



Universidad Nacional de la Matanza



Guía de Ejercicios Prácticos

Unidades 2 a 7

INDICE

Tabla de contenido

UNIDAD 2 - ESTRUCTURA SECUENCIAL.....	2
UNIDAD 4 - ESTRUCTURA DE DECISIÓN.....	4
UNIDAD 5 - ESTRUCTURA DE ITERACIÓN O REPETICIÓN	9
5.1 - ESTRUCTURA DE ITERACIÓN INDEFINIDA (WHILE / DO-WHILE)	9
5.2 - ESTRUCTURA DE ITERACIÓN DEFINIDA (FOR)	16
PRÁCTICA PARA EL PRIMER PARCIAL	21
UNIDAD 6 - FUNCIONES.....	25
UNIDAD 7 – VECTORES, ARREGLOS O ARRAYS	34
PRÁCTICA PARA EL SEGUNDO PARCIAL.....	47

Unidad 2 - Estructura Secuencial

- 2.1 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.
- 2.2 Se ingresan la cantidad de horas trabajadas y el valor por hora de un empleado. Determinar el sueldo.
- 2.3 Se ingresan las notas de dos evaluaciones de un alumno. Determinar la nota promedio.
- 2.4 Confeccionar un programa que ingrese una medida en 'pies' y la exhiba convertida a yardas, pulgadas, cms. y mts.

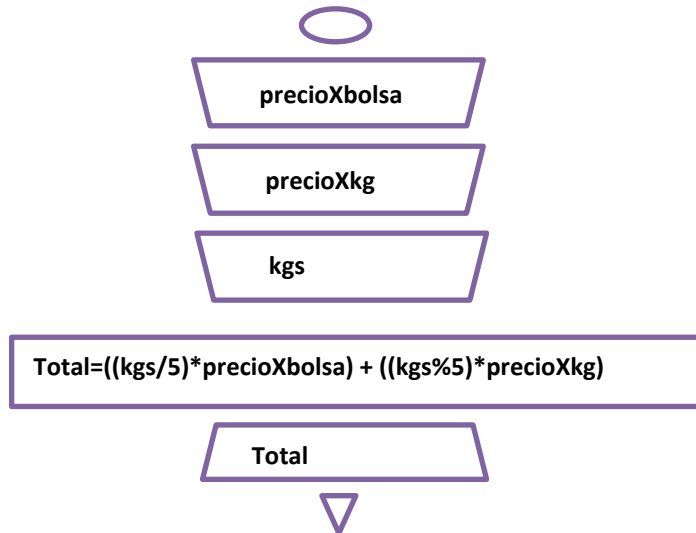
NOTA: 1 pie = 12 pulgadas; 1 yarda = 3 pies; 1 pulgada = 2,54 cms.

- 2.5 En un depósito de una pinturería existen tarros de pintura. Se sabe que el 50% son tarros de 1 litro, el 30% tarros de 4 litros. y el resto tarros de 20 litros. Ingresando como primer dato la cantidad de litros de pintura que hay en existencia, determinar la cantidad de tarros de cada capacidad en litros que hay en stock.
- 2.6 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).
- 2.7 Una empresa de aviación vende pasajes en clase turista y en primera clase, cuyo costo es un 30% más caro que clase turista. Se pide al usuario ingresar el importe de cada pasaje de clase turista, la cantidad de pasajes vendidos de clase turista y la cantidad vendida de primera clase. informar la recaudación total del vuelo.
- 2.8 Una pizzería vende empanadas por unidad o por docena. Como primer paso, se pide al usuario que ingrese el precio por docena y el precio por unidad del día. Si se compran empanadas que no se agrupan en docenas, las adicionales se cobran por unidad. Indicar el precio total a abonar, ingresando la cantidad de empanadas vendidas.
- 2.9 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).
- 2.10 Se ingresa un número entero de 3 cifras. Descomponerlo en unidad, decena y centena.
- 2.11 Se ingresa un número entero que representa una fecha con formato (ddmmaa). Se pide transformarlo a un número con formato (aammdd).
- 2.12 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. Ejemplo: 9929 → 2

¹Contracción 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

Ejercicio resuelto

Una frutería ofrece descuentos por la compra de frutas en bolsas o por kilo. Como primer paso, se solicita al usuario que ingrese el precio de cada bolsa de frutas (que contiene 5 kilos y es más económico) y el precio por kilo individual. Si se compran frutas que excedan el contenido de las bolsas completas, el precio adicional se cobra según el precio por kilo. Calcular el costo total de la compra, ingresando la cantidad de kilos comprados.



```

#include <stdio.h>

int main ()
{
    float precioXbolsa, precioXkg, Total;
    int kgs;

    printf("\nIngrese el precio de la bolsa de 5 kgs de fruta: \t");
    scanf("%f", &precioXbolsa);
    printf("\nIngrese el precio del kilo de fruta: \t");
    scanf("%f", &precioXkg);
    printf("\nIngrese los kilos comprados de fruta (numero entero): \t");
    scanf("%d", &kgs);
    Total=((kgs/5)*precioXbolsa) + ((kgs%5)*precioXkg);
    printf("\n\nEl total a pagar por %d kilos de fruta es: $ %.2f\n", kgs, Total);
    getchar();
    return 0;
}
  
```

Unidad 4 - Estructura de Decisión

- 4.1** Se ingresan 3 números enteros 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** Se ingresan 3 números enteros. Informarlos en orden creciente.
- 4.5** 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.
- 4.6** 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.
- 4.7** 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 del 20% extra si trabajo más de 50 horas y, además, si trabajó más de 150 horas se le otorga un 10% adicional.
- 4.8** La farmacia Sindical efectúa descuentos a sus afiliados según el importe de la compra con la siguiente escala:
- a. Si es menor de \$ 9550 el descuento es del 4.5%
 - b. Si está entre \$ 9550 y \$15000 el descuento es del 8%
 - c. Si es más de \$15000 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.
- 4.9** Una librería de artículos universitarios vende cuadernos A4 de una reconocida marca según la siguiente escala de precios:
- Por unidad, a \$ 5000 c/u.
 - Por paquete de 10 cuadernos cuesta \$ 40.000.-
 - Por paquete de 30 cuadernos cuesta \$ 100.000.-

Si la compra excede los 50 cuadernos se efectúa además un descuento del 10%.

