



Elementos de Programación

Guía de Ejercicios Prácticos

INDICE

UNIDAD 2 - SECUENCIAL	2
UNIDAD 4 - DECISIÓN	3
UNIDAD 5 - ITERACIÓN	4
5.1 ITERACIÓN DEFINIDA	4
5.2 ITERACIÓN CONTROLADA POR CONDICIÓN	7
UNIDAD 6 - FUNCIONES	10
UNIDAD 7 - ARREGLOS	12
7.1 VECTORES	12
7.2 MATRICES	16
UNIDAD 8 - STRINGS	20
UNIDAD 9 - ESTRUCTURAS	21
UNIDAD 10 - ARCHIVOS	23
UNIDAD 11 - CORTE DE CONTROL	33

Unidad 2 - Secuencial

- 2.1. Se ingresan la cantidad de horas trabajadas y el valor por hora de un empleado. Determinar el sueldo.
- 2.2. Se ingresan las notas de dos evaluaciones de un alumno. Determinar la nota promedio.
- 2.3. Confeccionar un programa que pueda ingresar 2 números enteros y calcule e informe con mensajes aclaratorios la suma, el producto, el cociente y el resto.
- 2.4. Confeccionar un programa que ingrese una medida en 'pies' y la exhiba convertida a yardas, pulgadas, cms. y mts. NOTA: 1 pie = 12 pulgadas; 1 yarda = 3 pies; 1 pulgada = 2,54 cms.
- 2.5. Conociendo la cantidad de tarros de pintura que existe en un depósito de una pinturería y sabiendo que el 50% son tarros de 1Lt, el 30% tarros de 4Lts. y el resto tarros de 20Lts. Determinar la cantidad de tarros de 1Lt., 4Lts. Y 20 Lts.
- 2.6. 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.
- 2.7. Se ingresa un número entero de 3 cifras. Descomponerlo en unidad, decena y centena.
- 2.8. Se ingresa un número entero que representa una fecha con formato (ddmmaa). Se pide transformarlo a un número con formato (aammdd).
- 2.9. Confeccionar un programa que ingrese un valor expresado en Kibibyte (KiB)¹ y lo informe expresado en: TiB, GiB, MiB, con leyendas aclaratorias. (1MiB = 1024 KiB; 1GiB = 1024 MiB; 1TiB = 1024 GiB)
- 2.10. Confeccionar un programa que solicite el ingreso del valor del radio (r) de un círculo y con dicho valor calcule la superficie del círculo, la longitud de la circunferencia (perímetro) y el volumen de la esfera.
$$\begin{aligned} \text{Superficie} &= \pi r^2 \\ \text{Perímetro} &= 2\pi r \\ \text{Volumen} &= \frac{4}{3}\pi r^3 \end{aligned}$$
- 2.11. 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 como por unidad. Indicar el precio total a abonar.
- 2.12. Una farmacia vende algunos artículos sin descuento y a otros con descuento del 20%. Confeccionar un programa que recibiendo el precio original y un código que indica si es o no con descuento, informe el precio final (0 no aplica el descuento y 1 aplica el descuento).
- 2.13. 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 hasta llegar a una sola cifra. Ej: 2561 → 5.

¹ Contracción de kilo binary byte. Unidad definida como estándar ISO/IEC desde el año 1999. Se creó para diferenciarlo del kilobyte que se corresponden con 1000 Bytes

