto a.

Ejercicio

Realizar el juego de Tatetí para dos jugadores. EL jugador 1 utilizará la letra X y el jugador 2 la letra O para marcar sus elecciones. El tablero armarlo con una matriz de 3x3 que debe ser visualizada en pantalla en cada jugada marcando la ubicación elegida por cada jugador. Luego de cada jugada se debe llamar a una función para verificar si el jugador ganó o no la partida. Si se llega a la última jugada y nadie gana, se debe informar del empate.

Complemento: Modificar el juego para que sea de un solo jugador. Deberá programar la inteligencia artificial para que la computadora elija donde realizar la jugada para intentar ganar la partida y no perder completando el espacio correspondiente cuando haga falta.

Ejercicio

Realizar el juego del rompecabezas de 8, dicho juego comienza con una matriz de 3x3 de números del 1 al 8 desordenados y un espacio en blanco. Como por ejemplo:

|  |  |  |
| --- | --- | --- |
| 2 | 5 | 1 |
| 7 | 6 | 3 |
| 8 |  | 4 |

El objetivo del juego es lograr que los números queden ordenados de la siguiente forma:

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 |  |

Para ello, el jugador puede mover un número a la vez al casillero libre, solo puedo mover alguno de los números adyacentes al casillero en blando y solo en forma horizontal o vertical. Por ejemplo tomando como partida la primera matriz desordenada se podría mover el 6 hacia abajo, el 4 hacia la izquierda o el 8 hacia la derecha.

El jugador deberá ingresar el número que quiere mover y automáticamente moverlo, si es posible, al casillero libre. Luego cada movimiento se debe controlar si se llega al objetivo de ordenar los números e informar que se ha ganado la partida. Puede ir contando la cantidad de movimientos que se llevó al jugador lograr llegar a ordenar el rompecabezas y mostrarlos al finalizar para que pueda volver a jugarlo y superarse.

Complemento: Puede agregar al ejercicio anterior distintos niveles para ir aumentando el nivel de complejidad partiendo de distintas matrices desordenadas. Los primeros niveles pueden ser matrices parcialmente desordenadas y luego a medida que avanza de nivel ir jugando con distintas matrices de inicio.

Ejercicio

BATALLA NAVAL: Realizar el juego con un tablero de 5x5. Inicialmente el retador deberá ingresar la ubicación de sus barcos indicando las coordenadas donde se encuentra. Los barcos ocupan un solo casillero y se deben ubicar 5 barcos en distintas posiciones. La matriz a utilizar es de char inicializándola con espacio en blanco y marcando la ubicación de los barcos con la letra B. Una vez ubicado los barcos se muestra al jugador una pantalla con las coordenadas para que pueda realizar su disparo. En caso de no encontrar un barco en la posición indicada se debe poner en la celda de la matriz una letra A (agua). Si se encuentra un barco se marcará como hundido con la letra X. Cada vez que se realiza un disparo se debe actualizar la matriz y mostrarla en pantalla pero sin mostrar la ubicación de los barcos (letra B). Se debe mostrar si el barco fue hundido con la letra X. Contar la cantidad de disparos necesarios para hundir los 5 barcos de forma que pueda desafiar a la otra persona a resolverlo en menos disparos.

Complementos:

 a.  Mejorar el juego al no permitir ubicar barcos en casilleros adyacentes teniendo en cuenta que al hundir un barco se deberán marcar todos los casilleros como agua (letra A).

 b.  Agregar la posibilidad de poner barcos de distinto tamaño, por ejemplo, 2 barcos de 2 casilleros y 3 de un casillero, lo que implica que un barco estará hundido solo cuando se alcancen con un disparo en todas sus partes.

Ejercicio

Realizar el juego de 3 líneas con un tablero de 6 filas por 10 columnas. El juego será para dos jugadores, uno utilizará fichas redondas representadas por la letra O y el otro jugador marcará sus fichas con la letra X.

El juego se desarrolla de a un turno por vez donde el jugador elije sobre que columna tirar su ficha (numerarlas del 1 al 10). La ficha debe “caer” hasta la primera posición libre. Cada vez que ponga una ficha en el tablero deberá verificarse si se ha ganado la partida al tener 3 fichas del mismo tipo consecutivas. Si se llena todo el tablero y ningún participante gana, se mostrará “empate”.

 a.  Chequear si se ganó el juego teniendo en cuenta si se forma 3 en línea solo en forma vertical.

 b.  Chequear si se ganó el juego teniendo en cuenta si se forma 3 en línea tanto en forma vertical como horizontal.

 c.  Chequear si se ganó el juego teniendo en cuenta si se forma 3 en cualquier dirección (vertical, horizontal , o diagonal).

# DICCIONARIOS

Ejercicio

Escribir un programa que guarde en una variable el diccionario {'Euro':'€', 'Dollar':'$', 'Yen':'¥'}, pregunte al usuario por una divisa y muestre su símbolo o un mensaje de aviso si la divisa no está en el diccionario.

Ejercicio

Escribir un programa que pregunte al usuario su nombre, edad, dirección y teléfono y lo guarde en un diccionario. Después debe mostrar por pantalla el mensaje <nombre> tiene <edad> años, vive en <dirección> y su número de teléfono es <teléfono>.

Ejercicio

Escribir un programa que guarde en un diccionario los precios de las frutas de la tabla, pregunte al usuario por una fruta, un número de kilos y muestre por pantalla el precio de ese número de kilos de fruta. Si la fruta no está en el diccionario debe mostrar un mensaje informando de ello.

|  |  |
| --- | --- |
| Fruta | Precio |
| Plátano | 1.35 |
| Manzana | 0.80 |
| Pera | 0.85 |
| Naranja | 0.70 |

Ejercicio

Escribir un programa que pregunte una fecha en formato dd/mm/aaaa y muestre por pantalla la misma fecha en formato dd de <mes> de aaaa donde <mes> es el nombre del mes.

Ejercicio

Escribir un programa que almacene el diccionario con los créditos de las asignaturas de un curso {'Matemáticas': 6, 'Física': 4, 'Química': 5} y después muestre por pantalla los créditos de cada asignatura en el formato <asignatura> tiene <créditos> créditos, donde <asignatura> es cada una de las asignaturas del curso, y <créditos> son sus créditos. Al final debe mostrar también el número total de créditos del curso.

Ejercicio

Escribir un programa que cree un diccionario vacío y lo vaya llenado con información sobre una persona (por ejemplo nombre, edad, sexo, teléfono, correo electrónico, etc.) que se le pida al usuario. Cada vez que se añada un nuevo dato debe imprimirse el contenido del diccionario.

Ejercicio

Escribir un programa que cree un diccionario simulando una cesta de la compra. El programa debe preguntar el artículo y su precio y añadir el par al diccionario, hasta que el usuario decida terminar. Después se debe mostrar por pantalla la lista de la compra y el coste total, con el siguiente formato

|  |  |
| --- | --- |
| Lista de la compra |  |
| Artículo 1 | Precio |
| Artículo 2 | Precio |
| Artículo 3 | Precio |
| … | … |
| Total | Coste |

Ejercicio

Escribir un programa que cree un diccionario de traducción español-inglés. El usuario introducirá las palabras en español e inglés separadas por dos puntos, y cada par <palabra>:<traducción> separados por comas. El programa debe crear un diccionario con las palabras y sus traducciones. Después pedirá una frase en español y utilizará el diccionario para traducirla palabra a palabra. Si una palabra no está en el diccionario debe dejarla sin traducir.

Ejercicio

Escribir un programa que gestione las facturas pendientes de cobro de una empresa. Las facturas se almacenarán en un diccionario donde la clave de cada factura será el número de factura y el valor el coste de la factura. El programa debe preguntar al usuario si quiere añadir una nueva factura, pagar una existente o terminar. Si desea añadir una nueva factura se preguntará por el número de factura y su coste y se añadirá al diccionario. Si se desea pagar una factura se preguntará por el número de factura y se eliminará del diccionario. Después de cada operación el programa debe mostrar por pantalla la cantidad cobrada hasta el momento y la cantidad pendiente de cobro.

Ejercicio

Escribir un programa que permita gestionar la base de datos de clientes de una empresa. Los clientes se guardarán en un diccionario en el que la clave de cada cliente será su NIF, y el valor será otro diccionario con los datos del cliente (nombre, dirección, teléfono, correo, preferente), donde preferente tendrá el valor True si se trata de un cliente preferente. El programa debe preguntar al usuario por una opción del siguiente menú: (1) Añadir cliente, (2) Eliminar cliente, (3) Mostrar cliente, (4) Listar todos los clientes, (5) Listar clientes preferentes, (6) Terminar. En función de la opción elegida el programa tendrá que hacer lo siguiente:

1. Preguntar los datos del cliente, crear un diccionario con los datos y añadirlo a la base de datos.
2. Preguntar por el NIF del cliente y eliminar sus datos de la base de datos.
3. Preguntar por el NIF del cliente y mostrar sus datos.
4. Mostrar lista de todos los clientes de la base datos con su NIF y nombre.
5. Mostrar la lista de clientes preferentes de la base de datos con su NIF y nombre.
6. Terminar el programa.

Ejercicio

El directorio de los clientes de una empresa está organizado en una cadena de texto como la de más abajo, donde cada línea contiene la información del nombre, email, teléfono, nif, y el descuento que se le aplica. Las líneas se separan con el carácter de cambio de línea \n y la primera línea contiene los nombres de los campos con la información contenida en el directorio.

"nif;nombre;email;teléfono;descuento\n01234567L;Luis González;luisgonzalez@mail.com;656343576;12.5\n71476342J;Macarena Ramírez;macarena@mail.com;692839321;8\n63823376M;Juan José Martínez;juanjo@mail.com;664888233;5.2\n98376547F;Carmen Sánchez;carmen@mail.com;667677855;15.7"

Escribir un programa que genere un diccionario con la información del directorio, donde cada elemento corresponda a un cliente y tenga por clave su nif y por valor otro diccionario con el resto de la información del cliente. Los diccionarios con la información de cada cliente tendrán como claves los nombres de los campos y como valores la información de cada cliente correspondientes a los campos. Es decir, un diccionario como el siguiente

{'01234567L': {'nombre': 'Luis González', 'email': 'luisgonzalez@mail.com', 'teléfono': '656343576', 'descuento': 12.5}, '71476342J': {'nombre': 'Macarena Ramírez', 'email': 'macarena@mail.com', 'teléfono': '692839321', 'descuento': 8.0}, '63823376M': {'nombre': 'Juan José Martínez', 'email': 'juanjo@mail.com', 'teléfono': '664888233', 'descuento': 5.2}, '98376547F': {'nombre': 'Carmen Sánchez', 'email': 'carmen@mail.com', 'teléfono': '667677855', 'descuento': 15.7}}

# PROGRAMACIÓN FUNCIONAL

Ejercicio

Escribir una función que aplique un descuento a un precio y otra que aplique el IVA a un precio. Escribir una tercera función que reciba un diccionario con los precios y porcentajes de una cesta de la compra, y una de las funciones anteriores, y utilice la función pasada para aplicar los descuentos o el IVA a los productos de la cesta y devolver el precio final de la cesta.

Ejercicio

Escribir una función que simule una calculadora científica que permita calcular el seno, coseno, tangente, exponencial y logaritmo neperiano. La función preguntará al usuario el valor y la función a aplicar, y mostrará por pantalla una tabla con los enteros de 1 al valor introducido y el resultado de aplicar la función a esos enteros.

Ejercicio

Escribir una función que reciba otra función y una lista, y devuelva otra lista con el resultado de aplicar la función dada a cada uno de los elementos de la lista.

Ejercicio

Escribir una función que reciba otra función booleana y una lista, y devuelva otra lista con los elementos de la lista que devuelvan True al aplicarles la función booleana.

Ejercicio

Escribir una función que reciba una frase y devuelva un diccionario con las palabras que contiene y su longitud.

Ejercicio

Escribir una función reciba una lista de notas y devuelva la lista de calificaciones correspondientes a esas notas.

Ejercicio

Escribir una función reciba un diccionario con las asignaturas y las notas de un alumno y devuelva otro diccionario con las asignaturas en mayúsculas y las calificaciones correspondientes a las notas.

Ejercicio

Escribir una función reciba un diccionario con las asignaturas y las notas de un alumno y devuelva otro diccionario con las asignaturas en mayúsculas y las calificaciones correspondientes a las notas aprobadas.

Ejercicio

Escribir una función que calcule el módulo de un vector.

Ejercicio

Una inmobiliaria de una ciudad maneja una lista de inmuebles como la siguiente:

[{'año': 2000, 'metros': 100, 'habitaciones': 3, 'garaje': True, 'zona': 'A'},

{'año': 2012, 'metros': 60, 'habitaciones': 2, 'garaje': True, 'zona': 'B'},

{'año': 1980, 'metros': 120, 'habitaciones': 4, 'garaje': False, 'zona': 'A'},

{'año': 2005, 'metros': 75, 'habitaciones': 3, 'garaje': True, 'zona': 'B'},

{'año': 2015, 'metros': 90, 'habitaciones': 2, 'garaje': False, 'zona': 'A'}]

Construir una función que permita hacer búsqueda de inmuebles en función de un presupuesto dado. La función recibirá como entrada la lista de inmuebles y un precio, y devolverá otra lista con los inmuebles cuyo precio sea menor o igual que el dado. Los inmuebles de la lista que se devuelva deben incorporar un nuevo par a cada diccionario con el precio del inmueble, donde el precio de un inmueble se calcula con las siguiente fórmula en función de la zona:

* Zona A: precio = (metros \* 1000 + habitaciones \* 5000 + garaje \* 15000) \* (1-antiguedad/100)
* Zona B: precio = (metros \* 1000 + habitaciones \* 5000 + garaje \* 15000) \* (1-antiguedad/100) \* 1.5

Ejercicio

Escribir una función que reciba una muestra de números y devuelva los valores atípicos, es decir, los valores cuya puntuación típica sea mayor que 3 o menor que -3. Nota: La puntuación típica de un valor se obtiene restando la media y dividiendo por la desviación típica de la muestra.

Ejercicio

Construir un pequeño programa que convierta números binarios en enteros.

Ejercicio

Escribir un pequeño programa donde:

- Se ingresa el año en curso.

- Se ingresa el nombre y el año de nacimiento de tres personas.

- Se calcula cuántos años cumplirán durante el año en curso.

- Se imprime en pantalla.

Ejercicio

Los números de las claves de dos cajas fuertes están mezcladas en un

número entero llamado clave maestra. Determine ambas claves, la primera

clave se construye con los dígitos impares de la clave maestra y la

segunda con los pares. Ejemplo: Clave Maestra= 12345, clave1=135,

clave2=24.