Se pide que se ingrese una cantidad de cuadernos que se quiere comprar y calcule e informe el importe

a pagar.

4.10 La plataforma on line “TuTele” desea realizar un sistema para calcular el costo de la publicidad de un determinado cliente. Para realizarlo ingresa por teclado los siguientes datos:

- Código de cliente (entero)
- Código de la publicidad (entero)
- Año de la publicidad (entero)
- Importe de la publicidad por cada proyección (real)
- Forma de pago (carácter, ‘E’: Efectivo; ‘C’: Cheque a 60 días)
- Cantidad de proyecciones al día (entero)

Calcular el importe a cobrar del día, según el siguiente esquema:

- Si la publicidad es del año actual (estreno) y el pago es con cheque a 60 días, el importe es un 40% más caro.
- Si la publicidad es de los dos años anteriores, el importe es un 20% más caro.
- Si la publicidad tiene más de tres años, y el pago es en efectivo, el importe es un 10% más barato.

Además, existe la siguiente promoción que se suma a lo anterior, si la cantidad de proyecciones diarias es 20 o más, se descuenta un 15% al importe a cobrar. Cualquier otro caso, no hay recargos ni descuentos. Informar el código del cliente, el código de la publicidad y el importe final a cobrar.

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

- código 1; 10; 100: 1000 pesos la unidad.
- código 2; 22; 222: 700 pesos la unidad. La caja de 10 unidades vale 6500 pesos.
- código 3; 33: 300 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: 100 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

4.12 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	7000
D	14000
F	21000
M	35000

T	45000
---	-------

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.

- 4.13** Confeccionar un programa que permita calcular cuánto hay que pagarle a un trabajador, ingresando su sueldo básico y las horas extras trabajadas. Para el pago de las horas extras se toma en cuenta la categoría del trabajador.

Categoría	Precio de la hora extra
A	\$ 8000
B	\$ 8800
C	\$ 10000
D	\$ 12000
E	\$ 14000

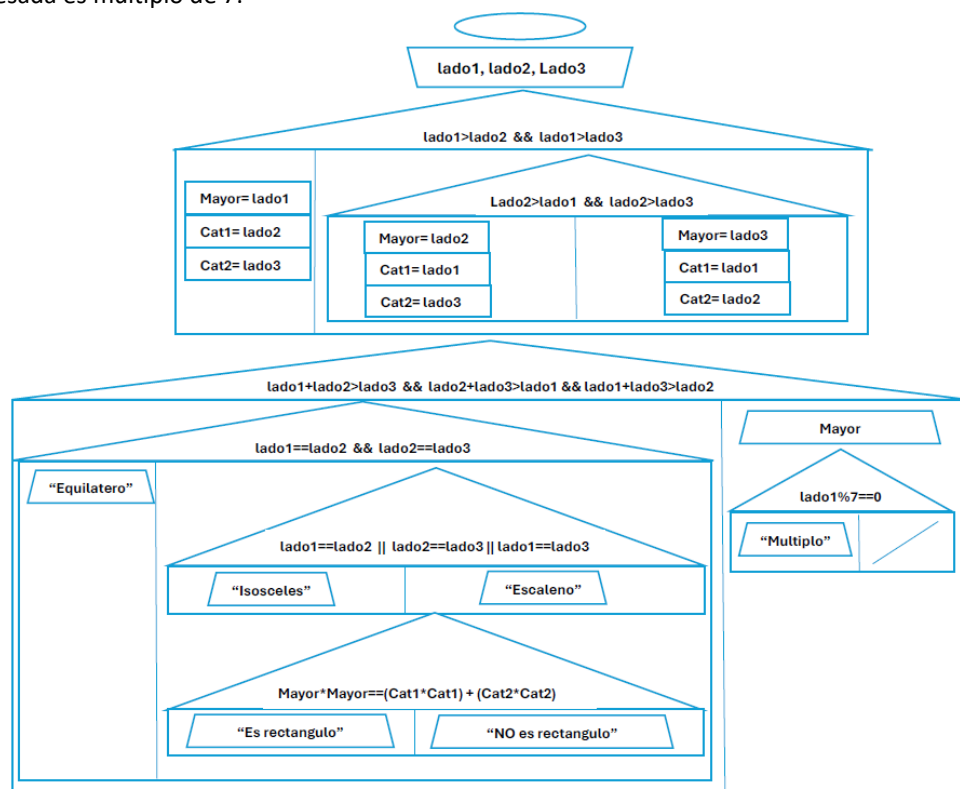
Ejercicio resuelto

Ingresar tres valores enteros correspondientes a longitudes, mayores a cero.

Determinar:

Si las tres longitudes forman un triángulo, informar qué tipo de triángulo es (equilátero, isósceles o escaleno) y si el triángulo es rectángulo.

Si las tres longitudes NO forman triángulo, informar cuál es la mayor longitud ingresada y si la primera longitud ingresada es múltiplo de 7.



```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    int lado1, lado2, lado3, Mayor, Cat1, Cat2;
    printf("\nIngrese la primera longitud - Mayor a cero \t");
    scanf("%d", &lado1);
    printf("\nIngrese la segunda longitud - Mayor a cero \t");
    scanf("%d", &lado2);
    printf("\nIngrese la tercera longitud - Mayor a cero \t");
    scanf("%d", &lado3);
    if(lado1>lado2 && lado1>lado3)
    {
        Mayor=lado1;
        Cat1=lado2;
        Cat2=lado3;
    }
    else
    {
        if(lado2>lado1 && lado2>lado3)
        {
            Mayor=lado2;
            Cat1=lado1;
            Cat2=lado3;
        }
        else
        {
            Mayor=lado3;
            Cat1=lado1;
            Cat2=lado2;
        }
    }
    if(lado1+lado2>lado3 && lado2+lado3>lado1 && lado1+lado3>lado2)
    {
        if(lado1==lado2 && lado2==lado3)
        {
            printf("\nLas longitudes forman un triangulo equilatero\n");
        }
        else
        {
            if(lado1==lado2 || lado2==lado3 || lado1==lado3)
            {
                printf("\nLas longitudes forman un triangulo isosceles\n");
            }
        }
    }
}
```

```
        else
            printf("\nLas longitudes forman un triangulo escaleno\n");
        if(Mayor*Mayor==(Cat1*Cat1)+(Cat2*Cat2))
            printf("\nEl triangulo formado por las tres longitudes forman triangulo rectangulo\n");
        else
            printf("\nEl triangulo formado por las tres longitudes NO forman triangulo rectangulo\n");
    }
else
{
    printf("\nLa maxima longitud ingresada es %d\n", Mayor);
    if(lado1%7==0)
        printf("\nLa primera longitud ingresada es multiplo de 7\n");
    }
    printf("\n\n\n");
    system("pause");
    return 0;
}
```