Unidad 4 - Decisión

- 4.1. Se ingresan 3 números distintos. Determinar el mayor.
- 4.2. Dados tres números determinar e informar con un mensaje si el primer número ingresado es menor que los otros dos.
- 4.3. Confeccionar un programa que pueda determinar de 3 números enteros que se ingresan si alguno de ellos es igual a la suma de los otros dos.
- 4.4. Ingresar 3 valores reales y:
 - a. Si los dos primeros son mayores al tercero informar "MAYORES AL TERCERO".
 - b. Si los tres son iguales informar "TRES IGUALES".
 - c. Si alguno de los dos primeros es menor al tercero informar "ALGUNO ES MENOR".
- 4.5. Se ingresan 3 números enteros. Informarlos en orden creciente.
- 4.6. Confeccionar un programa que permita ingresar 4 números enteros, de a uno por vez y determine el menor valor y su número de orden. Informe los valores ingresados e identifique al menor con mensaje aclaratorio.
- 4.7. Confeccionar un programa que solicite e ingrese 3 valores reales positivos, mayores que cero y determine e informe si forman o no triángulo. Para ello utilizar el teorema de la desigualdad del triángulo que establece que la suma de las longitudes de cualesquiera de dos lados de un triángulo es mayor que la longitud del tercer lado.
- 4.8. Realizar un programa que ingrese dos números e indique si el primer número es divisible por el segundo.
- 4.9. Confeccionar un programa que permita ingresar un carácter alfanumérico y determine e informe si lo ingresado corresponde a una vocal, con el mensaje "VOCAL". y su correspondiente valor numérico en ASCII.
- 4.10. Confeccionar un programa que permita convertir grados sexagesimales a radianes y viceversa, según el valor de un código que se ingresa junto al valor. Si código = 1 se ingresan grados, si es 2 se ingresan radianes.
$$1 \text{ radian} = \frac{180}{\pi} \text{ grados}$$
$$1 \text{ grado} = \frac{\pi}{180} \text{ radianes}$$
- 4.11. Escribir un programa que solicite el año y el número de mes y nos informe cuantos días tiene dicho mes. Debemos considerar que el año puede ser bisiesto. Un año es bisiesto cuando es divisible por 4 y no por 100 o el año es divisible por 400.
- 4.12. Ingresar el valor de la hora y la cantidad de horas trabajadas por un empleado. Calcular su sueldo tomando en cuenta que recibe un premio de \$500 si trabajo más de 50 horas y, además, si trabajó más de 150 horas se le otorgan \$1000 adicionales.
- 4.13. Un fabricante de repuestos para tractores ha descubierto que ciertos artículos identificados por los números de catálogo 12121 al 18081; 30012 al 45565 y 67000 al 68000 son defectuosos. Se desea confeccionar un programa al que informándole el número de catálogo indique si el artículo es o no defectuoso. Los artículos del catálogo van desde el 1200 al 90000. Si se ingresa otro número informar "FUERA DE CATALOGO".
- 4.14. La farmacia Sindical efectúa descuentos a sus afiliados según el importe de la compra con la siguiente escala:

- a. menor de \$55 el descuento es del 4.5%
- b. entre \$55 y \$100 el descuento es del 8%
- c. más de \$100 el descuento es del 10.5%

Confeccionar un programa que reciba un importe e informe: el descuento y el precio neto a cobrar, con mensajes aclaratorios.

4.15. Un negocio de artículos de computación vende DVDs según la siguiente escala de precios:

- sueltos, entre 1 y 9 a \$15 c/u.
- la caja de 10 cuesta \$120.
- la caja de 50 cuesta \$500.

Si la compra excede las 100 unidades se efectúa un descuento del 10%.

NOTA: se aceptan compras entre 1 y 500 DVDs.

Se pide que se ingrese una cantidad de DVDs que se quiere comprar y calcule e informe el importe a pagar.

4.16. Un negocio vende distintos artículos identificados por un código, según se muestra:

- código 1; 10; 100: 10 pesos la unidad.
- código 2; 22; 222: 7 pesos la unidad. La caja de 10 unidades vale 65 pesos.
- código 3; 33: 3 pesos la unidad. Si la compra es por más de 10 unidades se hace un descuento del 10% sobre el total de la compra.
- código 4; 44: 1 peso la unidad.

Confeccionar un programa que ingrese como dato el código de un artículo y la cantidad a comprar y se informe el importe de la compra, con las siguientes leyendas:

ARTÍCULO xxxxx CANTIDAD xxxx IMPORTE A PAGAR \$ xxxx.xx

4.17. Confeccionar un programa que pueda determinar el importe a pagar por una prestación médica según su código. Los importes se cobran según la siguiente tabla:

Código	Importe
A	20
D	40
F	60
M	150
T	150

Se debe ingresar el número de historia del paciente y el código de la prestación. Se debe emitir un ticket con el número de la historia, el código y el importe a pagar.

4.18. Confeccionar un programa que permita ingresar la fecha actual (día, mes y año) y la fecha de nacimiento de una persona (día, mes y año). Con ambos datos debe calcular la edad de dicha persona.