Ejercicio

Crear una función contar\_vocales(), que reciba una palabra y cuente cuantas letras "a" tiene, cuantas letras "e" tiene y así hasta completar todas las vocales.

Se puede hacer que el usuario sea quien elija la palabra.

Ejercicio

Escriba una función es\_bisiesto() que determine si un año determinado es un año

bisiesto.Un año bisiesto es divisible por 4, pero no por 100. También es divisible por 400

Ejercicio

Diseñar un sistema de puntos para el juego [El reino del dragón](https://www.pythondiario.com/2013/06/juego-en-python-reino-del-dragon-parte-1.html):

Dejo el enlace por si alguien no lo vio.

La idea es la siguiente: mientras el jugador vaya ganando, ira acumulando puntos.

Ejemplo: Si el jugador entra en la primera cueva y gana el tesoro, se le acreditan 100 puntos, entra en la segunda cueva y gana el tesoro, se le acreditan otros 100 puntos. Si el jugador pierde, saldrá en pantalla el total de los puntos que realizo y la opción de empezar de nuevo.

Ejercicio

Escribe un programa que te permita jugar a una versión simplificada del

juego Master Mind. El juego consistirá en adivinar una cadena de números

distintos. Al principio, el programa debe pedir la longitud de la cadena (de 2

a 9 cifras). Después el programa debe ir pidiendo que intentes adivinar la

cadena de números. En cada intento, el programa informará de cuántos números

han sido acertados (el programa considerará que se ha acertado un número si

coincide el valor y la posición).

Dime la longitud de la cadena: 4

Intenta adivinar la cadena: 1234

Con 1234 has adivinado 1 valores. Intenta adivinar

la cadena: 1243

Con 1243 has adivinado 0 valores. Intenta adivinar

la cadena: 1432

Con 1432 has adivinado 2 valores. Intenta adivinar

la cadena: 2431

Con 2431 has adivinado 4 valores.

Felicidades

Ejercicio  
Escribe un programa que pida dos palabras y diga si riman o no. Si coinciden

las tres últimas letras tiene que decir que riman. Si coinciden sólo las dos

últimas tiene que decir que riman un poco y si no, que no riman.

Ejercicio

Has un programa que pida al usuario una cantidad de dolares, una tasa de interés y un numero de años. Muestra por pantalla en cuanto se habrá convertido el capital inicial transcurridos esos años si cada año se aplica la tasa de interés introducida.

Recordar que un capital C dolares a un interés del x por cien durante n años se convierte en C \* (1 + x/100)elevado a n (años). Probar el programa sabiendo que una cantidad de 10000 dolares al 4.5% de interés anual se convierte en 24117.14 dolares al cabo de 20 años.

Ejercicio

Este programa pide primeramente la cantidad total de compras de una persona. Si la cantidad es inferior a $100.00, el programa dirá que el cliente no aplica a la promoción. Pero si la persona ingresa una cantidad en compras igual o superior a $100.00, el programa genera de forma aleatoria un número entero del cero al cinco. Cada número corresponderá a un color diferente de cinco colores de bolas que hay para determinar el descuento que el cliente recibirá como premio. Si la bola aleatoria es color blanco, no hay descuento, pero si es uno de los otros cuatro colores, sí se aplicará un descuento determinado según la tabla que  aparecerá, y ese descuento se aplicará sobre el total de compra que introdujo inicialmente el usuario, de manera que el programa mostrará un nuevo valor a pagar luego de haber aplicado el descuento.

|  |
| --- |
|  |
|  |

Ejercicio

De la galería de productos, el usuario introducirá el código y el número de unidades del producto que desea comprar. El programa determinará el total a pagar, como una factura.

Una variante a este ejercicio que lo haría un poco más complejo sería dar la posibilidad de seguir ingresando diferentes códigos de productos con sus respectivas cantidades, y cuando el usuario desee terminar el cálculo de la factura completa con todas sus compras. Te animas??

|  |
| --- |
|  |
| Ejemplo ejercicio 2 |

Ejercicio

Este programa muestra primero el listado de categorías de películas y pide al usuario que introduzca el código de la categoría de la película y posterior a ello pide que el usuario introduzca el número de días de atraso, y así se muestra al final el total a pagar.

|  |
| --- |
|  |
|  |

# INTEGRADOR

Ejercicio

La empresario Vegeta ,  desea realizar un informe sobre los sueldos de cada empleado,

tenemos registradas las horas trabajadas durante un determinado mes, los datos son:

Los sueldos de cada empleado:

Legajo (1,35).

dia (1,24).

Horas (real < 12).

Los datos de cada empleado deben ser verificados con la función correspondiente, el fin

de datos con legajo = 0.

Confeccionar un programa, diagrama y codificación que permita, a partir de dicha

información:

Emitir un informe que indique, por día, el total de horas trabajadas.

Emitir un informe que indique el total de horas trabajadas por todos los empleados en

este periodo.

Informar ordenado de mayor a menor por sueldo, el legajo y monto abonado de sueldo

(horas  x $15).

El/Los legajo/s del/os empleados que más horas trabajo. Puede haber varios, informar

horas y legajo.

Informar Legajos de empleados que no trabajaron el periodo.

Ejercicio

La empresa  de viajes comerciales,  “AVATAR”  desea realizar un informe sobre los

sueldos de los choferes, tenemos registradas las horas trabajadas durante un determinado

mes, los datos son:

Las Horas Trabajadas por día de cada chofer:

Codigo (1,50).

dia (1,28).

Horas (int  < 16).

Los datos de cada chofer, deben ser verificados con la función correspondiente, el fin de

datos con codigo = 0.

                 La valor hora Código 1 a 15 $65, Código 16 a 32 $ 55 el resto $ 50

Confeccionar un programa, diagrama y codificación que permita, a partir de dicha información,

1. Emitir un informe que muestre el importe que se abona a cada Chofer ordenado de

Mayor a menor.

               LISTADO DE SUELDOS ABONADOS ORDENADO

                 Codigo            SUELDO

                    33                 8.713,25

                    23                 7.470,00

                    12                 5.456,00

                     -

                    XX                 ---------

2. Informar el Importe Total de sueldos abonados en este periodo.

3. El/Los Codigo/s del/os choferes que más horas trabajo. Puede haber varios, informar

horas y legajo. Solo de los códigos 1 a 15.

4. Informar el día que mas horas de trabajo se realizaron en este periodo.(hay un solo

menor).

5. Cuantos choferes no registraron horas en el periodo.

• Usar función validar y buscar.

Ejercicio

El Video Club The Green Book, necesita realizar estadísticas de los pedidos en alquiler

de sus películas, para definir la reposición de los Títulos. Para tal fin nos proporciona la

siguiente información registrada durante el año 2020.

Un Listado con los géneros en Alquiler:

                                       Acción, Comedia, Drama, Thriller, Infantiles, Terror, Documental.

Por otro lado, los datos mes (1 a 12), género y Cantidad alquilada.

El fin de los datos es indicado con mes =0.

Confeccionar un programa, diagrama y codificación que permita, a partir de dicha información,

Informar el Total de Películas Alquiladas en el año 2018.

Informar el Promedio de Películas Alquiladas  por Genero.(no tomar en cuenta si el

mes está en 0 para promediar)

Informar el Total de Películas por Mes.

Ejemplo:

 Listado Total de Películas Alquiladas por Mes Ordenado de Menor  a Meyor

                                                    Mes         Total Alquileres

2              23.44

--                   xx.xxx

Que cantidad de Meses y Géneros no registraron alquiler.

• función validar, Carga Géneros, ordenar y buscar.

Ejercicio

La empresa de transporte “Alcón S.A.” realiza viajes en todo nuestro país, llevando

distintos tipos de mercadería. Posee una flota de 80 camiones. Cinco de sus camiones, con los

números 76 a 80, están equipados especialmente para traslado de mercadería que necesita

frío.  Esta empresa, nos solicita la confección de un programa que realice distintos informes

para su Departamento Ejecutivo.

a) Se nos suministra unas planillas en hojas de oficio, teniendo que hacer la carga de la

misma (fin de dato un camión con número = 0,  donde se registró la siguiente información:

- Número de Camión. (1 a 80)

- Mes de recaudación (1 a 12)

- Importe acumulado en el mes.

- Cantidad de Kilómetros recorridos en el mes.

b) Debemos obtener de los datos suministrados los siguientes informes:

1 Total, recaudado por camión durante todo el año informado.

2 Que camiones no trabajaron (sin recaudación) más de cuatro meses.

3 Camión que más kilómetros realizó durante todo el año.

4 Total, Recaudado por los camiones con equipamiento especial (76 a 80)

5 Mostrar en forma ordenada de menor a mayor los montos recaudados del Camión 22,

             informar mes monto.

Realizar dos funciones:    ordenar, validar Número de Camión y Mes, mostrar el Punto 4.

Ejercicio

La Empresa DENTACOR  S.R.L. que se dedica a la importación y venta productos odontológicos.

Desea controlar las ventas realizadas en uno de sus locales que cuenta con 7 vendedores  y

cuenta también con casi 5000 productos hasta el momento, para ello se cuenta con los

siguientes datos de cada producto (estos datos se ingresan por teclado y finalizan cuando

se ingresa un Cero en Número de Producto):

 Número de Producto (int, entre 3.000 y 11.000).

 Cantidad de unidades en Stock (int, entre 1 y 2000).

 Precio de Producto (float).

Además cuenta con los datos de las ventas que realizan sus 7 vendedores (estos

datos se ingresan por teclado y finalizan cuando se ingresa un Cero en Número de Vendedor):

             Número de Vendedor (int, entre 1 y 7).

             Número de Producto (int, entre 3.000 y 11.000).

             Cantidad de unidades vendidas (int, entre 1 y 100).

NOTA: Rechazar los pedidos de aquellos productos cuyo stock es inferior al pedido.

Determinar e informar:

     a) Informar el Stock de todos los productos luego de procesar las ventas de sus siete

              Vendedores.

     b) Informar las comisiones que van a recibir cada uno de sus vendores luego de haber

              procesado todas las ventas.(el % de comisión hasta 10 Unidades 2%, hasta 50

2,5%,más 3%)

     c) Informar el Producto que más facturo y cual el que menos facturo, luego de haber

              procesado todas las ventas.

   FUNCIONES:

    1. Para cargar en memoria todos los datos que se ingresan de los productos que tiene la

empresa.

    2. Para validar producto y vendedor

Ejercicio

El complejo de cines Sistema Paradiso de la localidad de Isidro Casanova, cuenta con 15 salas

en las que se proyectan hasta 6 funciones diarias. Hacer un programa que realice, informes

sobre la cantidad de espectadores. Para ello, al final del día se ingresan por teclado la cantidad

de espectadores que asistieron en el día. Por cada sala y función se ingresan, no ordenados, los

siguientes datos:

Número de sala (de 1 a 15)

Número de función (de 1 a 6)

Cantidad de espectadores

(Termina carga con sala =0)

Se pide informar:

1) A qué número de sala y a qué número de función asistió la menor cantidad de

espectadores.

2) Cuál es el número de sala al que asistió la mayor cantidad total de espectadores.

3) Las salas a las que no haya asistido ningún espectador en todo el día.

4) La cantidad de espectadores totales por función.

5) El número de sala y la cantidad promedio de espectadores por cada sala ordenados de

mayor a menor por cantidad promedio de espectadores.

El programa debe contar, por lo menos, con las funciones:

a) Para validar número de cine y número de función (una sola función para ambas

validaciones)

b) Para el ordenamiento del punto 5

1) A qué número de sala y a qué número de función asistió la menor cantidad de

espectadores.

2) Cuál es el número de sala al que asistió la mayor cantidad total de espectadores.

3) Las salas a las que no haya asistido ningún espectador en todo el día.

4) La cantidad de espectadores totales por función.

5) El número de sala y la cantidad promedio de espectadores por cada sala ordenados de

mayor a menor por cantidad promedio de espectadores.

El programa debe contar, por lo menos, con las funciones:

a) Para validar número de cine y número de función (una sola función para ambas

validaciones)

b) Para el ordenamiento del punto 5.

Ejercicio

La Empresa   “Charlie Chocolate Factory” de Willy Wonka, posee una planta distribuidora en

Capital Federal,   necesita hacer una estadística de ventas de los distintos productos que ofrece

al mercado.  Se cuenta con la siguiente información  no ordenada. Finaliza  la carga de la misma

con un producto =  0.

Producto  [1-50]

Mes          [1-12]

Ventas     (por unidades)

(se cuenta con un vector YA cargado de 50 elementos que contiene el precio de los productos

llamado p\_venta)

Se pide:   A - Mes que se vendieron más Unidades.

   B - Del mes de Enero que Producto se Vendió más Unidades.

   C - Del mes de Julio que producto recaudo (plata) más.

   D - Mostrar en forma ordenada la cantidad total de  productos vendidos por mes

         durante  todo el año.

                             Mes     Cantidad Vendida

                                7           35000

                               10          32400

                                 3          31300

1 30900

2 29800

Etc.

Funciones:

              1  -  Ingreso de Producto y Mes,  Validar

    2  -  El ordenamiento del punto   D

# ESTRUCTURA DE DATOS

Ejercicio

Una empresa que vende neumáticos desea realizar un programa para actualizar el total de ventas del mes de sus productos. Para ello primeramente se ingresan la información de los productos formados por:

 •  Código (5 caracteres)  •  Precio (real)

 •  Descripción (30 caracteres)

 •  Cantidad de unidades vendidas (al mes anterior, entero)

 •  Importe Total Vendido (al mes anterior, real)

Se sabe que la empresa no vende más de 50 productos. El ingreso de la carga de productos finaliza con un producto con descripción “FIN”.

Luego ingresan las ventas del mes:

 •  Código de Producto (5 caracteres).

 •  Cantidad pedida.

El ingreso de datos de las ventas finaliza con una cantidad igual a 0. Se solicita:

 a.  Actualizar la información de los productos con las ventas realizadas en el mes.

 b.  Al finalizar, mostrar el listado de productos actualizado, informando:

                      DESCRIPCION        CANT.UNIDADE VENDIDAS        IMPORTE TOTAL VENDIDO

                           XXXXX             XXXX                       $XXXXX,XX

Ejercicio

Un laboratorio dispone de una lista con sus 50 medicamentos. De cada uno conoce:

 •  Código del medicamento (entero, de 3 cifras)

 •  Precio (real)

 •  Stock (entero)

Se solicita:

 a.  Declarar un tipo de dato que contenga la información del medicamento.

 b.  Declarar un vector de estructura de tipo de datos creado en a.

 c.  Cargar en un sector de estructuras, los datos referentes a los medicamentos. Función INGRESO.

 d.  Consultar el precio según código del medicamento. Función BUSQUEDA\_MEDI.

 e.  Informar los códigos de los medicamentos cuyo stock es inferior a 10 unidades. Función INFORME.