Unidad 5 - Estructura de Iteración o Repetición

5.1 - Estructura de Iteración Indefinida (while / do-while)

A partir de esta unidad, todos los datos ingresados por teclado DEBEN ser validados, siempre que eso sea posible.

5.1.1 Confeccionar un programa que calcule e informe los valores de las potencias de 2 cuyo resultado sean menores a 6000. (1 2 4 8 16 ...).

5.1.2 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.

5.1.3 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. Por ejemplo: Si se ingresa el número 6, se deben generar los números: 2 , 3 , 5 , 7 , 11 , 13. El algoritmo para el cálculo de si un número es primo o no, debe estar optimizado para que, si encuentra un divisor, no siga buscando ya que se sabe que el número NO es primo.

5.1.4 En un proceso de control científico 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.

5.1.5 Construir un algoritmo que pueda ingresar varios números enteros, distintos de cero, de uno por vez. El ingreso de los datos finaliza al leer un valor nulo (cero).

Informar:

- a) La cantidad de valores positivos ingresados.
- b) El menor valor ingresado con su respectivo número de orden durante el ingreso.
- c) El porcentaje de números pares ingresados.
- d) El máximo valor múltiplo de 7 ingresado con su número de orden.

5.1.6 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:	\$ xxx.xx
Ventas con Tarjeta de Crédito:	\$ xxx.xx
Ventas con cheque:	\$ xxx.xx
Total de Venta:	\$ xxx.xx
Importe del IVA:	\$ xxx.xx

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

5.1.7 Un negocio de perfumería efectúa descuentos en sus ventas según el importe de estas, con la siguiente escala:

- menor a 10000 pesos el 3.5 %
- entre 10001 y 20000 pesos el 10 %
- entre 20001 y 30000 pesos el 20 %.
- Mayor a 30001 pesos el 25%

Confeccionar un programa que:

- Solicite el importe de la venta, calcule el descuento a efectuar e informe el importe neto a cobrar, con mensajes aclaratorios.
- Informe el importe promedio de todas las ventas realizadas.
- Informar cuál es la venta con el importe más grande.
- Informar cuántas ventas fueron inferiores a 10000 pesos.

5.1.8 Se desea realizar un programa para calcular el sueldo final a pagar a los empleados de una empresa. De cada empleado se ingresa:

- Número de empleado (entero, mayor a cero)
- Sueldo básico (real, mayor a cero)
- Antigüedad (entero, de 0 a 25)
- Cantidad de hijos (entero, entre 0 y 10)
- Estudios superiores (carácter, '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 1 ó 2 hijos: aumento del 10%, si tiene más hijos 15%
- Si el empleado posee estudios superiores: aumento del 5%

Al comienzo de la carga se debe preguntar si se desea ingresar un empleado ('S') o no desea ingresar más empleados ('N'), momento en el cual se termina la carga.

Determinar:

- Por cada empleado: número de empleado, el sueldo básico y el nuevo sueldo.
- Informar el total de sueldos a pagar antes y después de calcular los aumentos.
- Informar el porcentaje de incremento de dinero que la empresa debe pagar más a causa de los aumentos.
- Informar el promedio de sueldos pagados a aquellos empleados que tienen más de dos hijos, la antigüedad es mayor a 10 años y tiene estudios superiores.

5.1.9 La estación de servicio “PetroMatanza” imprime tickets por cada venta de los distintos combustibles que despachan durante un día de venta. Por cada ticket se ingresa:

- Número del artículo (entero, entre 40 y 90)
- Número del cliente (entero, mayor a cero)
- Precio por litro (real, mayor que cero)
- Cantidad de litros (real, mayor que cero)

El ticket debe contener además de los datos ingresados y el importe a cobrar. La finalización de la venta de combustible del día se logra con un ticket con número de artículo igual a cero.

Realizar un algoritmo que permita:

- a) Ingrese los datos e imprima el ticket.
- b) Al finalizar el día, informar la cantidad de tickets que se imprimieron y el valor cobrado en total.
- c) Indicar el número de cliente que tiene el ticket más grande en importe.

5.1.10 La empresa “EDENORTE” factura a sus clientes el consumo mensual de electricidad. Solicitar por única vez al comienzo del programa:

- El importe mínimo, para consumos que no excedan los 80 KW (real, mayor a cero).
- El importe de cada KW que exceda los 80 KW, contemplados en el consumo mínimo (real, mayor a cero).

Cuando el consumo es igual a cero KW, se cobra solamente el importe mínimo.

A continuación, ingresar los datos de cada usuario:

- Código del usuario (entero, mayor a cero)
- Estado del medidor anterior (entero, mayor o igual a cero)
- Estado del medidor actual (entero, mayor a cero)

La carga de datos concluye cuando se ingresa el código de usuario cero.

Se pide que se determine e informe:

- a) Para cada usuario, el código y el importe a abonar, es decir, el importe de la factura.
- b) El importe total facturado, la cantidad de facturas emitidas y el promedio facturado en total.

- c) La cantidad de usuarios que no registraron consumo de electricidad.
- d) El código de usuario con el menor consume, entre los que tienen consumo.