Unidad 5 - Iteración

5.1 Iteración Definida

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

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

- 5.1.3 Se ingresan 50 números enteros. Determinar el promedio de los números pares.
- 5.1.4 Se ingresan 100 letras. Determinar la cantidad de vocales ingresadas.
- 5.1.5 Confeccionar un programa para calcular la suma de los primeros N números naturales. El valor de N lo solicita por teclado el programa.
- 5.1.6 Ingresar N y N Números naturales. Determinar e informar:
- La sumatoria de los valores múltiplos de 3.
 - La cantidad de valores múltiplos de 5.
 - La sumatoria de los valores que se ingresan en orden par.
- 5.1.7 Dadas 20 notas y legajos de alumnos de un curso. Determinar:
- Cantidad de alumnos aplazados.
 - Cuando la nota sea mayor a 7, mostrar el mensaje: "El alumno con legajo xxx ha Promocionado".
- 5.1.8 Dadas las edades y estaturas de 45 alumnos de un curso. Determinar:
- Edad promedio.
 - Estatura promedio.
 - Cantidad de alumnos mayores de 10 años.
 - Cantidad de alumnos que miden menos de 1.40 cm.
- 5.1.9 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.
- 5.1.10 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:
- el promedio de los positivos.
 - el promedio de los negativos.
 - 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".
- 5.1.11 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.
- 5.1.12 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 $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$
- 5.1.13 Realizar un programa que:
- Muestre todos los números primos entre 1 y 100. (Un nro. es primo cuando es divisible solamente por 1 y por sí mismo)
 - Contar y mostrar la cantidad de primos encontrados.
- 5.1.14 Confeccionar un programa que determine e informe los números perfectos comprendidos entre 1 y 8000. Un número es perfecto cuando es igual a la suma de sus divisores positivos menores que él.
- 5.1.15 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:
- Cuantos disparos se efectuaron en cada cuadrante por cada participante
 - Cuantos disparos se efectuaron en total en el centro.

NOTA: no considere disparos sobre los ejes.

- 5.1.16 Se ingresan DNI y la nota promedio de 10 alumnos. Determinar el DNI del alumno de mayor nota promedio (considerar único).
- 5.1.17 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.
- 5.1.18 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

- 5.1.19 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:

```
*
**
***
****
*****
*****
```

- 5.1.20 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:

```
  *
 * *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
```

- 5.1.21 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.

```
*
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * *
* * * *
* * *
* *
*
```

5.2 Iteración Controlada por Condición

5.2.1 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 ...)

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

5.2.3 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.

5.2.4 Un negocio de venta de granos desea controlar las ventas realizadas. De cada una ingresa el importe total y un código que indica la forma de pago. El código puede ser:

- C: cheque, 20% de recargo.
- E: efectivo, 10% de descuento.
- T: con tarjeta, 12% de recargo.

Se debe ingresar una F para finalizar el día de venta y arrojar los siguientes totales.

Efectivo en Caja:	\$ xxxx.xx
Ventas con Tarjeta de Crédito:	\$ xxxx.xx
Ventas con cheque:	\$ xxxx.xx
Total de Venta:	\$ xxxx.xx
Importe del IVA:	\$ xxxx.xx

Nota: El IVA corresponde al 21% del total de ventas.

5.2.5 De cada uno de los alumnos de un curso se ingresa su número de DNI y las notas de los dos exámenes parciales confeccionar, un programa que muestre por cada alumno su condición final con el siguiente mensaje:

"El alumno con DNI: xxxxxxxx, obtuvo las notas: xx y xx [condición final]"

La condición final se establece según las siguientes reglas:

- PROMOCIONA: ambas notas ≥ 7
- RINDE EXAMEN FINAL: ambas notas ≥ 4
- REPROBO LA MATERIA: alguna nota no cumple lo anterior

El programa finaliza cuando se ingresa un valor negativo como número de DNI. Al finalizar mostrar la nota promedio de alumnos promocionados.

5.2.6 Un negocio de perfumería efectúa descuentos en sus ventas según el importe de las mismas, con la siguiente escala:

- menor a 50 pesos el 3.5 %
- entre 50 y 150 pesos el 10 %
- entre 151 y 300 pesos el 20 %.
- mayor a 300 pesos el 25 %

Confeccionar un programa que:

- a. Solicite un importe y calcule el descuento a efectuar y el importe neto a cobrar, con mensajes aclaratorios.
- b. Informe el importe promedio de todas las ventas realizadas.

Se debe contemplar que se puedan ingresar varios importes y para finalizar se ingresa un valor negativo o cero.

5.2.7 Ingresar diversos quintetos de números positivos de 1 cifra c/u, finalizando cuando las cinco cifras leídas sean iguales a cero. Determinar e informar cuantos de los quintetos ingresados han formado números capicúas 'triples' y cuantos 'quíntuples'.