Ejercicio

Se sabe que como máximo en una comisión de Elementos de Programación hay 80 alumnos. De cada alumno se conoce:

 •  Número de DNI (entero)

 •  A pellido y Nombre (80 caracteres)

 •  Nota1, Nota2 (entero)

 •  Nota Promedio (real, calculado según Nota1 y Nota2)

Se solicita:

 a.  Declarar un tipo de dato que contenga la información del alumno.

 b.  Declarar un vector de estructuras del tipo de dato creado en el punto a.

 c.  Cargar en un vector de estructuras, los datos referentes a los alumnos de la comisión, esta información termina con DNI igual al 0. Función INGRESO.

 d.  Indicar cuántos alumnos aprobaron (ambos parciales con nota >= 4 y cuántos reprobaron la materia. Función RESULTADO.

 e.  Informar los datos de los alumnos de (DNI – Apellido y Nombre – Nota Promedio) de los alumnos. PROMOCIONADOS (ambas notas >= 7). Función INFORME\_PROMO.

Ejercicio

Se ingresan las ventas de un comercio de insumos de computación. Por cada venta se ingresa:

 •  Número de cliente (entero de 4 dígitos no correlativos).

 •  Importe (mayor a cero)

 •  Número de vendedor (entero de 1 a 10)

El ingreso de datos finaliza con un número de cliente 999.

Se sabe que no son más de 100 clientes, la carga de los clientes se debe realizar al inicio del programa con la función CARGA\_CLIENTE () y para cada uno se ingresa:

• Código de cliente (entero de 4 dígitos no correlativos)

• Nombre y Apellido (50 caracteres máximo)

Se solicita:

 a.  Determinar la cantidad de ventas realizadas por cliente.

 b.  La cantidad de ventas realizadas por vendedor.

Informar de manera ordenada por total facturado (modo descendente), el total facturado a cada cliente, informando:

                      CODIGO DE CLIENTE        NOMBRE Y APELLIDO        TOTAL FACTURADO

                          XXXXX               XXXXXXX, XXXXXXX             $XXXXXXXX,XX

Ejercicio

De una persona se conoce los siguientes datos:

 •  Número de Legajo (int, entre 1 y 10.000).

 •  Apellido (string, 15 caracteres).

 •  Nombre (string, 15 caracteres).

 •  Altura (float, mayor de 0).

 •  Sexo (char, F o M).

 •  Día de Nacimiento (int, entre 1 y 31).

 •  Mes de Nacimiento (int, ente 1 y 12).

 •  Año de Nacimiento (int, ente 1950 y 2019).

Se solicita realizar los siguientes procesos:

 a.  Con datos ingresados por teclado: Crear una Estructura de Datos con los datos enunciados anteriormente y llamarla PERSONA.

 b.  Mostrar los datos de la Estructura de Datos del punto a.

 c.  Con datos ingresados por teclado: Modificar la Estructura de Datos del punto a., crear una Estructura de Datos que contenga Nombre y Apellido y otra Estructura de Datos que contenga Día/Mes/Año utilizarla en una nueva Estructurad de Datos llamada PERSONA2.

 d.  Mostrar los datos de la Estructura de Datos del punto c.

 e.  Crear una FUNCION que cargue los datos de la Estructurad de Datos del punto c y que retorne los datos al programa principal (main).

 f.  Con datos ingresados por teclado: Modificar la Estructurad de Datos llamada PERSONA2 del punto c. y agregarle un campo/miembro que contenga los códigos de las Materias Aprobadas (pueden llegar a ser hasta 36 materias) y llamarla PERSONA3.

 g.  Crear una FUNCION que realice el punto f.

 h.  Mostrar los datos de la Lista creada en el punto f.

 i.  Crear una FUNCION que realice el punto h.

# ARCHIVOS

Ejercicio

Escribe una función que lea las palabras de un archivo de texto (texto.txt) y construya una lista donde cada palabra es un elemento de la lista.

Ejercicio

Dado un archivo ejemplo2.txt, escribir un programa que genere un segundo archivo ejemplo2\_sin\_vocales.txt que incluya el mismo contenido que el primer archivo, pero sin ninguna vocal en las palabras. La idea es no tener abiertos los archivos en simultáneo.

Ejercicio

Dado el archivo ejemplo.txt, del ejercicio anterior, generar un archivo ejemplo\_datos.txt que contenga la cantidad de veces que se repite cada palabra del archivo.

Ejercicio

Script en Python que implemente una agenda de contactos haciendo uso de un diccionario. Para el diccionario las llaves serán los nombres de los contactos y como valor estará una tupla que contenga el número telefónico y el correo electrónico. La agenda se guiardará en un archivo de texto del cual se podrán recuperar contactos y en el cual se prodrán agregar los nuevos contactos registrados. Se tendrá un menú con las siguientes opciones:

1) Agregar contacto

2) Mostrar contactos

3) Buscar contacto

0) salir

Ejercicio

Escribir una función que pida un número entero entre 1 y 10 y guarde en un fichero con el nombre tabla-n.txt la tabla de multiplicar de ese número, done n es el número introducido.

Ejercicio

Escribir una función que pida un número entero entre 1 y 10, lea el fichero tabla-n.txt con la tabla de multiplicar de ese número, done n es el número introducido, y la muestre por pantalla. Si el fichero no existe debe mostrar un mensaje por pantalla informando de ello.

Ejercicio

Escribir una función que pida dos números n y m entre 1 y 10, lea el fichero tabla-n.txt con la tabla de multiplicar de ese número, y muestre por pantalla la línea m del fichero. Si el fichero no existe debe mostrar un mensaje por pantalla informando de ello.

Ejercicio

Escribir un programa que acceda a un fichero de internet mediante su url y muestre por pantalla el número de palabras que contiene.

Ejercicio

Escribir un programa que abra el fichero con información sobre el PIB per cápita de los países de la Unión Europea (url:https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/sdg\_08\_10.tsv.gz&unzip=true), pregunte por las iniciales de un país y muestre el PIB per cápita de ese país de todos los años disponibles.

Ejercicio

Escribir un programa para gestionar un listín telefónico con los nombres y los teléfonos de los clientes de una empresa. El programa incorporar funciones crear el fichero con el listín si no existe, para consultar el teléfono de un cliente, añadir el teléfono de un nuevo cliente y eliminar el teléfono de un cliente. El listín debe estar guardado en el fichero de texto listin.txt donde el nombre del cliente y su teléfono deben aparecer separados por comas y cada cliente en una línea distinta.

Ejercicio

El fichero [cotizacion.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/ficheros/cotizacion.csv) contiene las cotizaciones de las empresas del IBEX35 con las siguientes columnas: Nombre (nombre de la empresa), Final (precio de la acción al cierre de bolsa), Máximo (precio máximo de la acción durante la jornada), Mínimo (precio mínimo de la acción durante la jornada), Volumen (Volumen al cierre de bolsa), Efectivo (capitalización al cierre en miles de euros).

1. Construir una función reciba el fichero de cotizaciones y devuelva un diccionario con los datos del fichero por columnas.
2. Construir una función que reciba el diccionario devuelto por la función anterior y cree un fichero en formato csv con el mínimo, el máximo y la media de dada columna.

Ejercicio

El fichero [calificaciones.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/ficheros/calificaciones.csv) contiene las calificaciones de un curso. Durante el curso se realizaron dos exámenes parciales de teoría y un examen de prácticas. Los alumnos que tuvieron menos de 4 en alguno de estos exámenes pudieron repetirlo en la al final del curso (convocatoria ordinaria). Escribir un programa que contenga las siguientes funciones:

1. Una función que reciba el fichero de calificaciones y devuelva una lista de diccionarios, donde cada diccionario contiene la información de los exámenes y la asistencia de un alumno. La lista tiene que estar ordenada por apellidos.
2. Una función que reciba una lista de diccionarios como la que devuelve la función anterior y añada a cada diccionario un nuevo par con la nota final del curso. El peso de cada parcial de teoría en la nota final es de un 30% mientras que el peso del examen de prácticas es de un 40%.
3. Una función que reciba una lista de diccionarios como la que devuelve la función anterior y devuelva dos listas, una con los alumnos aprobados y otra con los alumnos suspensos. Para aprobar el curso, la asistencia tiene que ser mayor o igual que el 75%, la nota de los exámenes parciales y de prácticas mayor o igual que 4 y la nota final mayor o igual que 5.

273- Los datos de los alumnos de una comisión de Computación Transversal son:

• Número de DNI (entero).

• Nombre y Apellido (80 caracteres).

• Nota1, Nota2 (entero).

• Nota Promedio (real, calculado según Nota1 y Nota2)

 a.  Declarar un tipo de dato que contenga la información del alumno.

 b.  Con la información indicada para los alumnos. Grabar los datos en el archivo “ALUMNOS.dat”. Esta información de grabación finaliza con DNI cero.

 c.  Leer los datos del archivo, mediante la Función LECTURA.

Ejercicio

Tomando como base el archivo ALUMNOS.DAT creado en el ejercicio anterior se desea desarrollar un programa que, basado en la información del archivo, genere 3 archivos:

 •  Uno con los alumnos PROMOCIONADOS (ambas notas mayores o igual a 7)

 •  Otro archivo con los alumnos en condición CURSADO (ambas notas mayores o igual a 4 pero no promocionados)

 •  Y un tercer archivo con los REPROBADOS (al menos una nota, menor que 4)

Para comprobar los datos guardados realizar una función que reciba por parámetro el nombre del archivo a mostrar, lo abre y muestre su contenido. Se debe invocar a la función 3 veces para mostrar todos los archivos generados.

Ejercicio

Se desea realizar un programa que permita actualizar la lista de precios de una empresa en forma individual a cada uno de sus productos.

La empresa trabaja con a lo sumo 100 productos que están guardados en el archivo [PRODUCTOS.dat](https://elementosprogramacion.github.io/archivos/10.3/PRODUCTOS.dat) con la siguiente estructura:

 •  Código (entero)

 •  Precio (float)

 •  Descripción (de hasta 50 caracteres)

Para modificar un precio se ingresa por teclado el código del producto y el nuevo precio. La modificación de precios finaliza con un código de producto igual a 0.

Al finalizar:

a. Actualizar el archivo de productos con los nuevos precios

b. Exportar a un archivo con formato de texto (.csv) para que la lista de precios pueda ser visualizada directamente utilizando un programa de planillas de cálculo como por ejemplo el Excel.

276- Realizar un programa que permite actualizar una lista de precios en forma masiva, ingresando un porcentaje de incremento. El archivo original se llama [PRECIOS.dat](https://elementosprogramacion.github.io/archivos/10.4/PRECIOS.dat) y fue generado utilizando la siguiente estructura:

 •  Código (entero)

 •  Precio (float)

 •  Descripción (de hasta 50 caracteres)

No se sabe la cantidad de registros del archivo.

Se solicita generar un nuevo archivo llamado Precios\_actualizados.dat

Ejercicio

Se dispone de un archivo llamado [Stock.dat](https://elementosprogramacion.github.io/archivos/10.5/STOCK.dat) que contiene la información de los 10 productos que vende una fábrica. En el archivo se guarda:

 •  Código de artículo (entero)

 •  Descripción (50 caracteres máximo)

 •  Stock (entero)

Luego se ingresan por teclado las ventas a realizar indicando:

 •  Código de artículo

 •  Cantidad

La carga por teclado de las ventas finaliza con un código de artículo igual a 0.

Por cada venta se debe controlar si hay stock suficiente y si lo hay, restar el stock de dicho producto, sino hay stock se debe vender lo que quede disponible y grabar un registro en un archivo Faltantes.dat con la cantidad que no pudo venderse, dicho registro debe contener:

 •  Código de artículo

 •  Cantidad

Se genera un registro por cada venta que no pudo concretarse, pudiendo quedar en el archivo varios registros faltantes del mismo producto. Al finalizar actualizar el archivo Stock.dat con el stock actualizado luego de procesar las ventas.

Ejercicio

Realizar un programa que tome el archivo Faltantes.dat del ejercicio anterior y unifique las cantidades sumariándolas por artículo (la empresa trabaja con 10 artículos en total, pero no todos pueden estar en falta).

Generar una planilla Excel para enviar al proveedor con el pedido a realizar. Para ello se debe crear un nuevo archivo llamado pedido.csv que contenga los registros por artículo conteniendo:

 •  Código de artículo (entero)

 •  Cantidad (entero)

Ejercicio

Se dispone de un archivo que contiene información de los vuelos realizados por las distintas aerolíneas a lo largo del mes. El archivo se denomina [Vuelos.dat](https://elementosprogramacion.github.io/archivos/10.7/VUELOS.dat) y guarda los registros con la siguiente estructura:

 •  Código Aerolínea (alfanumérico de 10 caracteres máximo)

 •  Día (entero)  •  Número de Vuelo (entero)

 •  Costo del pasaje (real)

 •  Pasajeros (inicialmente en el archivo viene en 0)

El mes tiene 30 días y se sabe que como máximo cada aerolínea realiza 5 vuelos diarios. Luego se dispone de un segundo archivo llamado [Pasajeros.dat](https://elementosprogramacion.github.io/archivos/10.7/PASAJEROS.dat) que incluye la información de los viajeros del mes para la aerolínea con código “Aero1”, el archivo contiene los siguientes campos:

 •  DNI (entero)

 •  Número de Vuelo (entero)

Se desea realizar un programa que actualice la cantidad de pasajeros de la aerolínea con código “Aero1” con la información de los pasajeros que realizar los viajes. Al finalizar, grabar la información en un nuevo archivo llamado Aero1.dat. Agregar la opción para exportar el archivo Aero1 a formato csv

Ejercicio

Se dispone de un archivo denominado [INSCRIPTOS.dat](https://elementosprogramacion.github.io/archivos/10.8/INSCRIPTOS.dat) que contiene la información de los inscriptos a un curso de programación. El curso tiene una capacidad máxima para 60 personas. El archivo aún no está completo ya que la inscripción se realiza por partes. Por cada inscripto se tiene la siguiente información:

 •  DNI (entero)

 •  Apellido y Nombres (texto de 20 caracteres máximo)

 •  Pagado (campo entero donde 1 indica que pagó y 0 que aún adeuda la matrícula)

Se desea realizar un programa que cumpla con dos funciones:

 a.  Registrar los pagos de los alumnos ya inscriptos

 b.  Si hay cupo, agregar nuevos inscriptos

El programa solicitará el ingreso del DNI y lo buscará entre los inscriptos, en caso de que lo encuentre dará la opción para registrar el pago (si es que no está pago ya). En caso de que no lo encuentre dará la opción para inscribirlo al curso (el pago se realiza luego). En el momento que ya no quede cupo en el curso si llega un nuevo inscripto se le preguntará si desea quedar registrado para un curso futuro, y en caso afirmativo se le solicitará el teléfono y se guardará el nombre, DNI y teléfono en un archivo Interesados.dat.