5.1.11 Un negocio de perfumería efectúa descuentos en sus ventas según el importe de estas, con la siguiente escala:

- menor a \$ 15000, el 3.5 %
- entre \$ 15001 y \$ 25000, el 10 %
- entre \$ 25001 y \$ 35000, el 20 %
- mayor a \$ 35001, el 25 %

Por cada venta se ingresan los siguientes datos:

- Número de cliente (entero, mayor que cero)
- Importe bruto de ventas (real, mayor a cero)
- Forma de pago (carácter, 'E': efectivo, 'D': tarjeta de débito, 'C': tarjeta de crédito, 'B': billetera virtual)

Adicionalmente, se realiza un descuento del 10% de contado (efectivo) y un recargo del 10% con pago con tarjeta de crédito o billetera virtual. La carga de clientes finaliza con un cliente igual cero.

Confeccionar un programa que:

- a) Informe para cada cliente, el importe bruto, la forma de pago, el descuento/recargo y el valor neto a pagar.
- b) Informe el importe promedio de todas las ventas realizadas.
- c) Calcular el porcentaje de la venta en cada una de las formas de pago respecto del total de ventas.
- d) Indicar cuál fue el mayor importe neto a pagar, pagado de contado.
- e) Indicar cuál fue el menor importe bruto pagado con tarjeta de débito.

5.1.12 El gobierno de la Ciudad de Buenos Aires realiza una encuesta a los estudiantes de los colegios secundarios. Los primeros datos que ingresar son la fecha (enteros, mes entre 1 y 12, y año mayor a 2024) y la cantidad de colegios de la ciudad en que se realizó la encuesta (entero, mayor a cero).

De cada colegio se conoce:

- Número de escuela (entero, mayor a cero)
- La cantidad de alumnos inscriptos (entero, entre 1 y 35)
- La especialidad (carácter, 'T': técnico, 'N': no técnico).

De cada alumno del colegio que respondió la encuesta se conoce:

- DNI (entero, entre 1 y 99999999)
- Edad (entero, entre 10 y 25)
- Sexo (carácter, 'M', 'F' o 'X')

- 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 el ingreso de DNI igual a cero del estudiante.

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) El alumno de mayor edad de los estudiantes encuestados, informando el DNI, el número de escuela y la especialidad que posee.
- f) Número del colegio con mayor opinión buena sobre los conocimientos adquiridos.

Ejercicio resuelto

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 700000)
- Antigüedad en años (entero, mayor o igual a 0)
- Sexo (carácter, 'M', 'F' o 'X')
- Categoría (entero, entre 1 y 5)

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

- Las Categorías 2 y 3 tienen \$ 150000 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.

El ingreso finaliza con un legajo igual a cero.

Informar:

- a) El sueldo a abonar a cada empleado.
- b) Cantidad de empleados de más de 10 años de antigüedad.
- c) El mayor sueldo y el legajo del empleado que cobra dicho sueldo.
- d) Porcentaje de hombres, de mujeres y de no binario de empleados informados.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <ctype.h>
int main ()
{
    int legajo, ant, cat, cant_10=0, max_leg, total=0, contF=0, contM=0, contX=0;
    float sueldo, max;
    char sex;
    bool OK=true, primero=true;
    do{
        if(OK)
        {
            printf("\nIngrese el legajo del empleado - entre 1000 y 5000 - cero para terminar: \t");
            scanf("%d", &legajo);
            OK=false;
        }
        else
        {
            printf("\nERROR - Ingrese el legajo del empleado - entre 1000 y 5000 - cero para terminar: \t");
            scanf("%d", &legajo);
        }
    }while((legajo<1000 || legajo>5000) && legajo != 0);
    while(legajo !=0)
    { total++;
        printf("\nIngrese el sueldo basico del empleado - mayor a cero: \t");
        scanf("%f", &sueldo);
        while(sueldo <= 0)
        { printf("\nERROR - Ingrese el sueldo basico del empleado - mayor a cero: \t");
            scanf("%f", &sueldo);
        }
        printf("\nIngrese la antigüedad del empleado - mayor o igual a cero: \t");
        scanf("%d", &ant);
        while(ant < 0)
        { printf("\nERROR - Ingrese la antigüedad del empleado - mayor o igual a cero: \t");
            scanf("%d", &ant);
        }
        printf("\nIngrese el sexo del empleado - F, M o X: \t");
        fflush(stdin);
        scanf("%c", &sex);
        sex=toupper(sex);
        while(sex != 'F' && sex != 'M' && sex != 'X')
        { printf("\nERROR -Ingrese el sexo del empleado - F, M o X: \t");
            fflush(stdin);
            scanf("%c", &sex);
            sex=toupper(sex);
        }
        OK=true;
        do{
            if(OK)
            { printf("\nIngrese la categoria del empleado - entre 1 y 5 : \t");
                scanf("%d", &cat);
```

```

        OK=false;
    }
    else
    { printf("\nERROR - Ingrese la categoria del empleado - entre 1 y 5 : \t");
      scanf("%d", &cat);
    }
}while(cat<1 || cat>5);
if(cat==2 || cat==3)
    sueldo+=150000;
else
    if(cat==4)
        sueldo+=(sueldo*0.1);
    else
        if(cat==5)
            sueldo+=(sueldo*0.3);
if(ant>10)
    {sueldo+=(sueldo*0.1);
    cant_10++;
    }
printf("\nEl sueldo del empleado de legajo %d es: %.2f\n\n", legajo, sueldo); //Punto a
if(primerio || sueldo>max)
{ max=sueldo;
  max_leg=legajo;
  primerio=false;
}
if(sex=='F')
    contF++;
else
    if(sex=='M')
        contM++;
    else
        contX++;
OK=true;
do{
    if(OK)
    { printf("\nIngrese el legajo del empleado - entre 1000 y 5000 - cero para terminar: \t");
      scanf("%d", &legajo);
      OK=false;
    }
    else
    { printf("\nERROR - Ingrese el legajo del empleado - entre 1000 y 5000 - cero para terminar: \t");
      scanf("%d", &legajo);
    }
}while((legajo<1000 || legajo>5000) && legajo != 0);
}
if(total>0)
{ printf("\nLa cantidad de empleados con mas de 10 anios de antiguedad es: %d\n", cant_10); //Punto b
  printf("\nEl empleado que más sueldo cobro es el del legajo %d con un importe de $ %.2f\n", max_leg,
max); //Punto c
  printf("\nPorcentaje de empleadas mujeres: %.2f", (float)contF/total*100); //Punto d
  printf("\nPorcentaje de empleados hombres: %.2f", (float)contM/total*100);
  printf("\nPorcentaje de empleadxs no binarios: %.2f", (float)contX/total*100);
}

```