5.2.8 Confeccionar un programa que me permita calcular y mostrar una cantidad de números primos que se solicita por la pantalla. Los números se calculan a partir del 2. Ejemplo: Si ingreso 6 genero los números: 5 - 7 - 11 - 13 - 17 - 19. El algoritmo para el cálculo de si un número es primo o no debe estar optimizado para que si encuentra un divisor no seguir buscando ya que se sabe que no es primo.

5.2.9 Se desea realizar un programa para calcular el sueldo final a pagar a cada empleado de una empresa. De cada uno se tiene, sueldo básico, antigüedad, cantidad de hijos y estudios superiores ('S' o 'N'). Además, se conocen los porcentajes de aumento del sueldo que dependen de los siguientes factores:

- Si el empleado tiene más de 10 años de antigüedad: aumento del 10%
- Si el empleado tiene más de 2 hijos: aumento del 10%, si solo tiene uno 5%
- Si el empleado posee estudios superiores: aumento del 5%

Luego de ingresar los datos de un empleado se debe preguntar si se desea ingresar otro empleado o no. Se termina la carga cuando no se deseen ingresar más empleados.

Determinar:

- a. Por cada empleado: número de empleado, el sueldo básico y el nuevo sueldo.
- b. Sueldo nuevo promedio de la empresa.

5.2.10 En un proceso de control se evalúan valores de temperatura, finalizando cuando de ingresa un valor ficticio de temperatura -50. Determinar el valor máximo y mínimo medido y sus posiciones respecto al ingreso.

5.2.11 Se ingresan números enteros, hasta que se ingrese el número cero. Determinar los 2 valores mayores.

5.2.12 Confeccionar un programa que permita el ingreso de 6 ternas de valores enteros.

- a. de cada terna informar el mayor valor y su orden en la terna.
- b. de cada terna informar cuantos valores son múltiplos de 7.
- c. de las 6 ternas determinar e informar el promedio de los primeros valores de cada una.

5.2.13 Realizar un programa que le solicite al usuario el ingreso de una vocal (este dato de ingreso se debe validar). Luego el programa debe dibujar en pantalla la vocal ingresada en cinco líneas y cuatro columnas. La vocal debe estar formada en cada carácter por la misma letra en mayúscula.

El programa finaliza al ingresar la letra f.

Restricciones: Por cada printf se debe exhibir un solo carácter o salto de línea o espacio.

Ejemplos:

Ingreso del carácter a:

AAAA

A A

AAAA

A A

A A

Ingreso del carácter e:

EEEE

E

EEEE

E

EEEE

Extender el programa para que la vocal se exhiba más grande ingresando un factor de multiplicación. Por ejemplo si se ingresa un 2 debe dibujarse el doble de grande la letra en pantalla.

5.2.14 Se ingresan datos de los empleados de una empresa. Por cada empleado se ingresa:

- Legajo (entero entre 1000 y 5000)
- Sueldo básico (float mayor a 1000)
- Antigüedad en años (mayor o igual a 0)
- Sexo ('M' o 'F')
- Categoría (entero entre 1 a 5)

Por cada empleado ingresado se debe calcular el sueldo final a abonar sabiendo que:

- Las Categorías 2 y 3 tienen \$500 de bonificación.
- La Categoría 4 tiene 10% de bonificación.
- La Categoría 5 tiene 30% de bonificación.
- Si la antigüedad es mayor a 10 años recibe una bonificación del 10% adicional.

Todos los datos ingresados deben ser validados. El ingreso finaliza con un legajo igual a cero.

Informar:

- El sueldo a abonar a cada empleado.
- Cantidad de empleados de más de 10 años de antigüedad.
- El mayor sueldo y el legajo del empleado que cobra dicho sueldo.
- Cantidad de hombres y de mujeres.

5.2.15 El gobierno de la Ciudad de Buenos Aires realiza una encuesta a los estudiantes de los colegios secundarios. Los primeros datos a ingresar son la fecha (día, mes y año) y la cantidad de colegios de la ciudad en que se realizó la encuesta.

De cada colegio se conoce: número de escuela, la cantidad de alumnos inscriptos y la especialidad ('T': técnico, 'N': no técnico).

De cada alumno del colegio que respondió la encuesta se conoce: DNI, fecha de nacimiento (día, mes y año), sexo ('M' o 'F'), opinión con respecto a los conocimientos adquiridos ('B': buenos, 'M': malos). Se identifica el fin de ingreso de datos de los alumnos que respondieron la encuesta por cada colegio con ingreso 0 en el DNI del estudiante.