El ingreso de datos finaliza con un DNI negativo. 2 Al finalizar se debe actualizar el archivo de inscriptos con los cambios realizados.

Ejercicio

Una fábrica produce como máximo 20 productos diferentes y necesita un reporte de la producción realizada durante el año. Para ello se dispone del archivo “ARTICULOS.dat”, el cual contiene:

 •  Código de artículo (entero de 4 cifras no correlativo)

 •  Descripción del artículo (30 caracteres máximo)

 •  Cantidad de stock remanente del año anterior (entero)

A continuación, se procesa por teclado, la información correspondiente a todo el año:

 •  Número del mes de producción. Ingresar y controlar mediante la Función LEE\_CONTROL.

 •  Código de artículo (entero de 4 cifras no correlativo) Ingresar y controlar un número de 4 cifras. Función LEE\_CONTROL.

 •  Cantidad de unidades que se produjeron en ese mes. Controlar si el número es mayor que 0. Función MAYOR\_CERO.

Esta información no se encuentra ordenada, y termina con número de mes igual a 0.

Se solicita construir un programa que determine e informe:

 a.  Para cada artículo, la cantidad de unidades producidas en cada uno de los 12 meses del año, de acuerdo con el siguiente modelo:

       CODIGO ARTICULO   DESCRIPCION       ENERO  FEBRERO   ...   DICIEMBRE

         XXX             XXXXXXXXXXXX       XX     XX       ...        XX

 b.  Confeccionar un listado con el/los códigos y descripción de aquellos artículos que NO se produjeron en todo el año.

 c.  Generar el archivo actualizado llamado “ARTI\_ACTUALIZADO.dat” con la suma del stock remanente del año anterior más la cantidad producida en el primer trimestre del presente año.

Ejercicio

La empresa TODOLIMPIO que comercializa un máximo de 25 productos, dispone del archivo “LIMPIEZA.dat”, donde cada registro contiene:

 •  Número de Artículo (entero, 5 cifras)

 •  Unidades en stock (entero)

 •  Precio unitario (real)

Cada vez que se realiza una venta se ingresa por teclado, los siguientes datos:

 •  Número de Cliente (1 a 20) Función LEE\_CONTROL

 •  Número de Artículo (entero, 5 cifras). Función LEE\_CONTROL

 •  Cantidad de Unidades Vendidas. Función MAYORCERO.

Esta información termina con número de cliente igual a 0.

Si la cantidad de unidades vendidas es superior al stock existente de cada uno de los productos, NO realizar la venta y marcarla para un posterior reporte.

Se solicita confeccionar un programa que actualice el archivo “PRODUCTOS.dat” y e informar:

 a.  La facturación total por cliente.

 b.  La cantidad de veces que cada cliente realizó una compra.

 c.  Emitir un listado ordenado por monto de facturación de forma descendente de acuerdo con el siguiente diseño:

       CLIENTE   MONTO FACTURADO       CANTIDAD DE VECES QUE

                                         REALIZO UNA COMPRA

         XXX      XXXXXX,XX                    XX

 d.  Emitir un listado con el número de artículos que no se pudo realizar las ventas por falta de stock.

 e.  Actualizar el archivo “LIMPIEZA.dat”, en UNIDADES en STOCK según las ventas ingresadas.

Ejercicio

La municipalidad de Marina del Mar desea efectuar un control de la cobranza del impuesto Municipal en los 6 bimestres del año. Existen 50 contribuyentes, numerados de 1 a 50.

 •  El primer dato para ingresar es el número del año a procesar (xxxx)

Se suministra el archivo “IMPORTES.dat”, ordenado por año-bimestre, con 18 ítems, que contienen los importes de los impuestos de los últimos 3 años, cada ítem contiene:

 •  Año del impuesto (entero)

 •  Bimestre (entero)

 •  Importe (real)

Se dispone además de un segundo archivo, llamado “CONTROL.dat”, ordenado por número de contribuyente, con un registro por cada cobro efectuado, con los siguientes datos (solo considerar los del año en estudio).

 •  Número de contribuyente (entero)

 • Año del impuesto (entero)

 • Bimestre (entero)

Se solicita confeccionar un programa para determinar e informar:

 a.  El importe total recaudado en el año.

 b.  Las deudas de cada contribuyente según el siguiente formato de impresión:

              DEUDAS REGISTRADAS EN EL AÑO XXXX

         CONTRIBUYENTE               BIMESTRE 1 2 3 4 5 6                  IMPORTE ADEUDADO

             XXXXXX                             X       X                   $ XXXXXXX,XX

Ejercicio

Un Laboratorio de medicamentos quiere actualizar las ventas acumuladas a la fecha, dispone para este trabajo de un archivo llamado “VENTAS.dat” con las ventas acumuladas del año, hasta la fecha de proceso con los siguientes datos:

 •  Número de visitador médico de 1 a 15 (entero)

 •  Código del medicamento de 1 a 30 (entero)

 •  Cantidad de unidades vendidas (entero)

Este archivo está clasificado y ordenado en forma ascendente por número de visitador médico y luego por el código del medicamento. Pueden NO estar todos los códigos de medicamentos ni todos los visitadores médicos “activos”.

La información que se registra por teclado de las ventas del día NO está ordenada y para finalizar se ingresa un número de cliente igual a 0. Los datos disponibles son:

 •  Número de Cliente (entero entre 1 y 50000)

 •  Número de Visitador Médico (entero)

 •  Código de medicamento (entero)

 •  Cantidad vendida (entero)

 •  Importe facturado (float)

Se solicita escribir un programa que permita:

 a.  Leer el archivo “VENTAS.dat” y los cargue en la matriz TOTAL de 15 por 30 elementos.

 b.  Ingresar por teclado, la fecha de proceso (día, mes, año)

 c.  Procesar las ventas del día

 d.  Actualizar los acumulados de la matriz TOTAL y al finalizar actualizar el archivo “VENTAS.dat”

 e.  Imprimir un listado con el siguiente diseño:

                        VENTAS DEL XX/XX/XXXX

         CODIGO DE MEDICAMENTO             IMPORTE DE LA FACTURACION

              XX                            $XXXXX,XX

             ----                            --------

              XX                            $XXXXX,XX

        --------------------------------------------------------------

           TOTAL VENDIDO                    $XXXXX,XX

 f.  Al finalizar el proceso informe el número de Visitador Médico que en el día vendió el mayor importe. (Total Facturado).

Ejercicio

Se desean procesar las transacciones bancarias de las Cuentas Corrientes del BANCO NACIÓN. Se solicita:

a.  Leer los archivos “SALDOS.dat” con el saldo del día anterior de cada cliente y lo almacene en la memoria. El archivo está ordenado por número de cuenta y está formado por registros con el siguiente diseño (máximo 300 clientes).

 •  Número de cuenta (8 dígitos)

 •  Razón Social (28 caracteres)

 •  Crédito máximo (real)

 •  Saldo (real, que puede contener signo -)

 b.  Ingresar la fecha del día de proceso, controlando que sean valores “lógicos”, confeccionando y utilizando una función, llamada LeerValorLogico y utilizarla para el ingreso y validación. (día, mes y Año). Utilizar una estructura.

 c.  También se dispone de otro archivo “MOVIM.dat” con los movimientos del día para actualizar los saldos de las cuentas. Cada registro consta de:

 •  Número de Cuenta

 •  Código de Operación (R: retiro / D: depósito)

 •  Importe (real)

 d.  Si el cliente no se encuentra en el archivo, puede ser una CUENTA NUEVA (Nro. De cuenta mayor al último número existente en el archivo Saldos.dat.) o debido a un error. Esta verificación de búsqueda, realizarla confeccionando y utilizando una Función llamada VeriCuenta. Si hay error se muestra y rechaza la operación, si es cuenta nueva, solo puede ser un depósito, en cuyo caso se solicita por teclado el ingreso de la Razón Social y se lo agrega al final, con el doble del importe como crédito máximo y con el importe como saldo.

 e.  Listar las Cuentas Nuevas, con el siguiente diseño:

                             CUENTAS NUEVAS DEL XX/XX/XXXX;

         NUMERO DE CUENTA             CREDITO MAXIMO               SALDO

             XXXXXX                    $XXXXX,XX                $ XXXXXXX,XX

             XXXXXX                    $XXXXX,XX                $ XXXXXXX,XX

Ejercicio

Se dispone de un archivo “EXPENSAS.dat” con los importes según cada uno de las 5 categorías de un complejo de edificios. Este archivo consta de un (1) solo registro con los 5 importes correspondientes en forma correlativa a la categoría 1, categoría 2 hasta la 5. Además, se dispone del archivo “PROPIETARIO.dat” con los datos de cada propietario:

• DNI (entero)

• Apellido y Nombre (50 caracteres)

• Domicilio (50 caracteres)

Se sabe que existen como MÁXIMO 100 PROPIETARIOS. También se dispone del archivo “PAGOS.dat” con los efectuados por cada propietario que ha pagado la expensa:

 •  DNI (entero)

 •  Mes de pago (entero)

 •  Categoría (entero de 1 a 5)

Se solicita:

 a.  Informar el siguiente listado, con “x” el mes o meses que NO pagó.

         APELLIDO Y NOMBRE   MES 1    MES 2...      MES 12

          XXXXXX, XXXXXX        X      X

          XXXXXX, XXXXXX               X             X

 b.  Total pagado por cada propietario, ordenando en forma ascendente por total pagado.

         APELLIDO Y NOMBRE     TOTAL PAGADO

         XXXXXX, XXXXXX        $ XXX,XX

         XXXXXX, XXXXXX        $ XXX,XX

 c.  Generar el archivo “DEUDAS.dat” con aquellos propietarios que NO han pagado más de 6 meses, con igual diseño que el archivo “PROPIETARIOS.dat”.

 d.  Cuando NO se encuentra el DNI, grabar en el archivo “ERROR.dat” la información contenida en el archivo “PAGOS.dat”.

Ejercicio

Desarrollar un programa que permita:

 a.  Solicitar el ingreso desde el teclado del código de uno de los 3 departamento de la empresa AZ CONSULTORES, confeccionando y utilizando una función con parámetros que controles su validez (los códigos de departamento pueden ser 100, 200 o 300), llamarla LeeCodigoDeparta.

 b.  Leer luego desde un archivo secuencial, llamado “PERSONAL.dat”, los datos que sean necesarios de los empleados que corresponden al departamento seleccionado, reservándolos en memoria y utilizando una adecuada estructura de datos.

El diseño de los registros es:

 •  Departamento (entero)

 •  Legajo (Número, entero NO correlativo de 4 cifras)

 •  Apellido y Nombres (60 caracteres)

 •  Años de Antigüedad (entero)

Cada departamento tiene como máximo 300 empleados.

 c.  Ingresar desde el teclado los sueldos de dichos empleados:

 •  Legajo (Número, entero NO correlativo de 4 cifras)

 •  Sueldo (real)

El ingreso se tiene que finalizar con un número de legajo negativo.

 d.  Controlar la validez del legajo (existencia dentro del Departamento) confeccionando y utilizando una Función llamada ControLega. Si se detecta un legajo inexistente se debe informar Error y solicitarlo nuevamente. Grabar un archivo “ERROR.dat” solamente con el número de legajo.

 e.  Confeccionar un listado con los empleados cuyo sueldo es menor que el promedio general de sueldos del departamento en estudio, informando el sueldo y el nuevo sueldo con el suplemento por antigüedad incluido. El suplemento por antigüedad (sobre el sueldo ingresado en el punto c.) es de un 1’% para menos de 10 años y 15% para 10 o más años.

                             REMUNERACIONES MENORES AL PROMEDIO

          LEGAJO            SUELDO           ANTIGUEDAD          NUEVO SUELDO

          XXXXXX            $XXXXX,XX           XX              $ XXXXXXX,XX

          ------            ---------           --               -----------

          XXXXXX            $XXXXX,XX           XX              $ XXXXXXX,XX

Ejercicio

Se dispone de un archivo [VENTAS.dat](https://elementosprogramacion.github.io/archivos/10.16/VENTAS.dat) con el siguiente formato:

 •  Código de producto (alfanumérico 10 caracteres)

 •  Código de vendedor (entero 1 a 5)

 •  Cantidad (entero).

El archivo puede contener más de una venta para el mismo producto/vendedor. Luego se dispone de un segundo archivo llamado [PRODUCTOS.dat](https://elementosprogramacion.github.io/archivos/10.16/PRODUCTOS.dat) que contiene:

 •  Código de producto (alfanumérico 10 caracteres)

 •  Precio de venta (float)

Como máximo hay 15 productos.

 a.  Se desea mostrar la cantidad de ventas realizada por cada vendedor de cada producto (utilizar una matriz).

          PRODUCTO    VENDEDOR 1    VENDEDOR 2     VENDEDOR 3     VENDEDOR 4     VENDEDOR 5

          PROD 1          9            0              1                 4           1

          PROD 2          19           10             0                 0           1

 b.  Registrar nuevas ventas ingresando por teclado:

 •  Código de producto

 •  Código de vendedor

 •  Cantidad.

(el ingreso por teclado finaliza con vendedor igual a 99)

 c.  Mostrar nuevamente el listado del punto a actualizado con las nuevas ventas.

 d.  Mostrar un listado ordenado de mayor a menor por importe de ventas mostrando código de vendedor e importe total vendido.

 e.  Mostrar el / los productos que vendieron menor cantidad de unidades.

 f.  Actualizar el archivo ventas.dat.

Ejercicio

El archivo [TEMPERATURAS.dat](https://elementosprogramacion.github.io/archivos/10.17/TEMPERATURAS.dat) contiene información de las temperaturas registradas en 5 localidades a lo largo del primer trimestre del año con la siguiente estructura:

 •  Código Localidad (alfanumérico 10 caracteres)

 •  Mes (entero 1 a 12)

 • Temperatura (float)

Por cada localidad pueden existir varias mediciones para un mismo mes, pero no se sabe la cantidad. Además, se dispone de un segundo archivo llamado [LOCALIDADES.dat](https://elementosprogramacion.github.io/archivos/10.17/LOCALIDADES.dat) que contiene 5 registros con la información de las localidades:

 •  Código Localidad (alfanumérico 10 caracteres)

 •  Descripción

Se desea realizar un programa que basado en la información del archivo temperaturas.dat:

 a.  Genere un listado de temperaturas promedio en cada localidad en cada mes con el siguiente formato (para ello usar una matriz), es posible que para una localidad mes no se tengan lecturas de temperaturas por lo tanto deberá mostrarse con un signo de interrogación.