```
}  
else  
    printf("\nNo se registraron datos de sueldos de empleado.");  
printf("\n\n\n");  
system("pause");  
return 0;  
}
```

5.2 - Estructura de Iteración Definida (for)

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

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

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

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

5.2.5 Ingresar un número entero positivo N y luego ingresar 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.

5.2.6 Dadas los legajos y las notas de 20 alumnos de un curso, determinar:

- a. La cantidad de alumnos aplazados.
- b. Cuando la nota sea mayor a 7, mostrar el mensaje: "El alumno con legajo xxx ha Promocionado".
- c. El promedio de notas de los alumnos que han sido enviados a final, con por lo menos dos decimales en dicho promedio.

5.2.7 Dadas las edades y estaturas de 45 alumnos de un curso, determinar:

- a. La edad promedio del curso.
- b. La estatura promedio del curso.
- c. La cantidad de alumnos mayores de 10 años.
- d. El porcentaje de alumnos que miden menos de 1.40 cm.

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

- El promedio de números ingresados positivos.
- El promedio de números ingresados negativos.
- La cantidad de ceros ingresada.




Si el valor ingresado N es mayor que 12 o si es menor o igual a 0, informar “CANTIDAD INVALIDA” y a continuación finalizar el programa.

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

NOTA: Hay dos formas de pensar este ejercicio.

La primera (opción 1), es contar todas las ternas que aparecen, aunque repitan los números.

Ejemplo:

5, 4, 5 Una terna

5, 4, 5, 3 Dos ternas

5, 4, 5, 3, 8 Tres ternas


La segunda opción es contar todas las ternas que aparecen, pero **sin repetir** los números.

Ejemplo:

5, 5, 33, 12, 8, 79, 23....


Una Dos
terna ternas

5.2.10 Realizar un programa que:

- Muestre todos los **números primos** entre 1 y 100. Un número es primo cuando es divisible solamente por la unidad y por sí mismo.
- Contar y mostrar la cantidad de primos encontrados.

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

divisores positivos menores que él.

5.2.12 Escriba un programa que pregunte cuántos números se van a introducir, pida esos números, y realice los siguiente:

- a) Informe el mayor valor ingresado y en qué posición se ingresó.
- b) Informar el menor valor ingresado con la posición de ingreso.
- c) Informar la media aritmética. Se recuerda que la media aritmética de un conjunto de valores es la suma de esos valores dividida por la cantidad de valores.
- d) Informar el promedio de los valores negativos.
- e) Informar el porcentaje de valores positivos ingresados respecto del total de números ingresados.

5.2.13 Una empresa les paga a sus 10 empleados con base en las horas trabajadas en la semana. Para esto, se registra el código del empleado (entero) y las horas trabajadas en la semana. Al comienzo de la carga, se ingresa **por única vez** el valor en pesos de la hora.

Determinar:

- a) El sueldo semanal de cada empleado, informando código y sueldo.
- b) El código del empleado que más sueldo cobró en la semana.
- c) El porcentaje de empleados que trabajaron menos de 20 horas en la semana.
- d) El promedio de dinero pagado en sueldos a los empleados durante la semana.

5.2.14 Elaborar un programa que lea 15 números reales e informe:

- a) La sumatoria de los números leídos.
- b) La productoria de los números leídos.
- c) El promedio de los números leídos.
- d) El mayor y el menor número leído.
- e) El porcentaje de números mayores a 1000 que se ingresaron.

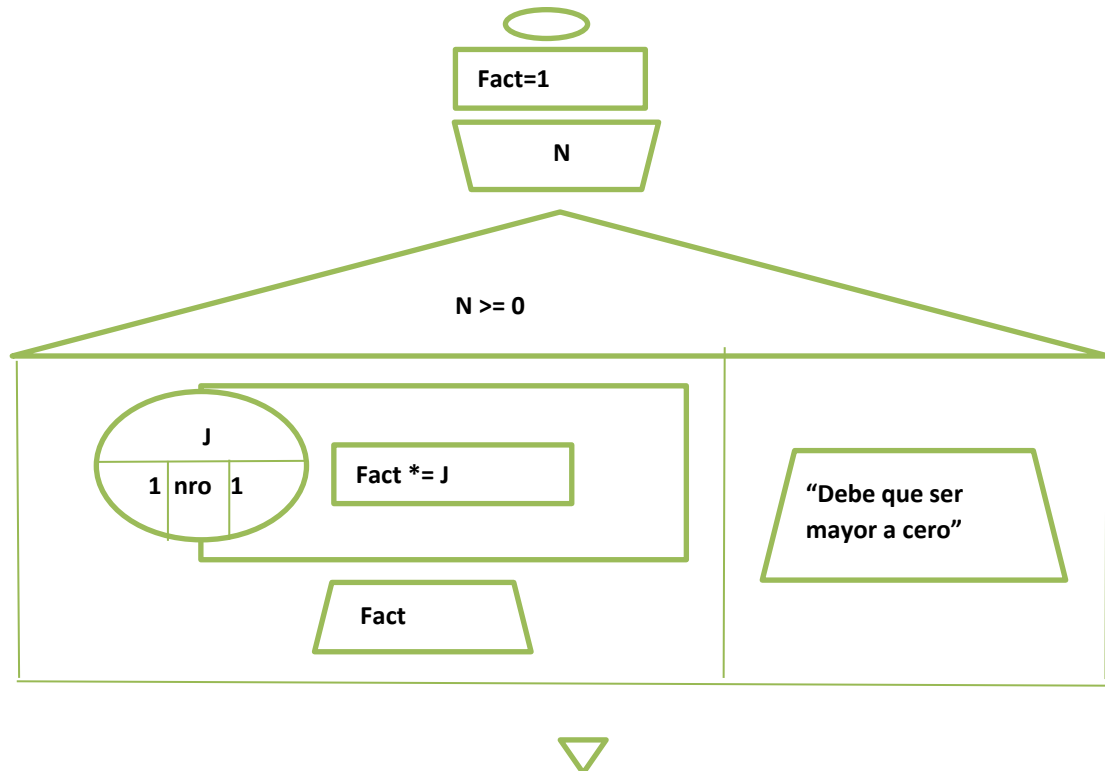
5.2.15 Confeccionar un programa que permita el ingreso de 6 ternas de valores enteros e informe:

- 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, informar el promedio de los menores valores de cada una.

Ejercicio resuelto

Calcular el valor del factorial de un número ingresado por teclado. El valor del factorial de un número 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 valor del factorial de 5 es 120 ya que $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$