Validar todos los datos que se ingresan por teclado.

Informar:

- a. Por cada colegio, el porcentaje de alumnos que contestaron la encuesta sobre la cantidad de alumnos inscriptos del colegio.
- b. Cantidad de estudiantes de colegio no técnicos que respondieron la encuesta.

- c. Porcentaje de mujeres de escuelas técnicas sobre la cantidad de estudiantes que respondieron la encuesta.
- d. Porcentaje de estudiantes que respondieron que los conocimientos adquiridos eran malos, sobre el total de encuestados.
- e. Mayor edad de los estudiantes encuestados.
- f. Número del colegio con mayor opinión buena sobre los conocimientos adquiridos.

Unidad 6 - Funciones

- 6.1. 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.

- 6.2. 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:

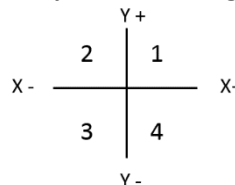
- EstaDentroDelRango: que reciba 3 enteros correspondientes a un número a validar y los límites superior e inferior del rango. La función debe retornar un 1 si el número a validar se encuentra dentro del rango indicado o un 0 si no lo está.
- LeerYValidar: que reciba los límites superior e inferior de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

- 6.3. 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 de disparos en cada eje 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:

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

- 6.4. 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:
- Cuántos números son primos.
 - Cantidad de números pares.
 - 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 primo 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.

- 6.5. 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.

- 6.6. 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.

- 6.7. 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.
- 6.8. 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).
- 6.9. 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.
- 6.10. 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).
- 6.11. 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).
- 6.12. 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).
- 6.13. 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).
- 6.14. 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.

Unidad 7 - Arreglos

7.1 Vectores

- 7.1.1 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.
- 7.1.2 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.
- 7.1.3 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.
- 7.1.4 Construir una función que reciba dos vectores enteros de igual tamaño y retorne un 1 si son iguales o un 0 sino lo son.
- 7.1.5 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 solicitadasSe 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.

7.1.6 Ídem 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:

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

7.1.7 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:

- Cargar los datos de los empleados mediante la función CargaEmpleados. La carga finaliza con un empleado con DNI 99.
- Generar un vector con la cantidad de empleados en cada sector mediante la función CuentaPorSector y mostrarlo en forma de listado.
- Determinar el importe total a pagar de sueldos mediante la función TotalAPagar.
- 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.

7.1.8 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:

- Listado de alumnos con su correspondiente DNI y la nota obtenida (en forma de listado)
- La máxima nota obtenida y el DNI de los alumnos que la obtuvieron.

7.1.9 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:

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

CANT	UNIDADES	CÓDIGO
XXX		XXX
XXX		XXX

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

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

- Generar el vector VA con la información ingresada por teclado.
- 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)
- Calcular el vector VC como VA+VB.
- Copiar el vector VC en orden inverso en otro vector llamado VD.
- Hallar el valor máximo del vector VA y en qué posición se halló (el mayor es único).
- Hallar el valor mínimo del vector VB y en qué posiciones se hallaron (el mínimo se puede repetir).
- Dado un valor ingresado por teclado, indicar en qué posición se encontró del vector VC.
- Ordenar el vector VD en forma ascendente (de menor a mayor).
- Generar un vector VE con aquellos valores negativos que se encuentran en el vector VA.
- Mostrar el contenido de todos los vectores.

FUNCIONES:

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

7.1.11 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 0 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 contenga 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 contenga los códigos de artículos de aquellos que están en los depósitos entre el 15 y 50.

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.

7.1.12 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.

7.1.13 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.

7.1.14 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:

- La cantidad de cada tipo de mueble fabricado durante el mes.
- El tipo de mueble del cual se ha fabricado la mayor cantidad. (puede haber más de uno).
- El listado ordenado en forma ascendente por cantidad de muebles fabricados por operario (sin importar el tipo), indicando:

Código de Operario
XX

Cantidad de muebles fabricados
XXX

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

7.1.15 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:

- Informar el stock de cada código de producto, indicando el código de producto y el stock del mismo.
- Informar el código del producto que tiene el stock más alto.
- Informar el código del producto de aquellos que hayan superado el stock promedio.

FUNCIONES:

- Para controlar los datos ingresados por teclado (la función debe controlar un solo dato).
- Para informar el punto a).