          LOCALIDAD       ENERO    FEBRERO     MARZO     VARIACION TERMICA

          MAR DEL PLATA     23        32         31              8

          CHILECITO         30        35         37              7

          MENDOZA           27        ?          ?               0

          CALAFATE          ?         11         9               2

 b.  Grabar un archivo llamado promedios.dat que guarde por cada localidad y cada mes el promedio de temperaturas donde cada registro contenga: código de localidad, mes y promedio.

 c.  Grabar un archivo llamado variaciones.dat que por cada localidad guarde la variación térmica (temperatura máxima - temperatura mínima). Los registros de este archivo contendrán dos datos código de localidad y el valor de la variación.

Ejercicio

Realizar un programa que leyendo los archivos generados en el ejercicio anterior (promedios.dat, variaciones.dat y localidades.dat) vuelva a mostrar el listado de temperaturas promedio y variación térmica del punto a del ejercicio anterior (matriz).

291- Se dispone del archivo Productos.dat que contiene información acerca de los 100 productos que comercializa una empresa, además, sabe que cada depósito produce como máximo de 10 productos. Cada registro del archivo contiene:

 •  Código de producto (7 caracteres alfanuméricos).

 •  Descripción (Máximo de 20 caracteres).

 •  Precio Unitario (float, mayor a cero).

 •  Número de depósito (int 1 a 10).

Luego se dispone del archivo VentasDeposito5.dat con las ventas realizadas por la empresa de los productos enviados por el depósito número 5. Cada registro del archivo contiene:

 •  Código de producto (7 caracteres alfanuméricos).

 •  Cantidad vendida (entero 1 y 1000).

 •  Código de comercio (int 1 y 5).

No se sabe la cantidad de registros del archivo y un producto puede ser vendido por varios comercios. Se pide:

 a.  Cargar la información del archivo Productos.dat en un vector de estructuras.

 b.  Cargar la información del archivo VentasDeposito5.dat en una matriz.

 c.  Informar el importe total vendido de cada producto por cada comercio con el siguiente formato:

          PRODUCTO    VENDEDOR 1    VENDEDOR 2     VENDEDOR 3     VENDEDOR 4     VENDEDOR 5

          ABC1023         xx.xx          xx.xx         xx.xx          xx.xx        xx.xx

          GHT1028         xx.xx          xx.xx         xx.xx          xx.xx        xx.xx

          ZDR1035         xx.xx          xx.xx         xx.xx          xx.xx        xx.xx

 d.  Generar el archivo ImporteDeVentas.dat la información del punto 3.

Ejercicio

Se desea efectuar un estudio estadístico del movimiento de encomiendas dentro del país. Se dispone de un primer archivo llamado GEOGRA, de máximo 200 registros con los siguientes campos:

 •  Código de Localidad (entero, Mayor que 0, no consecutivo)

 •  Código de Provincia (entero, de 1 a 24)

 •  Código de Zona Geográfica (entero, de 1 a 9)

También se cuenta con el archivo ENCOM, con los datos de las encomiendas despachadas que posee los siguientes campos:

 •  Código de Localidad (entero, Mayor que 0, no consecutivo)

 •  Peso de la Encomienda en kg. (real, Mayor que 0)

Si un código de Localidad del archivo ENCOM no se encuentra en el archivo GEOGR, debe guardarse el registro en un archivo llamado ERROR.

Se solicita confeccionar un programa que determine e informe:

 a.  Los códigos de las localidades que se encuentran dentro de la zona que ha recibido la menor cantidad de encomiendas.

 b.  La distribución del total de kilos de encomiendas por zona de acuerdo al siguiente formato:

          ZONA    TOTAL DE KILOS

        1         XXXX

          ...        XXXX

        9         XXXX

 c.  Las provincias que no han recibido encomiendas.

Ejercicio

Un banco tiene el archivo secuencia SALDOS, con los saldos de sus casos 20.000 clientes de caja de ahorro, al inicio del mes. Cada registro tiene los siguientes datos:

 •  Número de caja de ahorro (entero, de 6 cifras)

 •  Apellido y nombre (alfanuméricos de 30 caracteres)

 •  Saldo (real, mayor o igual a cero)

Existe, además, un segundo archivo secuencial con los movimientos del mes, MOVI, ordenado por sucursal del banco, Cada registro contiene:

 •  Sucursal del banco (entero de 4 cifras no correlativos).

 •  Número de cuenta de ahorro.

 •  Mes (entero).

 •  Día entero).

 •  Código de Operación (entero 1: depósito (suma al saldo); 2: extracción (resta al saldo)).

 •  Importe (real, mayor que cero)

En el caso de las extracciones, antes de proceder a realizar la resta del importe al saldo, evaluar si éste es mayor al importe (no puede quedar saldo negativo en la Caja de Ahorro). En este caso, guardar en el archivo SIN\_SALDO los datos que vienen en el archivo MOVI.

Confeccionar un programa para que:

 a.  Se actualice el archivo SALDOS con los movimientos registrados.

 b.  Imprima un listado con las cuentas que en el mes han realizado más de 5 extracciones.

Ejercicio

 Para estudiar la población extranjera en la Argentina, se decide realizar un censo por las distintas provincias. Para ello, se dispone del archivo PROVINCIAS de 24 registros de dos campos:

 •  Código de Provincia (cadena de 3 posiciones)

 •  Nombre de la Provincia (cadena de 30 posiciones) Argentina tiene 24 provincias, incluyendo CABA.

La información sobre el censo realizado se encuentra en un archivo secuencial (CENSO) cuyos registros están formados por:

 •  Código de Provincia

 •  Nacionalidad (entero 1 a 50)

 •  Cantidad de habitantes (entero, mayor a cero)

El archivo no está ordenado.

En el archivo secuencial (MODIF\_CENSO), que tampoco está ordenado, se tienen modificaciones, cuyo formato de registro es:

 •  Código de modificación (carácter: ´A´ o ´M´)

 •  Código de Provincia

 •  Nacionalidad

 •  Cantidad (entero, mayor a cero)

Si el código de modificación es ´A´, debe sumarse la Cantidad a la existente; si es ´M´, debe reemplazarse Cantidad de Habitantes con la informada. Se pide:

 a.  Almacenar la información del archivo en memoria utilizando una estructura adecuada. Puede ocurrir que no haya información alguna Provincia – Nacionalidad, en cuyo caso se asume 0 habitantes.

 b.  Actualizar los datos con las modificaciones.

 c.  Informar el total de habitantes por Provincia ordenado en forma descendente por cantidad de habitantes.

 d.  Grabar el archivo actualizado.

Ejercicio

Se dispone de un archivo llamado [RATING.dat](https://elementosprogramacion.github.io/archivos/11.1/RATING.dat) que guarda la información de los puntos obtenidos de rating en los distintos programas de canales de cable a lo largo del día. Cada registro contiene:

 •  Número de canal (entero)

 •  Programa (texto de 35 caracteres máximo)

 •  Rating (float)

El archivo se encuentra ordenado por canal. Se solicita informar:

 a.  Aquellos canales que obtengan menos de 15 puntos en total.

 b.  El canal más visto.

 c.  Generar un archivo con el promedio de rating de cada canal que incluya dos campos:

 •  Número de canal

 •  Promedio

Ejercicio

Se dispone de un archivo con las ventas del mes de cada una de las sucursales de una empresa ([VENTAS.DAT](https://elementosprogramacion.github.io/archivos/11.2/ventas.dat) El archivo se encuentra ordenado por sucursal y contiene la siguiente estructura:

 •  Sucursal (15 caracteres máximo)

 •  Código de producto (entero)

 •  Cantidad (entero)

Por otro lado se dispone de un archivo de los productos ([PRODUCTOS.dat](https://elementosprogramacion.github.io/archivos/11.2/productos.dat) que vende la empresa y el estado del stock al mes pasado con la siguiente estructura:

 •  Código de producto (entero)

 •  Descripción (20 caracteres máximo)

 •  Precio (float)

 •  Stock (entero)

 •  Punto de pedido (entero)

 •  Cantidad para pedido (entero)

No se sabe la cantidad exacta de productos pero sí se sabe que no hay más de 200. Se desea:

 a.  Mostrar el importe total vendido en cada sucursal.

 b.  Mostrar la recaudación total de la empresa en el mes.

 c.  Determinar la sucursal que vendió mayor cantidad de productos.

 d.  Actualizar el archivo de productos con el nuevo stock.

 e.  Realizar un archivo llamado pedidos.dat que incluya aquellos productos que deben solicitarse (código y cantidad) de aquellos productos que al finalizar de procesar las ventas del mes queden con stock menor al punto de pedido.

Ejercicio

 Se dispone de un archivo con nombre [LLAMADAS.dat](https://elementosprogramacion.github.io/archivos/11.3/LLAMADAS.dat) que contiene llamadas telefónicas realizadas en una empresa. Por cada llamada se dispone de la siguiente información:

 •  Sector (15 caracteres máximo)

 •  Duración en segundos (entero)

 •  Tipo de llamada (entero: 1 local, 2 larga distancia, 3 celular)

El archivo se encuentra organizado por Sector. Se solicita:

 a.  Calcular y mostrar por cada sector, la cantidad de llamadas realizadas de cada tipo.

 b.  Determinar cuál es el sector que habló la mayor cantidad de tiempo.

 c.  En un archivo llamado [COSTOS.dat](https://elementosprogramacion.github.io/archivos/11.3/COSTOS.dat) hay un único registro que contiene los costos por segundo de los tres tipos de llamadas (3 float). Se debe generar el archivo gastos.dat que totalice los gastos de llamadas de cada sector.

 d.  Realizar una función que muestre los datos almacenados en gastos.dat en forma de listado.

 e.  Generar un archivo para cada sector con el detalle de llamadas valorizado. El archivo se debe generar con nombresector.dat y debe contener:

 •  Duración en segundos (entero)

 •  Tipo de llamada (entero)

 •  Costo de la llamada (float)

Ejercicio

La empresa de colectivos línea 166 desea saber cuánto debe cobrar en un mes determinado según los viajes realizados y registrados en el sistema mediante la tarjeta SUBE para ello dispone de un archivo de viajes ([VIAJES.dat](https://elementosprogramacion.github.io/archivos/11.4/viajes.dat)) con los viajes del mes que está ordenado por número de Interno (colectivo) que contiene la siguiente información:

 •  Número de Interno (entero)

 •  Número tarjeta SUBE (entero)

 •  Código de Tarifa (texto de 10 caracteres máximo)

 •  Día (entero)

 •  Hora (entero)

Se dispone de un segundo archivo llamado [TARIFAS.dat](https://elementosprogramacion.github.io/archivos/11.4/tarifas.dat) que contiene información de las tarifas por cada mes con la siguiente estructura:

 •  Mes (entero)

 •  Código de Tarifa (texto de 10 caracteres máximo)

 •   Precio (float)

Se sabe que como máximo se manejan 8 tarifas distintas en cada mes.

Se solicita:

 a.  Ingresar por teclado el mes a procesar y recuperar las tarifas correspondientes.

 b.  Mostrar el importe total a cobrar por la empresa.

 c.  Generar un archivo que contenga la cantidad de viajes y la recaudación de cada interno en el mes.

 d.  Informar el interno que realizó menos cantidad de viajes.

 e.  Por cada interno mostrar cuantos viajes se realizaron por cada tipo de tarifa ordenado por cantidad de viajes en forma descendente.

Funciones obligatorias:

 •  CargaTarifas: que levanta a un vector en memoria las tarifas del mes ingresado por teclado.

 •  OrdenarViajes: para ordenar los viajes según lo pedido en el punto e.

 •  MostrarViajes: para mostrar en forma de listado los viajes ordenados del punto e./p>

Ejercicio

En un estadio de fútbol se incorporó un nuevo sistema para censar la cantidad de espectadores que pasan por cada una de las entradas de forma de analizar el comportamiento de la gente y poder diseñar una mejor distribución de las vías de acceso.

El estadio tiene 15 puertas numeradas del 1 al 15. La recolección de datos se realizó en todos los partidos disputados en el estadio a lo largo de varios meses para poder tener un volumen de información suficiente que permita realizar un análisis.

La información registrada se encuentra en un archivo ([EVENTOS.dat](https://elementosprogramacion.github.io/archivos/11.5/eventos.dat)) con la siguiente estructura:

 •  Código del Evento (texto de 10 caracteres máximo)

 •  Número de puerta (entero)

 •  Cantidad de Espectadores (entero)

La información se haya ordenada por Código de Evento. Se solicita calcular y mostrar:

 a.  Cantidad de espectadores por cada evento.

 b.  El evento con menor cantidad de espectadores.

 c.  Grabar un archivo con el promedio de espectadores que ingresaron por cada una de las 15 entradas en todos los eventos analizados, informando el número de puerta y promedio.

Ejercicio

En la feria internacional del automóvil, una empresa autopartista que dispone de un stand registra la información de los visitantes interesados en sus productos. Por cada visitante se ingresa:

 •  Nombre y Apellido (texto de máximo 35 caracteres)

 •  DNI (entero)

 •  Localidad (texto de 40 caracteres máximo)

 •  Email (texto de máximo 20 caracteres)

 •  Fecha (estructura con día, mes y año)

Se solicita:

 a.  Realizar una función para ingresar la información por teclado hasta que se ingrese un visitante con nombre FIN. (se admite como máximo cargar 500 visitantes).

 b.  Una vez finalizado el ingreso de datos se debe generar un archivo llamado visitantes.dat que esté ordenado por localidad.

 c.  Tomando el archivo visitantes.dat se debe calcular y mostrar la cantidad de visitas recibidas por cada localidad y generar un archivo para cada una de ellos conteniendo la información de los visitantes para que dicho archivo puede ser enviado a la sucursal más cercada de forma de contactar a los interesados. Cada archivo generado será nombrado con Nombre de la Localidad.dat.

Ejercicio

La empresa RED VIAL S.A. desea controlar el flujo vehicular en sus cabinas de peaje a lo largo del mes. Para ello se debe procesar la información provista por las barreras que se encuentra almacenada en el archivo [BARRERAS.dat](https://elementosprogramacion.github.io/archivos/11.7/barreras.dat) que contiene la siguiente información:

 •  Día (entero)

 •  Hora (0 a 24)

 •  Número de barrera (entero)

Dentro del archivo la información se encuentra ordenada por día y hora. Determinar:

 a.  La cantidad de vehículos por cada hora.

 b.  La cantidad de vehículos de cada día.

 c.  Ingresar por teclado los precios del peaje para horario normal y horario pico (de 7 a 11 y de 17 a 19). Con esos valores guardar un archivo con la recaudación de cada día.