```
#include <stdio.h>
#include <stdlib.h>

int main ()
{
    int Fact=1, N, J;

    printf("\nIngrese un número entero mayor o igual a cero\t");
    scanf("%d", &N);
    if (N >= 0)
    {
```

```
    for (J = 1; J <= N; J++)    //Sería lo mismo: J=N;J>=1;J--
        Fact *= J;
    printf ("\nEl factorial del número %d es %d\n", N, Fact);
}
else
    printf("\nEl número debe ser mayor o igual a cero!!!\n");

printf("\n\n");
system("pause");
return 0;
}
```

Práctica para el Primer Parcial

P1.1 Observe el siguiente código:

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int s, j, k, i, l;
6      s=0;
7      for (j=1;j<=5;j++);
8      { printf("%d", j );
9      if (j%2 ==0) { s=s+j; }
10     }
11     printf("\n%d",s);
12     i= 10;
13     while (i>0)
14     i=i-1;
15     printf("\n%d",i);
16     printf("\n\n");
17     system ("pause");
18     return 0;
19 }
```

- ¿Considera que hay errores? De ser así, indique la línea y qué tipo de error es (compilación considerando warnings, lógico o de ejecución).
- Si hubiera errores y fueran solucionados, ¿cuál sería la salida de este programa?
- ¿Por qué se inicializa la variable **S**? Justifique si es necesario y por qué si o no lo es.
- ¿Cambiaría la línea 13 por: **while(i>=0)**. ¿Por qué? ¿Modificaría algo en la ejecución? Justifique

P1.2 Observe los siguientes trozos de código. Suponiendo x=9, indique si ambos códigos son equivalentes (tienen la misma salida).

```

if (x<10)
    x=x+1;

if (x>=10)
    x=x-1;
```

```

if (x<10)
    x++;
else
    x--;
```

P1.3 Observe el siguiente código.

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int x, y;
6      float n;
7
8      scanf("%2f",n);
9      while (n!=0)
10     { if (n%2==0)
11       {
12         y++;
13       }
14       x++;
15       scanf("%2f",n);
16     }
17     printf("\n%d", &x);
18     printf("\n%d", &y);
19     printf("\n\n");
20     return 0;
21 }
```

- Este código contiene errores. Indique las líneas de código donde están, qué tipo de error es y cómo se soluciona.
- Una vez corregidos los errores, indicar cuál es el objetivo del programa (qué hace).
- Cuál sería el resultado final para las variables **x** e **y**, con el siguiente juego de prueba para los ingresos de la variable **n**: 7, 12, -8, 33, 0.
- Indique el valor de verdad de la siguiente afirmación acerca del código: "Es necesario incluir la biblioteca **<ctype.h>** para poder usar la instrucción **system ("pause")**"

P1.4 Sean *P*, *Q* y *R* condiciones lógicas y **Sentencia 1**, **Sentencia 2** y **Sentencia 3** sentencias ejecutables. Determinar para las siguientes estructuras alternativas los valores de verdad (verdadero y falso) de las condiciones lógicas que determinan la ejecución de cada sentencia, completando el cuadro.

```
if (P)
    if (Q && R)
        //Sentencia 1;
    else
        //Sentencia 2;
else
    if (R)
        //Sentencia 3
```

P	Q	R	Sentencia/s ejecutada/s
F	F	F	
F	F	V	
F	V	F	
F	V	V	
V	F	F	
V	F	V	
V	V	F	
V	V	V	

P1.5 Observe el siguiente trozo de código: ¿Qué hace? ¿Necesitaría sentencias adicionales para que funcione correctamente?

```
char car;
scanf("%c", &car);
car=tolower(car);
```

P1.6 Observe el siguiente programa:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n,m,a,s,x;
6
7      scanf("%d%d", &n,&m);
8      while(n!=m && (n!=0 || m!=0))
9      {
10         if(n>m)
11         { a=n;
12           n=m;
13           m=a;
14         }
15         s+=n;
16         x++;
17         scanf("%d%d", &n,&m);
18     }
19     printf("Promedio: %.2f", s/x);
20
21 }
```

- ¿Tiene errores el código? En caso de que así sea, indicar en qué líneas están dichos errores, de qué tipo es cada uno y cómo se arregla.
- ¿Cuál es el objetivo del programa? ¿Qué entrega en la salida?
- ¿Qué valor o valores y de qué variable o variables detendrán la ejecución del **while**?

P1.7 Sean *P*, *Q* y *R* condiciones lógicas y **Sentencia 1**, **Sentencia 2** y **Sentencia 3** sentencias ejecutables. Determinar para las siguientes estructuras alternativas los valores de verdad (verdadero y falso) de las condiciones lógicas que determinan la ejecución de cada sentencia, completando el cuadro.

```
if ((P && Q) || R)
{
    //Sentencia 1;
    if (P)
        //Sentencia 2;
    else
        Sentencia 3;
}
```

P	Q	R	Sentencia/s ejecutada/s
F	F	F	
F	F	V	
F	V	F	
F	V	V	
V	F	F	
V	F	V	
V	V	F	
V	V	V	

Ejemplos de Parcial 1 resueltos

Los siguientes programas tienen errores, indicar la línea en la que están, que tipo de error es y cómo solucionarlo.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x = 5;
6
7      if x > 3
8      {
9          printf("x es mayor que 3\n");
10     }
11     return 0;
12 }
```

Línea 5, 7, 9 y 11 → No está indentado (error de formato).

Línea 7 → Falta paréntesis en la condición del if (error sintáctico).