7.1.16 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:
 - Legajo (entero, mayor a cero).
 - Sección (carácter, desde 'A' hasta 'H').
 - 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:

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

Sección	Sueldos al: XX/XX/XXXX	Total de sueldos por sección
H		XXXXXX, XX
B		XXXXXX, XX
...	
TOTAL SUELDOS		XXXXXXXX, XX

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

7.1.17 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      Importe vendido      Cantidad de facturas
Vendedor
  xx          XXXXX, XX           xx
  xx          XXXXX, XX           xx
  ...          .....
  xx          XXXXX, XX           xx

TOTAL        XXXXXXXX, XX       xxx
  
```

Desarrollar 2 programas:

- Con código de vendedor numérico correlativo de 1 a 12.
- Con código de vendedor numérico de 3 cifras, no correlativos.

7.1.18 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:

- Según los parciales si promocionó, aprobó, reprobó o estuvo ausente (uno o ambos parciales igual a cero).
- Listar los alumnos que no cumplen con la asistencia (mayor o igual a 75%).
- Informar cuántos alumnos que promocionaron NO cumplen en la asistencia.
- Informar los alumnos que sacaron 10 en el parcial 2.
- Informar los alumnos con menor asistencia (puede haber varios).
- Al finalizar, informar el promedio total de notas de cada parcial y de asistencia.

7.2 Matrices

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

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

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

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

7.2.5 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:

- la cantidad de asientos disponibles y la cantidad de asientos reservados.
- los números de filas que quedaron vacías.
- la o las filas con mayor cantidad de espectadores.
- un listado con la cantidad de personas que se sentaron en los mismos números de butacas de todos los cines ordenado de mayor a menor. Por ejemplo:

Butaca	Cantidad
1	20
3	15
2	10
.....	

7.2.6 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.

7.2.7 Una empresa desea registrar las ventas realizadas de cada uno de sus 20 productos a lo largo 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:

- El detalle de unidades vendidas de cada producto en cada uno de los 12 meses.
- El producto con el cual se obtuvo la mayor recaudación.
- El trimestre de menor recaudación del año.

7.2.8 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		
	DEPARTAMENTOS DEUDORES		
PISO	DPTO 1	DPTO2 ...	DPTO 15
PISO 1	X		
PISO 2		X	X
PISO 3	X		
PISO 4		X	X
PISO 5	X	X	

7.2.9 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.

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

- La suma de los elementos de las columnas.
- Las sumas de los elementos de las filas.
- La suma total de los elementos.
- El valor promedio de toda la matriz.
- El valor promedio por columna.
- El máximo valor de la matriz y en qué posición se encuentra (el máximo es único).
- 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.

7.2.11 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 9, 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:

- La cantidad de pedidos por sucursal.
- Cuál fue la sucursal que pidió la menor cantidad en total (puede haber varias sucursales que tienen el mismo valor mínimo).

FUNCIONES

- Para inicializar la matriz.
- Para mostrar el resultado del punto b.

7.2.12 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.

7.2.13 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.

7.2.14 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 blanco 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.

7.2.15 BATAALLA 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:

- 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).
- 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.

7.2.16 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 elige 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”.

- Chequear si se ganó el juego teniendo en cuenta si se forma 3 en línea solo en forma vertical.
- Chequear si se ganó el juego teniendo en cuenta si se forma 3 en línea tanto en forma vertical como horizontal.
- Chequear si se ganó el juego teniendo en cuenta si se forma 3 en cualquier dirección (vertical, horizontal, o diagonal).

Unidad 8 - Strings

- Ingresar una frase de hasta 500 caracteres y contar cuántas palabras contiene dicha frase.
 - Considerar que las palabras están separadas por un único espacio.
 - Considerar que las palabras pueden estar separadas por más de un espacio consecutivo.
- 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.
- 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:
 - 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.
 - Mostrar el listado de alumnos ordenado alfabéticamente de menor a mayor.

- 8.4. 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:
- Calcular la recaudación total del día y el producto del cual se vendió menor cantidad de unidades.
 - Mostrar el listado de productos con su precio ordenado en forma alfabética por código de producto.
- 8.5. 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 alquileres en dólares (real, mayor que 0)

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

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