Ejercicio

Una empresa dispone de un archivo ARTIC\_PRECIO, con los siguientes datos de sus casi 500 artículos:

 •  Número de Artículo (entero, de 8 cifras)

 •  Precio Unitario (real, mayor que cero)

Existe un segundo archivo PROVIN, con los siguientes datos de sus 30 vendedores:

 •  Nombre del Vendedor (alfanumérico de 30 caracteres)

 •  Número de Provincia (entero, de 1 a 24)

Se tiene un tercer archivo llamado VTA\_CANT, ordenado por región del país, con las ventas realizadas. Por cada venta se informa lo siguiente:

 •  Región del país (alfanumérico de 3 caracteres)

 •  Nombre de Vendedor

 •  Mes (entero, de 1 a 12)

 •  Número de Artículo

 •  Unidades Vendidas (entero mayor que cero)

Confeccionar un programa que partiendo de la información suministrada permita determinar e informar:

 a.  Un listado con el importe de las ventas realizadas en cada provincia en cada uno de los meses:

          PROVINCIA    MES1     MES2...     MES12

              XX       XXXX      XXXX       XXXXX

 b.  Un listado con los nombres de los vendedores que no han tenido ventas en más de 3 meses consecutivos.

 c.  Informar en qué región del país se vendió menos (en importe).

Ejercicio

 Un organismo Municipal dispone de un archivo secuencial, ordenado por código de patente, llamado PARQUE con los datos de su parque automotor. Se sabe que en la comuna no hay más de 1000 automóviles registrados.

Cada registro contiene:

 •  Código de patente (alfanumérico de 6 caracteres).

 •  Apellido y nombre del titular (alfanumérico de 30 caracteres)

 •  Domicilio (alfanumérico de 30 caracteres)

 •  Categoría del vehículo (alfanumérico de 4 caracteres)

 •  Año de fabricación (entero mayor a 1900)

Dispone además del archivo llamado PAGO\_PAT, ordenado por categoría del vehículo, con lo recaudado durante todo el año, mes a mes, donde cada registro contiene:

 •  Categoría del vehículo

 •  Código de patente

 •  Importe abonado en el mes 1 ó un cero si no abonó.

 •  Ídem desde el mes 2 hasta el mes 12

Solo hay información de aquellas patentes en las que se abonaron uno o más meses.

Confeccionar un programa que determine e informe:

 a.  La siguiente salida para quienes mantienen deudas:

       PATENTE   NRO.DE MESES ADEUDADOS        DOMICILIO

       IVU977        1 5 10 11 12              MASPERO 137

 b.  Indicar cuál es la categoría de vehículo que recibió la menor cantidad de importe recaudado.

Ejercicios

Confeccionar un programa, con su correspondiente diagrama, para:

 a.  Recuperar la información de las ventas acumuladas por cada artículo hasta la fecha de los 15 vendedores de una empresa, guardadas en el archivo ACUMULADO de acuerdo al siguiente formato de registro:

 •  Número de vendedor (entero de 1 a 15)

 •  Número de artículo (entero de 1 a 50)

 •  Importe total vendido a la fecha (real mayor que cero)

Este archivo solamente tiene datos de los vendedores que vendieron alguna vez algún artículo. El dato del mismo vendedor para el mismo artículo en este archivo no se repite.

 b.  Ingresar la fecha del proceso (día, mes, año).

 c.  Procesar luego los datos de las ventas del mes, que se encuentran en el archivo VENMES, ordenado por número de cliente, con el siguiente diseño:

• Número de cliente (entero de 6 cifras no correlativos)

• Razón Social del cliente (alfanumérico de 35 caracteres).

• Número de vendedor • Número de artículo

• Cantidad vendida (entero mayor que cero)

• Precio unitario (real mayor que cero)

 d.  Actualizar los importes de las ventas anteriores acumuladas en ACUMULADO, generando el archivo ACUMLADO2.

 e.  Informar cuántas ventas se hicieron a cada cliente en el tiempo analizado.

 f.  Informar el número del vendedor que en total tiene registrado el mayor importe de ventas del artículo número 2. Puede haber más de un vendedor en esta condición.

Ejercicio

Un banco tiene el archivo CTAS\_CTES de 100.000 registros como máximo, con información sobre las cuentas corrientes, que se actualiza con los movimientos realizados durante un período. La información del archivo consiste en:

• Número de Cuenta (entero no correlativo)

• Nombre del titular (cadena de 30 caracteres)

• Fecha de último movimiento

• Saldo (real)

• Estado (caracter)

En un segundo archivo MOVIMIENTOS, ordenado por código de sucursal, vienen los movimientos del período (que son débitos y créditos), de los cuales se informan:

• Código de Sucursal (cadena de 3 caracteres)

• Número de cuenta

• Código de movimiento (carácter, C: crédito; D: débito)

• Fecha de movimiento

• Importe de movimiento (real, mayor que cero)

Hacer un programa (Diagrama de lógica y codificación en C) que:

a. Se ingrese por teclado la fecha del proceso (día, mes, año)

b. Se actualice la información del archivo CTAS\_CTES con los movimientos del archivo MOVIMIENTOS, verificando:

• El número de Cuenta del Movimiento que debe existir.

• El estado de la cuenta no debe ser ´I´(Inactiva).

Si se cumplen las dos condiciones, actualizar el saldo con el importe del movimiento (créditos suman, débito restan) y la fecha de último movimiento de la cuenta con fecha de movimiento del movimiento si ésta es posterior a aquella (\*). Si esto último no se cumple, grabar los datos del archivo MOVIMIENTOS en el archivo ERRORES.

c. Después de procesar el archivo MOVIMIENTOS, actualizar el campo Estado con el valor ´I´ cuando el saldo quede en cero y el año de la fecha de último movimiento sea anterior al año del proceso menos 3 años.

d. Generar y grabar 2 archivos:

• El primero de cuentas actualizadas llamado CUENTAS\_ACT, con los registros que no tengan el valor ´I´ en el campo Estado;

• y otro archivo llamado CUENTA\_INACT, con los registros que tengan el valor ´I´ en el campo Estado.

e. Indicar el importe de las operaciones de débito y cantidad de operaciones de crédito por sucursal.

(\*) Utilizar la función FechaMayor (…), a la que se le envían dos fechas y devuelve 1 (uno) si la segunda es mayor a la primera o 0 (cero) si la segunda es menor o igual a la primera. No definirla, sólo utilizarla.

Ejercicio

Este archivo está clasificado y ordenado en forma ascendente por número de visitador médico y luego por el código del medicamento. Pueden NO estar todos los códigos de medicamentos ni todos los visitadores médicos “activos”.

Se desea actualizar las ventas de una empresa con varias sucursales que comercializa caso 2650 artículos. Para ello cuenta con siguientes archivos:

• Archivo ARTICULOS, con el siguiente diseño:

i. Código de artículo (números no consecutivos)

ii. Descripción )20 caracteres)

iii. Unidades en stock

iv. Stock mínimo

v. Precio unitario

Este archivo está ordenado por código de artículo:

• Archivos VENTAS, ordenado por sucursal, con el siguiente diseño:

i. Código de sucursal

ii. Código de producto

iii. Cantidad unidades vendidas

iv. Número de cliente

v. Código de vendedor

vi. Fecha de la venta (día, mes, año)

Desarrollar el diagrama de lógica y su correspondiente codificación C para que permita:

a. Ingresar por teclado la fecha del día (día, mes, año)

b. Actualizar los artículos con la información del archivo VENTAS. Si no existe el artículo, o la cantidad de la venta es mayor que las unidades en stock, rechazar la venta con la leyenda aclaratoria y guardarle en el archivo ERROR.

c. Grabar el archivo ARTICULOS actualizado.

d. Informar, para cada sucursal, el código de sucursal, la cantidad de ventas y el importe total vendido. (no considerar las ventas rechazadas). Encabezar el listado con el título “Ventas por Sucursal al día XX/XX/XXXX”.

e. Finalmente informar cuáles son los artículos que, una vez actualizados, han quedado con stock menor que el stock mínimo (informar código y descripción del artículo)

LIBRERIA PANDA

Ejercicio

Escribir un programa que pregunte al usuario por las ventas de un rango de años y muestre por pantalla una serie con los datos de las ventas indexada por los años, antes y después de aplicarles un descuento del 10%.

Ejercicio

Escribir una función que reciba un diccionario con las notas de los alumno de un curso y devuelva una serie con la nota mínima, la máxima, media y la desviación típica.

Ejercicio

Escribir una función que reciba un diccionario con las notas de los alumnos de un curso y devuelva una serie con las notas de los alumnos aprobados ordenadas de mayor a menor.

Ejercicio

Escribir programa que genere y muestre por pantalla un DataFrame con los datos de la tabla siguiente:

|  |  |  |
| --- | --- | --- |
| Mes | Ventas | Gastos |
| Enero | 30500 | 22000 |
| Febrero | 35600 | 23400 |
| Marzo | 28300 | 18100 |
| Abril | 33900 | 20700 |

Ejercicio

Escribir una función que reciba un DataFrame con el formato del ejercicio anterior, una lista de meses, y devuelva el balance (ventas - gastos) total en los meses indicados.

Ejercicio

El fichero [cotizacion.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/pandas/cotizacion.csv) contiene las cotizaciones de las empresas del IBEX35 con las siguientes columnas: nombre (nombre de la empresa), Final (precio de la acción al cierre de bolsa), Máximo (precio máximo de la acción durante la jornada), Mínimo (precio mínimo de la acción durante la jornada), volumen (Volumen al cierre de bolsa), Efectivo (capitalización al cierre en miles de euros). Construir una función que construya un DataFrame a partir del un fichero con el formato anterior y devuelva otro DataFrame con el mínimo, el máximo y la media de dada columna.

Ejercicio

El fichero [titanic.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/pandas/titanic.csv) contiene información sobre los pasajeros del Titanic. Escribir un programa con los siguientes requisitos:

1. Generar un DataFrame con los datos del fichero.
2. Mostrar por pantalla las dimensiones del DataFrame, el número de datos que contiene, los nombres de sus columnas y filas, los tipos de datos de las columnas, las 10 primeras filas y las 10 últimas filas
3. Mostrar por pantalla los datos del pasajero con identificador 148.
4. Mostrar por pantalla las filas pares del DataFrame.
5. Mostrar por pantalla los nombres de las personas que iban en primera clase ordenadas alfabéticamente.
6. Mostrar por pantalla el porcentaje de personas que sobrevivieron y murieron.
7. Mostrar por pantalla el porcentaje de personas que sobrevivieron en cada clase.
8. Eliminar del DataFrame los pasajeros con edad desconocida.
9. Mostrar por pantalla la edad media de las mujeres que viajaban en cada clase.
10. Añadir una nueva columna booleana para ver si el pasajero era menor de edad o no.
11. Mostrar por pantalla el porcentaje de menores y mayores de edad que sobrevivieron en cada clase.

Ejercicio

Los ficheros [emisiones-2016.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/pandas/emisiones-2016.csv), [emisiones-2017.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/pandas/emisiones-2017.csv), [emisiones-2018.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/pandas/emisiones-2018.csv) y [emisiones-2019.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/pandas/emisiones-2019.csv), contienen datos sobre las emisiones contaminates en la ciudad de Madrid en los años 2016, 2017, 2018 y 2019 respectivamente. Escribir un programa con los siguientes requisitos:

1. Generar un DataFrame con los datos de los cuatro ficheros.
2. Filtrar las columnas del DataFrame para quedarse con las columnas ESTACION, MAGNITUD, AÑO, MES y las correspondientes a los días D01, D02, etc.
3. Reestructurar el DataFrame para que los valores de los contaminantes de las columnas de los días aparezcan en una única columna.
4. Añadir una columna con la fecha a partir de la concatenación del año, el mes y el día (usar el módulo datetime).
5. Eliminar las filas con fechas no válidas (utilizar la función isnat del módulo numpy) y ordenar el DataFrame por estaciones contaminantes y fecha.
6. Mostrar por pantalla las estaciones y los contaminantes disponibles en el DataFrame.
7. Crear una función que reciba una estación, un contaminante y un rango de fechas y devuelva una serie con las emisiones del contaminante dado en la estación y rango de fechas dado.
8. Mostrar un resumen descriptivo (mínimo, máximo, media, etc.) para cada contaminante.
9. Mostrar un resumen descriptivo para cada contaminante por distritos.
10. Crear una función que reciba una estación y un contaminante y devuelva un resumen descriptivo de las emisiones del contaminante indicado en la estación indicada.
11. Crear una función que devuelva las emisiones medias mensuales de un contaminante y un año dados para todos las estaciones.
12. Crear un función que reciba una estación de medición y devuelva un DataFrame con las medias mensuales de los distintos tipos de contaminantes.

Librería matploib

Ejercicio

Escribir un programa que pregunte al usuario por las ventas de un rango de años y muestre por pantalla un diagrama de líneas con la evolución de las ventas.

Ejercicio

Escribir una función que reciba una diccionario con las notas de las asignaturas de un curso y una cadena con el nombre de un color y devuelva un diagrama de barras de las notas en el color dado.

Ejercicio

Escribir una función que reciba una serie de Pandas con las notas de los alumnos de un curso y devuelva un diagrama de cajas con las notas. El diagrama debe tener el título “Distribución de notas”.

Ejercicio

Escribir una función que reciba una serie de Pandas con el número de ventas de un producto durante los meses de un trimestre y un título y cree un diagrama de sectores con las ventas en formato png con el titulo dado. El diagrama debe guardarse en un fichero con formato png y el título dado.

Ejercicio

Escribir una función que reciba una serie de Pandas con el número de ventas de un producto por años y una cadena con el tipo de gráfico a generar (lineas, barras, sectores, areas) y devuelva un diagrama del tipo indicado con la evolución de las ventas por años y con el título “Evolución del número de ventas”.

Ejercicio

Escribir una función que reciba un dataframe de Pandas con los ingresos y gastos de una empresa por meses y devuelva un diagrama de líneas con dos líneas, una para los ingresos y otra para los gastos. El diagrama debe tener una leyenda identificando la línea de los ingresos y la de los gastos, un título con el nombre “Evolución de ingresos y gastos” y el eje y debe empezar en 0.

Ejercicio

El fichero [bancos.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/matplotlib/bancos.csv) contiene las cotizaciones de los principales bancos de España con : Empresa (nombre de la empresa), Apertura (precio de la acción a la apertura de bolsa), Máximo (precio máximo de la acción durante la jornada), Mínimo (precio mínimo de la acción durante la jornada), Cierre (precio de la acción al cierre de bolsa), Volumen (volumen al cierre de bolsa). Construir una función reciba el fichero bancos.csv y cree un diagrama de líneas con las series temporales de las cotizaciones de cierre de cada banco.

Ejercicio

El fichero [titanic.csv](https://aprendeconalf.es/docencia/python/ejercicios/soluciones/matplotlib/titanic.csv) contiene información sobre los pasajeros del Titanic. Crear un dataframe con Pandas y a partir de él generar los siguientes diagramas.