```
1  #include <stdio.h>
2
3  int main()
4  {
5      for (int i = 0, i < 5, i++)
6          printf("i es: %d\n", &i);
7      return 0;
8  }
```

Línea 4 → No declara las variables (error lógico y de formato)

Línea 5 → Los parámetros del for se separan con ; (error de sintaxis)

Línea 6 → En el printf poner el operador & indica que se pide mostrar la dirección de memoria y no el contenido de i (error lógico)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int contador=0, num, suma;
6      printf("Ingresar números enteros distintos de cero");
7      scanf("%3d", num);
8
9      while (contador != 0)
10         suma += num;
11         contador++;
12         printf("Ingresar número entero");
13         scanf("%3d", num);
14
15     printf("Promedio: %.2f", suma/contador);
16     return 0;
17 }
```

Línea 5 → Se debe inicializar la variable suma en cero ya que es un acumulador (error lógico)

Líneas 7 y 13 → En el scanf no pueden ir formatos. Falta el & en la lectura de num, lo que puede provocar errores de memoria (ambos errores lógicos)

Líneas 10 y 13 → Faltan los {}. Así sólo se procesa dentro del while la línea 10, el resto solamente una vez fuera del while (error lógico)

Línea 15 → Si el primer dato ingresado es cero, da error en la división (error de ejecución). Si el promedio es real, debe castearse la división, sino dará un valor entero (error lógico)

Unidad 6 - Funciones

A partir de esta unidad, todos los datos ingresados por teclado **DEBEN** ser validados con **FUNCIONES**, siempre que eso sea posible.

6.1. Se ingresan números enteros comprendidos entre 100 y 2000. Determinar:

- 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**: Recibe 3 enteros correspondientes a un número a validar y los límites superior e inferior del rango. La función debe retornar un 1 (verdadero) si el número a validar se encuentra dentro del rango indicado o un 0 (falso) si no lo está.
- **LeerYValidar**: Recibe los límites superior e inferior de un rango y retorna 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* analizada en el punto anterior.

6.2. Se ingresa valores positivos entre 1 y 50, finalizando el ingreso con el valor -10. Calcular:

- a. Porcentaje de números son primos.
- b. Cantidad de números pares.
- c. Promedio de los números impares.

Utilizar las siguientes funciones:

- **IngrDatoValCF**: Recibe tres parámetros que son el rango y la condición de fin, que es un número que no pertenece al rango válido. El ingreso se hace dentro de la función y se retorna el dato ingresado y validado.
- **CalculaResto**: Recibe por parámetro el Dividendo y el Divisor, retorna el resto de la división.
- **esPar**: Recibe por parámetro un número, retorna 1 (verdadero) si es Par, 0 (falso) si es Impar. Invoca a la función *CalculaResto*.
- **esPrimo**: Recibe por parámetro un número, retorna 1 (verdadero) si es primo, 0 (falso) si no es primo. Invoca a la función *CalculaResto*. (un número es primo si sólo es divisible por 1 y por sí mismo).
- **Promedio**: Recibe por parámetro una variable acumuladora y una variable contadora. Retorna el promedio si contador es mayor a 0 sino Retorna 0. Se debe preguntar después del retorno el valor del promedio.

6.3. 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:

- Informar para cada terna, el mayor y el menor de los valores.
- Informar el promedio total de los valores mayores de cada terna.
- Indicar cuál es el valor mínimo ingresado tomado en cuenta todas las ternas.

Confeccionar y utilizar una función con parámetros llamada MayorMenor, que permita calcular el menor o el mayor valor de cada terna según un parámetro de configuración.

- 6.4.** Realizar un programa que al comienzo 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:

Si se ingresa 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.

- 6.5.** Confeccionar un programa que permita ingresar varias ternas de valores reales y 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 de la terna
- 2 - Promedio de la terna
- 3 - Suma de la terna
- 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.

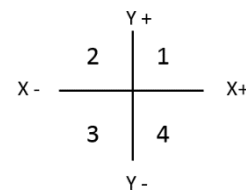
- 6.6. Se realizó un concurso de tiro al blanco. Por cada participante se ingresa un número (entero, mayor a cero) que lo identifica. El ingreso finaliza con un número de participante negativo.

Cada participante efectúa 5 disparos, registrándose las coordenadas X-Y de cada disparo.

- No considerar disparos sobre los ejes, pero sí en el centro (si es sobre los ejes las coordenadas deberán volver a ingresarse).
- Para determinar el cuadrante utilizar la función CUADRANTE que reciba las dos coordenadas y retorne el cuadrante al cual pertenece (1 a 4) y 0 para indicar un tiro en el centro.

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

- ✓ Cuadrantes 1 y 2: 50 puntos
- ✓ Cuadrantes 3 y 4: 40 puntos
- ✓ Centro: 100 puntos



Determinar:

- a. El puntaje obtenido por cada participante, detallando cuantos disparos realizó en cada cuadrante.
 - b. Mostrar el número del participante ganador y el puntaje obtenido.
 - c. Calcular y mostrar la cantidad total de disparos en el centro de todos los participantes.
- 6.7. Realizar un programa que ingrese tres valores enteros correspondiente a una fecha (día, mes y año), valide que la fecha sea correcta y una vez validada informe lo siguiente:
- a. La fecha del día siguiente a la fecha ingresada.
 - b. La fecha del día anterior a la fecha ingresada.
 - c. Pedir al usuario que ingrese un valor N (entero, mayor que cero) e informe la fecha que era hace N días y la fecha que será dentro de N días.
- 6.8. Realizar un programa que ingrese la fecha actual y la fecha de nacimiento de una persona y calcule su edad. Las fechas deben ser validadas con una función. Para el cálculo de la edad realizar una función que reciba las dos fechas y retorne la edad de la persona.