ALQUILER DE AUTOS DEL DÍA: DD-MM-AAAA			
COTIZACION DEL DÓLAR: \$ XX,XX			
NRO. DE AUTO	DIAS DE ALQUILER	PRECIO DEL ALQUILER EN PESOS	FECHA DE DEVOLUCION DEL AUTO
XXXXX	XX	XXXX,XX	XX/XX/XXXX
XXXXX	XX	XXXX,XX	XX/XX/XXXX
TOTAL RECAUDADO X ALQUILER (EN PESOS)		\$ XXXXX,	
TOTAL RECAUDADO X ALQUILER (EN DÓLARES)		\$ XXXXX,XX	

- 8.6. 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. Se 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.

Unidad 9 - Estructuras

- 9.1. 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:

- Actualizar la información de los productos con las ventas realizadas en el mes.
- Al finalizar, mostrar el listado de productos actualizado, informando:

DESCRIPCION	CANTIDAD UNIDADES VENDIDAS	IMPORTE TOTAL VENDIDO
XXXXXXXX	XXXX	\$XXXXX, XX

9.2. 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:

- Declarar un tipo de dato que contenga la información del medicamento.
- Declarar un vector de estructura de tipo de datos creado en a.
- Cargar en un sector de estructuras, los datos referentes a los medicamentos. Función INGRESO.
- Consultar el precio según código del medicamento. Función BUSQUEDA_MEDI.
- Informar los códigos de los medicamentos cuyo stock es inferior a 10 unidades. Función INFORME.

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

- Número de DNI (entero).
- Apellido y Nombre (80 caracteres).
- Nota1, Nota2 (entero).
- Nota Promedio (real, calculado según Nota1 y Nota2).

Se solicita:

- Declarar un tipo de dato que contenga la información del alumno.
- Declarar un vector de estructuras del tipo de dato creado en el punto a.
- 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.
- Indicar cuántos alumnos aprobaron (ambos parciales con nota ≥ 4 y cuántos reprobaron la materia. Función RESULTADO.
- Informar los datos de los alumnos de (DNI – Apellido y Nombre – Nota Promedio) de los alumnos. PROMOCIONADOS (ambas notas ≥ 7). Función INFORME_PROMO.

9.4. 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:

- Determinar la cantidad de ventas realizadas por cliente.
- La cantidad de ventas realizadas por vendedor.
- Informar en forma ordenada por total facturado (modo descendente), el total facturado a cada cliente, informando:

CODIGO DE CLIENTE	NOMBRE Y APELLIDO	TOTAL FACTURADO
X	XXXXX XXXXXXXX	\$ XXXXXXXXXX,XX

9.5. 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:

- Con datos ingresados por teclado: Crear una Estructura de Datos con los datos enunciados anteriormente y llamarla **PERSONA**.
- Mostrar los datos de la Estructura de Datos del punto a.
- 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**.
- Mostrar los datos de la Estructura de Datos del punto c.
- Crear una FUNCION que cargue los datos de la Estructurad de Datos del punto c y que retorne los datos al programa principal (main).
- 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**.
- Crear una FUNCION que realice el punto f.
- Mostrar los datos de la Lista creada en el punto f.
- Crear una FUNCION que realice el punto h.

Unidad 10 - Archivos

10.1. 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)
- Declarar un tipo de dato que contenga la información del alumno.
 - Con la información indicada para los alumnos. Grabar los datos en el archivo "ALUMNOS.dat". Esta información de grabación finaliza con DNI cero.
 - Leer los datos del archivo, mediante la **Función LECTURA**.

- 10.2. 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.

- 10.3. 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 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.

- 10.4. 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 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.

- 10.5. Se dispone de un archivo llamado 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.

10.6. 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)

10.7. 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 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 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.

10.8. Se dispone de un archivo denominado 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.

10.9. 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:

Código Artículo	Descripción	Ene.	Feb.	...	Oct.	Nov.	Dic.
XXX	XXXXXXXXXX	XX	XX	...	XX	XX	XX
XXX	XXXXXXXXXX	XX	XX	...	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.

10.10. 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 REALIZÓ UNA COMPRA
XXX	XXX, 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.

10.11. 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:

- El importe total recaudado en el año.
- Las deudas de cada contribuyente según el siguiente formato de impresión:

CONTRIBUYENTE	DEUDAS REGISTRADAS EN EL AÑO XXX						IMPORTE ADEUDADO	
XXXXXX	BIMESTRES	1	2	3	4	5	6	\$ XXXXXX, XX
			X				X	

10.12. 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:

- Leer el archivo “VENTAS.dat” y los cargue en la matriz TOTAL de 15 por 30 elementos.
- Ingresar por teclado, la fecha de proceso (día, mes, año).
- Procesar las ventas del día.
- Actualizar los acumulados de la matriz TOTAL y al finalizar actualizar el archivo “VENTAS.dat”.
- Imprimir un listado con el siguiente diseño:

VENTAS DEL XX/XX/XX	
Código de medicamento	Importe de la Facturación
XX	\$ XXXXXX, XX
-----	-----
TOTAL VENDIDO	\$ XXXXXX, XX