1. Diagrama de sectores con los fallecidos y supervivientes.
2. Histograma con las edades.
3. Diagrama de barras con el número de personas en cada clase.
4. Diagrama de barras con el número de personas fallecidas y supervivientes en cada clase.
5. Diagrama de barras con el número de personas fallecidas y supervivientes acumuladas en cada clase.

# INTEGRADORES

Ejercicio

La Universidad del Litoral desea realizar un informe sobre los exámenes regulares tomados en el

segundo parcial en todas las facultades, para lo cual ingresa como primer dato el código de una de las

carreras que se realizan en esta Universidad (entero, entre 1001 y 4000), sobre la cual va a hacer el

estudio.

A continuación, se ingresa las materias que conforman la carrera, según el siguiente diseño:

Código de la materia (entero, entre 999 y 10000)

Se sabe que no son más de 60 materias (va a depender de la carrera que se elija para hacer el

informe).  El final de la carga se detecta con un código de materia igual a cero.

Luego, se ingresan las notas obtenidas por cada alumno de la Universidad en su examen, según el

siguiente detalle:

o Nro. de DNI (entero, mayor que 1.000.000 y menor que 100.000.000).

o Carrera que cursa (entero, de 501 a 805)

o Facultad en la que se realizó el parcial (entero, entre 1 y 14)

o Código de la materia (entero, mayor que 999 y menor que 10000).

o Nota (entero, mayor que 0 y menor o igual que 10).

Solamente se procesarán los alumnos que correspondan a la carrera ingresada como primer dato. La

carga de este último lote finaliza con un DNI igual a cero.

Se pide:

a) Informar el examen con menor nota detallando número de DNI, Carrera, Materia y Nota. (Si

hay más de uno, informar el último que aparezca).

b) Informar el porcentaje de materias cuyo promedio sea mayor o igual a 7 (siete).

c) Informar la cantidad de exámenes tomados en cada una de las facultades, de cada una de las

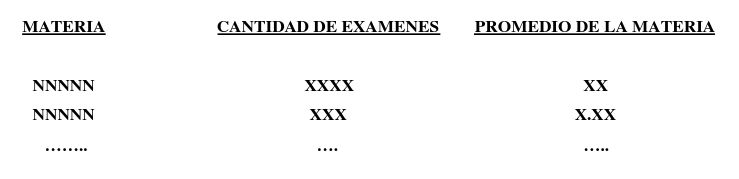
materias.

d) Emitir un informe que indique, por código de materia, la cantidad de exámenes y la nota

promedio obtenida por los alumnos, informando al final la cantidad de exámenes tomados y

el promedio de la carrera. Este informe debe estar ordenado en forma descendente por el

promedio, con el siguiente diseño:



Ejercicio

La empresa textil <El trapito= desea realizar un informe sobre los sueldos de cada sección para lo

cual en primer lugar se debe ingresar la fecha del día de proceso (día, mes y año) y controlar su

validez.  Si no lo es, debe reingresarse hasta que sea correcta.

A continuación, ingresar los datos de sus 35 sucursales en el país, con el siguiente formato:

o Código de la sucursal (entero, mayor a cero).

o Provincia en la que se encuentra (entero, de 1 a 24).

Finalmente, se ingresan los empleados que cobran sueldo en el período informado, la sucursal y la

sección a la cual pertenecen:

o Código de la sucursal

o Sección (entero, de 1 a 8).

o Legajo del empleado (entero, mayor a cero)

o Sueldo a cobrar (real, mayor a cero)

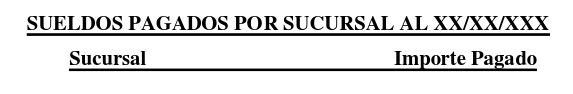
Este último lote de datos no viene ordenado y finaliza con un número de legajo igual a cero.

Determinar:

a) Informar por cada sucursal, que importe se paga para en cada una de las secciones.

b) Informar el total de sueldos de cada sucursal. Este informe debe estar ordenado en forma

descendente por el total de sueldos, con el siguiente diseño:



c) El legajo del empleado con el mayor sueldo consignando legajo, provincia en la que radica la

sucursal y sueldo. (Si hay más de uno, informar cualquiera de ellos).

d) Informar el porcentaje de sucursales que pagan más de $ 1.000.000 en concepto de sueldo.

Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos a y b).

Ejercicio documento terminado en 2 (dos)

La empresa de productos químicos <El Cloro Loco=, desea realizar un estudio sobre los sueldos de

cada departamento para lo cual dispone de los siguientes datos:

En un primer ingreso, se ingresan los datos de las 12 categorías de empleados que poseen:

Código de la Categoría (entero mayor a cero)

Sueldo correspondiente a la categoría (real, mayor a cero)

En un segundo lote de datos, se ingresan los datos de cada empleado, con el siguiente diseño:

Estos datos vienen sin orden alguno y terminan con un legajo igual a cero.

o Legajo (entero, mayor que 0).

o Departamento (carácter, de 8A9 hasta 8H9).

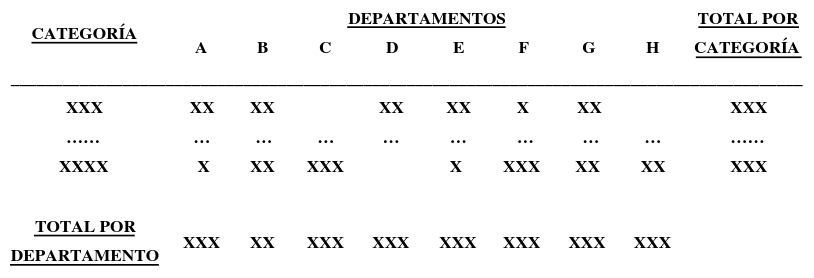
o Categoría (entero, de 100 a 1000)

Estos datos vienen sin orden alguno y terminan con un legajo igual a cero.

Determinar e informar

a) Realizar un listado, indicando por cada categoría, cuántos empleados hay en cada

departamento, según el siguiente diseño:



) El legajo del empleado con mayor sueldo consignando legajo, departamento y sueldo.

(Si hay más de uno, informar cualquiera de ellos).

c) Informar el promedio de sueldos por departamento.

d) Emitir un informe que indique, por departamento, el total de sueldos de la misma.

Este informe debe estar ordenado en forma descendente por el total de sueldos.

Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos a y d).

Nota: el manejo de caracteres debe hacerse sin tomar el código ASCII del carácter.

Ejercicio

Se desea realizar un estudio estadístico sobre los resultados de un censo a nivel nacional, sobre la

cantidad de habitantes de distintas nacionalidades en cada una de las provincias de la República

Argentina. Para ello se dispone de un ingreso de datos con los códigos de las 24 provincias de

nuestro País.

Se tiene también un primer lote de información cargada hasta el momento sobre el censo realizado, a

ingresar por teclado, cuyos datos están formados por:

Código de la Provincia (entero, mayor a cero)

Nacionalidad (Entero, 1 a 50)

Cantidad de Habitantes (Entero, mayor a cero)

No hay garantía de que estos datos vengan ordenados, ni que haya información de todas las

provincias y de todas las nacionalidades.  El fin de datos se verifica con un código de provincia igual

a cero.

En un segundo lote de datos, se tienen modificaciones, con el siguiente formato:

Código de modificación (Carácter: 8A9, 8M9 o 8F9)

Código de la Provincia

Nacionalidad

Cantidad (entero, mayor a cero)

Si el código es 8A9, debe sumarse la Cantidad a la existente; si es 8M9, debe reemplazarse Cantidad

de Habitantes con la ingresada en este segundo lote de datos. La información finaliza con el código

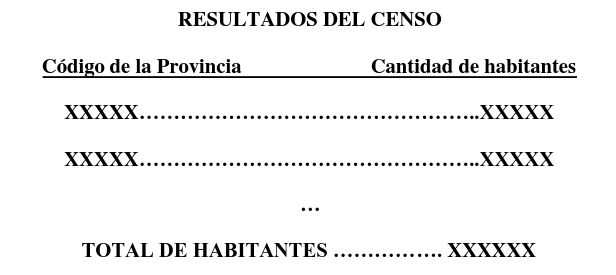
de modificación igual a 8F9.

Determinar:

a) Informar para cada provincia, de cada nacionalidad, la cantidad de habitantes final.

b) Genere un listado con la cantidad de habitantes (resultante de la actualización) de cada

Provincia, ordenado decrecientemente por cantidad de habitantes, con el siguiente formato:



c) Indicar para todas las nacionalidades, cuál es el código de la provincia en la que tiene menor

cantidad de habitantes (puede haber más de una con la menor cantidad de habitantes).

d) Indicar el porcentaje de provincias cuya cantidad de habitantes de todas las nacionalidades

supere los 100.000 habitantes.

Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos a y b).

Ejercicio

Una empresa de alquiler de máquinas expendedoras de bebidas calientes en distintos organismos

públicos, cada día recibe un listado en el cual están guardados todas las ventas de bebidas realizadas

el día anterior en todos los organismos. Se desea conocer información de alguno de estos

organismos, cuyo código (entero, de cuatro cifras, mayor a cero), debe ser previamente ingresado por

el usuario desde el teclado al comenzar el programa.

Luego ingresar los códigos de las 12 bebidas que se venden habitualmente y su precio, con el

siguiente formato:

o Código de la bebida (entero, mayor a cero)

o Precio unitario de la bebida (real, mayor de cero)

A continuación, ingresar el listado de ventas de un determinado período de tiempo, procesando

solamente aquellos datos en que el código del organismo público coincida con el código del

organismo público ingresado al principio del programa, con el siguiente formato: (los datos vienen

sin orden alguno):

o Código del organismo público (entero de 4 cifras, mayor a cero)

o Número de la máquina (entero de 1 a 50)

o Código de bebida vendida (entero, mayor a cero)

Esta información termina cuando se ingresa un número de máquina negativo.  Puede haber máquinas

fuera de servicio en el período evaluado o alguna bebida que temporalmente no esté disponible.

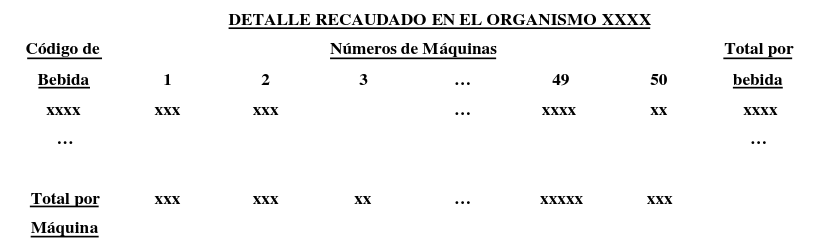
Determinar:

a) El código de la bebida que recaudó más dinero en dicho organismo, en total en todas las

máquinas.

b) Informar cuánto dinero se recaudó por bebida en cada máquina dentro del organismo

estudiado, con el siguiente formato:



c) Informar el importe vendido por máquina ordenado en forma descendente por importe. No

considerar las que estuvieron inactivas durante el día analizado en dicho organismo.

d) Indicar el porcentaje de bebidas que vendió más de $ 15.000.

Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos b y c).

Ejercicio

El consorcio del barrio privado <El Zorzal=, quiere calcular los importes de las expensas

correspondientes al servicio de seguridad de un determinado mes.

Para ello, se pide al comienzo del programa ingresar el mes (entero, de 1 a 12) y del año (entero,

mayor a 2020) a informar y a continuación pida el ingreso del valor del importe de seguridad por

metro cuadrado del predio de construcción (real, mayor a cero).

A continuación, ingresar los siguientes datos de las 45 construcciones que existen en el barrio

privado:

Código del propietario (entero, de 1 a 45)

Categoría de la construcción (carácter, de 8A9 a 8E9)

Superficie del predio en metros cuadrados (real, mayor a cero)

Para finalizar la carga de los datos se ingresa un código de propietario igual a cero y se sabe que la

información no tiene orden alguno.  Cada predio tiene solamente una categoría asociada.

Desarrollar e informar:

a) El código del propietario que menos expensas por seguridad paga de cada categoría (asumir

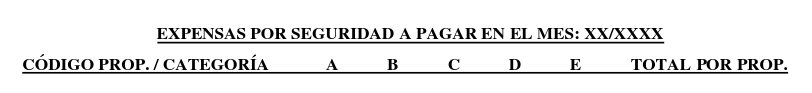
que solamente es uno).

b) Indicar el código de propietario y la categoría del propietario que paga más de $ 80.000 de

expensas.

c) Informar cuánto dinero debe pagar cada propietario de cada uno de los predios, con el

siguiente formato:



d) Informar cuánto dinero se recaudará de expensas por seguridad en cada categoría durante el

mes, ordenado de menor a mayor por importe.

Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos c y d).

Ejercicio

En el contexto de la Pandemia de COVID 19, el Ministerio de Salud de la Nación (MSN) realiza un

estudio correspondiente al lapso Diciembre 2020 a Mayo 2021, respecto a las vacunas de la primera

dosis entregadas semanalmente a cada distrito (provincias y CABA). Para ello, en primer lugar, se

ingresa los siguientes datos correspondiente a los 24 distritos del País:

Código del distrito (entero, de cuatro cifras, mayor a cero)

A continuación, se ingresan las cantidades entregadas por distrito y vacuna de forma semanal hasta el

momento.  En cada entrega a cada distrito, se tienen los siguientes datos:

Código de la vacuna (entero, de 1 a 10)

Cantidad entregada de la primera dosis (entero, mayor a cero)

Distrito al cual se entrega (entero, de cuatro cifras, mayor a cero)

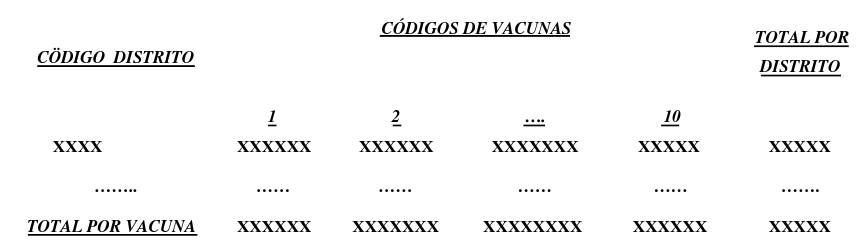
Número de semana (entero, de 1 a 21)

La última entrega a distritos de primeras dosis de vacunas se detecta con un código de vacuna igual a

0 (cero).

Determinar e informar:

a. Por cada distrito, la cantidad de vacunas recibidas de cada marca, con el siguiente diseño:



b. Informar el código de la vacuna y el distrito donde se hizo la entrega más grande semanal.  Si

hubiera más de uno en la misma situación, informar el último que aparezca.

c. Indicar el porcentaje de semanas en las cuales no se entregó vacunas a ningún distrito.

d. Por cada distrito, el código del mismo y la cantidad de vacunas recibidas, de todas las marcas

posibles.

Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos a y d).

Ejercicio

El canal de televisión CINETOTAL desea emitir un ciclo de películas clásicas durante 8 (ocho)

semanas a partir del mes de junio del corriente año, para lo cual realiza una votación on line a través

de las redes sociales, para que sus televidentes elijan alguna de los 130 títulos de películas clásicas

con las que cuenta el canal de televisión, y además en qué semana prefiere que se emita dicha

película.

En un primer lote de dato se ingresa el listado de los códigos (entero, mayor a cero) de las 130

películas.

En la votación se le solicita al televidente:

Código de la Película de su preferencia

Semana preferida para la proyección (entero, de 1 a 8)

Se debe tener en cuenta que cada ingreso de un par de datos de nombre de la película y semana de

proyección corresponde a un solo voto.  El fin de carga de datos se realiza con un código de película

igual a -1.  No se garantiza ningún orden en la información.

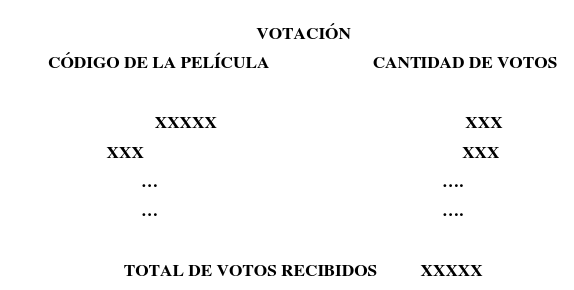
Determinar e informar:

a) Informe cuántos votos recibió cada película en cada una de las semanas.

b) Informe qué semana recibió la mayor cantidad de votos para cualquiera de las películas.

c) Informe el total de votos recibido por cada película, ordenado por cantidad de votos en

forma descendente, con el siguiente formato:



d) Informar el porcentaje de películas que NO recibieron ningún voto en ninguna de las

semanas.

Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos a y c).

Ejercicio

Una empresa textil desea realizar un informe sobre el stock de cada uno de sus depósitos para lo cual

dispone de los siguientes datos:

La fecha del día de proceso, en tres campos enteros: día, mes y año. Debe ingresarse al inicio

del programa y controlarse que sea fecha válida y, si no lo es, debe reingresarla hasta que sea

correcta.

En una primera carga ingresar los 30 códigos de artículos, las unidades existentes en stock y el

costo unitario de cada artículo, según el siguiente detalle:

o Código de Artículo (entero, de 6 cifras, mayor a cero)

o Unidades en Stock (entero, mayor o igual a cero)

o Costo Unitario (real, mayor que 0)

A continuación, ingresar los movimientos del día, según el siguiente detalle:

o Código de Artículo (entero, de 6 cifras, mayor a cero).

o Depósito (entero entre 1 y 15).

o Código de operación (carácter, 8I9 si ingresa al stock (suma); 8E9 si sale del stock (resta))

o Cantidad (entero, mayor que 0).

De un mismo artículo y depósito, pueden venir muchos datos o ninguno. Controlar el fin de datos

con un código de artículo menor a cero.

Determinar e informar:

a) El artículo con mayor stock valorizado consignando código de Artículo, Costo

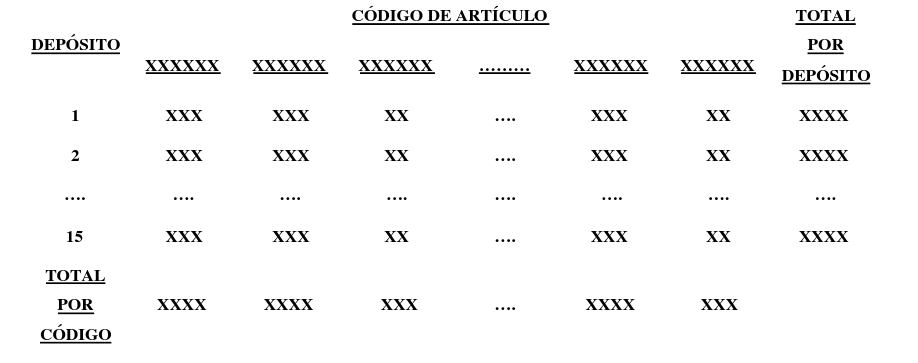
unitario y unidades en stock. (Si hay más de uno, informarlos a todos).

b) Informar el porcentaje de artículos que quedan con stock igual a cero, respecto del

total de artículos.

c) Por cada depósito, informar las unidades en stock de cada artículo según el siguiente

detalle:



d) Informar al finalizar los movimientos del día, el stock total valorizado a costo del

mismo y el depósito al cual pertenece. Este informe debe estar ordenado en forma

descendente por el total de stock valorizado. (El stock valorizado a costo de cada

artículo es las unidades en stock por el costo unitario).

Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos c y d).

Ejercicio

Una empresa de turismo desea analizar las ventas efectuadas a la fecha para el verano 2022 de los 19

tours que ofrece.

Para ello, ingresa en primer lugar la fecha del día del proceso (día, mes y año).

A continuación, se ingresa desde el teclado los datos de los tours.  Por cada tour, se ingresa:

Código de tour (entero, entre 10000 y 100000)

Precio unitario (real, mayor a cero)

Luego se ingresan los datos de las ventas del período, por teclado, con el siguiente detalle:

Código de cliente (entero, mayor a cero)

Código del vendedor (entero, de 1 a 10)

Código de tour

Cantidad vendida (entero, mayor que cero)

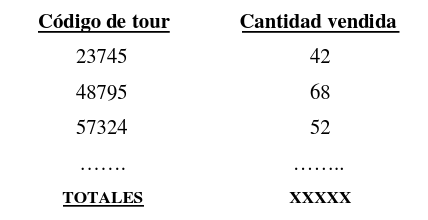
Estos datos no se ingresan ordenados y la lectura debe detenerse cuando se ingrese un código de

cliente menor a cero. Si se ingresa un código de tour erróneo, rechazar la venta con una leyenda

aclaratoria.

Determinar e informar:

a) Al finalizar el proceso, informar la cantidad vendida de cada tour.

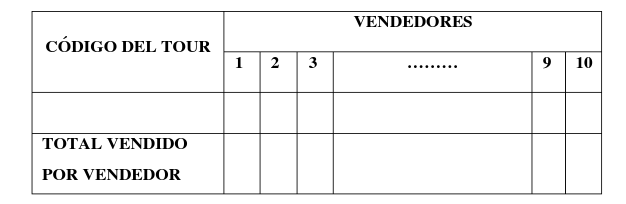


b) Informar qué cliente compró la mayor cantidad de tours, en una única compra. Si hubiera

más de uno, informar el primero que aparezca.

c) Informar por cada tour, el importe vendido por cada uno de los vendedores, con el siguiente

diseño:



d) Informar el promedio de ventas respecto al importe cobrado y la cantidad vendida.

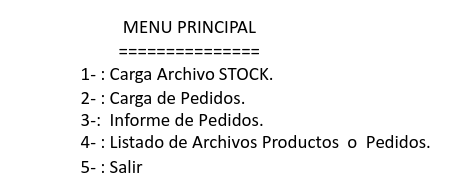
 Funciones obligatorias: Validación de datos, Búsqueda y Listados (puntos a y c).

Ejercicio

La Empresa “ENIGMA” de Alan Turing que se dedica a la venta por mayor de productos de informática.

Desea controlar los pedidos de sus clientes.    Solicita que se le confeccione un programa con las siguientes

opciones que ingresa con Menú:



A continuación, se detalla que debe realizar cada una de las opciones:

1-: Carga Archivo STOCK.

Comercializa casi 3500 productos diferentes, para ello se cuenta con los siguientes datos que se hallan en

un archivo llamado “stock.dat”, se ingresa por teclado:

                          • Código de Producto (int, entre 1000 y 10000)  VALIDAR

                          • Descripción de Producto (char, 30 caracteres) VALIDAR

                                       • Precio (float) VALIDAR

                         • Punto de Pedido (int). VALIDAR (Medida, para solicitar reposición de productos)

                         • Stock (int > 0) VALIDAR

Los productos se deben guardar en un archivo llamado “stock.dat”. Para no ser cargados cada vez que se

inicia el programa, este punto debe preguntar; C - CONTINUA , I – INICIA , S – SALE . Esto es para el modo de

apertura del archivo, que permite continuar o grabar desde 0, o  bien salir del punto sin hacer nada.  Se pueden

agregar productos siempre y cuando no se supere los 3500. El ingreso por teclado de productos finaliza con una

Descripción de Producto = “FIN”.

2- : Carga de Pedidos.

Permite hacer la carga por teclado los pedidos que debemos cumplir para el mes siguiente, se almacena

en archivo “pedidos.dat”, con la siguiente estructura:

                           • Código de Producto (int, entre 1000 y 10000) VALIDAR la Existencia con Buscar

                           • Cantidad de unidades pedidas (int > 0) VALIDAR

                           • Número Cliente (int, entre 1 y 3000) VALIDAR

                          • Número de Pedido (int, entre 1 y 2000)  VALIDAR

                          • Dia\_Fecha (int, entre 1 y 31)  (corresponde día de entrega del pedido).

NOTA: Rechazar los pedidos de aquellos productos cuyo stock es inferior al pedido. Grabar los Pedidos que

no se pueden cumplir, en otro archivo llamado   “No\_stock.dat” con la misma estructura de pedidos.dat. Usar la

función Buscar para verificar la existencia del Producto y Descontar la Cantidad de Unidades pedidas.

Consideraciones: • Cada vez que se selecciona una opción del menú se ejecutan las acciones y luego se vuelve a

mostrar el menú hasta que el usuario decida salir y cerrar el programa. Cada punto es individual, recuerde grabar.

                             • Tenga en cuenta, que hay acciones que no se pueden realizar si antes no se hicieron otras,

                ejemplo: Si no hay carga de punto 1 y 2, no tendremos resultados en puntos siguientes.

Recuerda al termino enviar este documento, más el Código desarrollado y en formato PDF

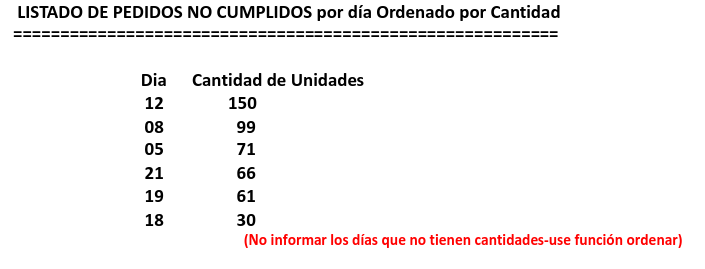
Al salir de este punto (Código de Producto= 99, debe actualizar el archivo “stock.dat”, actualizados con

los pedidos cumplidos en este punto.

3-:  Informe de Pedidos.

Tomando la información almacenada en el archivo “No\_stock.dat” informar un listado ordenado

por Cantidad de Unidades pedidas de mayor a menor,  los distintos día, con el siguiente formato:



4- : Listado de Archivos Generadores o Pedidos.

       Listado por Pantalla de Archivos cargados en el punto 1 y 2. Pedir por pantalla cual se quiere mostrar:

A- Lista Archivo STOCK   (Datos:  Código Producto,  Descripción ,Punto Pedido, Stock.

B- Lista Pedidos (Código Producto,  Cantidad Pedida, Numero de Cliente, Fecha ).

5- : Salir

Ejercicio

 Una empresa dedicada al streaming desea realizar el control de las descargas de sus

películas en la última semana. Para ello dispone del archivo “Peliculas .dat” con un registro por

cada película ofrecida por la plataforma, de cada registro se tiene: - código de película

(alfanumérico 4 caracteres) - título de la película (35 caracteres) - género (20 caracteres). No se

conoce la cantidad exacta de películas, pero sí se conoce que no son más de 500. Se dispone

también del archivo “Descargas.dat” con un registro por cada descarga realizada en la última

semana. De cada descarga se tiene: - código de película - nombre del usuario (20 caracteres).

Se pide realizar un programa en lenguaje C para: . a) Generar el archivo “PeliDescarga.dat”, con

un registro por cada película que haya tenido descargas, cuyos registros tengan código y título

de película y cantidad total de descargas realizadas en la semana. b) Informar, por cada película

del género “Drama”, código, título y cantidad total de descargas realizadas en la semana. c)

Informar cantidad de películas que tuvieron más de 50 descargas. d) Informar cantidad de

películas del género “Comedia” que no tuvieron descargas.

## BASE DE DATOS

Hace unos días me dejaron un comentario en una de las entradas anteriores, planteando un ejercicio que me ha parecido interesante compartir con ustedes.

Para darle un poco de complejidad al ejercicio, la idea es hacerlo utilizando una base de datos (en este caso utilizaremos Sqlite3): puedes ver la entrada que hice sobre Sqlite3 ([Python y Sqlite3 como base de datos](https://www.pythondiario.com/2013/12/python-y-sqlite3-como-base-de-datos.html)) para que sirva como guía.

Ejercicio

Una

PyME, tiene la siguiente estructura de pagos para sus 10 empleados:

Un sueldo

base

Una bonificación del 1% del sueldo base, por

cada mes trabajado

Una asignación familiar del 5% del sueldo base,

por cada hijo

La suma de los tres valores anteriores, conforman

la “base imponible”.

Todos los empleados están en FONASA, así que

deben cotizar el 7% de la base imponible en salud. Los empleados están en una de dos: AFPs, la

primera cobra (entre imposición y otros gastos) el 12 % de la base imponible, mientras que la

segunda cobra el 11.4%

#### Construyan un programa Python que:

a) Pida el ingreso de datos de los 10 empleados

y los almacene. Debe pedir: nombre, apellido, sueldo base, afap, fecha de ingreso

y cantidad de hijos.

b) El programa debe calcular la base imponible,

según lo indicado arriba y luego descontar según corresponda.

c) El programa debe calcular lo que se debe

pagar a FONASA y el monto de cada AFAP.

d) El programa debe calcular los promedios de

pago a los empleados

e) El programa debe implementar control de

excepciones en cada ingreso de información.

El mensaje debe ser claro al usuario, indicando

que debe corregir en cada intento de ingresar los datos.E

Se entiende por:

FONASA: El Fondo Nacional de Salud, FONASA, es el organismo público encargado de otorgar cobertura de atención, tanto a las personas que cotizan el 7% de sus ingresos mensuales en FONASA, como a aquellas que, por carecer de recursos propios, financia el Estado a través de un aporte fiscal directo.

AFAP: AFAP significa “Administradora de Fondos de Ahorro Previsional”. Son empresas que administran parte del aporte de los trabajadores afiliados para que en el futuro tengan una mejor jubilación.

Ejercicio 245

Determinar la cantidad de dígitos de un numero (1- 100000)