- Al finalizar el proceso informe el número de Visitador Médico que en el día vendió el mayor importe. (Total Facturado).

10.13. Se desean procesar las transacciones bancarias de las Cuentas Corrientes del BANCO NACIÓN. Se solicita:

- 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 -)
- 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.
- 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)
- 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.
- Listar las Cuentas Nuevas, con el siguiente diseño:

CUENTAS NUEVAS DEL XX/XX/XXXX		
Número de Cuenta	Crédito Máximo	Saldo
xxxxxxx	xxxxxxx, xx	xxxxxxx, xx
xxxxxxx	xxxxxxx, xx	xxxxxxx, xx

10.14. 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:

- Informar el siguiente listado, con "x" el mes o meses que NO pagó.

Apellido y Nombre	Mes 1	Mes 2 ...	Mes 12
xxxxxxx, xxxxxxxx	x	x	
xxxxxxx, xxxxxxxx		x	x

- b. Total pagado por cada propietario, ordenando en forma ascendente por total pagado.

Apellido y Nombre	Total Pagado
xxxxxxx, xxxxxxxx	\$ xxx,xx
xxxxxxx, xxxxxxxx	\$ 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".

10.15. 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 controle 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 **ControlLegajo**. 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	ANTIGÜEDAD	NUEVO SUELDO
XXX	XXXXX,XX	XXX	XXXXX,XX
...
XXX	XXXXX,XX	XXX	XXXXX,XX

10.16. Se dispone de un archivo 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 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
Prod1	9	0	1	4	1
Prod2	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.

10.17. El archivo 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 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	Ene	Feb	Mar	Variación Térmica
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.

10.18. 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).

10.19. 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:

- Cargar la información del archivo **Productos.dat** en un vector de estructuras.
- Cargar la información del archivo **VentasDeposito5.dat** en una matriz.
- Informar el importe total vendido de cada producto por cada comercio con el siguiente formato:

Producto\Comercio	1	2	3	4	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

- Generar el archivo **ImporteDeVentas.dat** la información del punto 3.

10.20. 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:

- Los códigos de las localidades que se encuentran dentro de la zona que ha recibido la menor cantidad de encomiendas.
- La distribución del total de kilos de encomiendas por zona de acuerdo al siguiente formato:

ZONA	TOTAL DE KILOS
1	XXXX
...	XXXX
9	XXXX

- Las provincias que no han recibido encomiendas.

10.21. 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.

10.22. 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.

Unidad 11 - Corte de Control

11.1. Se dispone de un archivo llamado 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.

11.2. Se dispone de un archivo con las ventas del mes de cada una de las sucursales de una empresa. 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 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.

11.3. Se dispone de un archivo con nombre 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 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 nombre `sector.dat` y debe contener:
- Duración en segundos (entero)
 - Tipo de llamada (entero)
 - Costo de la llamada (float)

11.4. 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 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` 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:

- Ingresar por teclado el mes a procesar y recuperar las tarifas correspondientes
- Mostrar el importe total a cobrar por la empresa
- Generar un archivo que contenga la cantidad de viajes y la recaudación de cada interno en el mes.
- Informar el interno que realizó menos cantidad de viajes
- 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 d.
- `MostrarViajes`: para mostrar en forma de listado los viajes ordenados del punto d.

11.5. 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 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:

- Cantidad de espectadores por cada evento.
- 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.

11.6. 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 cercana de forma de contactar a los interesados. Cada archivo generado será nombrado con Nombre de la Localidad.dat.

11.7. 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 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.

11.8. 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:

•

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	MES 1	MES 2	MES 12
XX	XXXXXX	XXXXXXX	XXXXXXXXX

- 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).

11.9. 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 o 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 MESES ADEUDADOS	DOMICILIO
DFE591	1 5 10 11 12	San Martín 1067

- b. Indicar cuál es la categoría de vehículo que recibió la menor cantidad de importe recaudado

11.10. 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.

11.11. 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.**

- 11.12. 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 guardarlo 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)