Ejercicio resuelto

Un club de esgrima está desarrollando un sistema para registrar a los participantes de un torneo de esgrima local. Como primer dato y por única vez, ingresar la fecha del día. Luego, por cada participante se ingresan los siguientes datos:

- DNI del participante (entero, entre 1000000 y 99999999)
- Código de Inscripción (entero, de 1001 a 9999)
- Fecha de nacimiento (enteros, día – mes – año)

El código de inscripción es un número de 6 dígitos que el participante elige como clave personal. Por problemas de seguridad y responsabilidad del seguro del club, no se admiten personas menores a 18 años ni mayores a 65 años. Si el participante no cumple con el requisito de la edad, informar “REGISTRACIÓN INVÁLIDA” y pasar al próximo participante. La carga de participantes termina con un DNI igual a cero.

Se pide:

- Informar el DNI de cada participante, el código de inscripción, la edad, y dos métodos de comprobación: la suma de los dígitos del código de inscripción y el producto de los últimos dos dígitos del DNI.
- La cantidad de participantes rechazados por la edad.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <stdbool.h>
```

```
bool esBisiesto(int);
```

```
int CantDiasMes(int, int);
```

```
bool esFechaVal(int, int, int);
```

```
int IngrDatoValCF(int, int, int);
```

```
int CalculoEdad(int, int, int, int, int, int);
```

```
int Suma4Dig(int);
```

```
int Prod2Dig (int);
```

```
int main ()
```

```
{
```

```
    int dni, cod_ins, dhoy, mhoy, anhoy, dnac, mnac, anac, edad, rech=0;
```

```
    bool OK=true, HayPart=false;
```

```
    do{
```

```
if(OK)
{
    printf("\nIngresar dia, mes y anio de hoy: \t");
    scanf("%d%d%d", &dhoy, &mhoy, &anhoy);
    OK=false;
}
else
{printf("\nERROR - Ingresar dia, mes y anio de hoy: \t");
    scanf("%d%d%d", &dhoy, &mhoy, &anhoy);
}
}while(!esFechaVal(dhoy, mhoy, anhoy));

printf("\nIngresar DNI del participar - entre 1000000 y 99999999 - cero para terminar: \t");
dni=IngrDatoValCF(1000000, 99999999, 0);
while(dni != 0)
{
    HayPart=true;
    printf("\nIngresar codigo de inscripcion - entre 1001 y 9999: \t");
    cod_ins=IngrDatoValCF(1001, 9999, 9999);
    OK=true;
    do{
        if(OK)
        {
            printf("\nIngresar dia, mes y anio de nacimiento: \t");
            scanf("%d%d%d", &dnac, &mnac, &anac);
            OK=false;
        }
        else
        {
            printf("\nERROR - Ingresar dia, mes y anio de nacimiento: \t");
            scanf("%d%d%d", &dnac, &mnac, &anac);
        }
    }

    }while(!esFechaVal(dnac, mnac, anac));

    edad=CalculoEdad(dhoy, mhoy, anhoy, dnac, mnac, anac);
```

```
if(edad>=18 && edad<=65)
{
    printf("\nParticipante DNI: %d\nCodigo de inscripcion: %d\tEdad: %d", dni, cod_ins, edad);
    printf("\nSuma de digitos codigo inscripcion: %d", Suma4Dig(cod_ins));
    printf("\nProducto dos ultimos digitos DNI: %d", Prod2Dig(dni));
}
else
{
    printf("\nREGISTRACION INVALIDA\n");
    rech++;
}

printf("\n\nIngresar DNI del participar - entre 1000000 y 99999999 - cero para terminar: \t");
dni=IngrDatoValCF(1000000, 99999999, 0);
}

if(HayPart)
    printf("\nLa cantidad de participantes rechazados por la edad es: %d", rech);
else
    printf("\nNo inscripción de participantes");

printf("\n\n\n");
system("pause");
return 0;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int IngrDatoValCF(int li, int ls, int cf)
{
    int dato;
    bool OK=true;

    do{
        if(OK)
        {
            scanf("%d", &dato);
            OK=false;
        }
        else
    }
```

```
{
    printf("\nERROR - Ingrese un valor entre %d y %d o %d: \t", li, ls, cf);
    scanf("%d", &dato);
}
}while((dato<li || dato>ls) && dato != cf);

return dato;
}
/////////////////////////////////////////////////////////////////
bool esFechaVal(int dia, int mes, int anio)    //Definición de función
{
    bool fv=false;

    if (anio > 1900)
        if (mes >= 1 && mes <= 12)
            if (dia >= 1 && dia <= CantDiasMes(mes, anio))
                fv = true;

    return fv;
}
/////////////////////////////////////////////////////////////////
int CantDiasMes(int m, int a)
{
    int cdm = 31;

    if(m == 4 || m == 6 || m == 9 || m == 11)
        cdm = 30;
    else
        if (m == 2)
            cdm = 28 + esBisiesto(a);

    return cdm;
}
/////////////////////////////////////////////////////////////////
bool esBisiesto(int a)
{
```

```
bool bis;

if((a % 4 == 0 && a % 100 != 0) || a % 400 == 0)
    bis=true;
else
    bis=false;

return bis;
}

/////////////////////////////////////////////////////////////////
int CalculoEdad(int dh, int mh, int ah, int dn, int mn, int an)
{
    int ed;

    ed=ah-an;
    if(mh>mn)
        ed++;
    if(mh==mn)
        if(dh>=dn)
            ed++;
    return ed;
}

/////////////////////////////////////////////////////////////////
int Suma4Dig(int num)
{
    int aux, d1, d2, d3,d4;

    d4=num%10;
    aux=num/10;
    d3=aux%10;
    aux=aux/10;
    d2=aux%10;
    d1=aux/10;

    return (d1+d2+d3+d4);
}
```


////////////////////////////////////

```
int Prod2Dig (int num)
```

```
{
```

```
    int aux, d1, d2;
```

```
    d2=num%10;
```

```
    aux=num/10;
```

```
    d1=aux%10;
```

```
    return (d1*d2);
```

```
}
```

Unidad 7 – Vectores, arreglos o arrays

A partir de esta unidad, todos los ejercicios DEBEN ser resueltos utilizando FUNCIONES y validando todos los datos ingresados por teclado, siempre que eso sea posible.

7.1 Dada una lista de 10 valores enteros. Se pide:

- a) Cargar la lista en memoria en un vector llamado VecA.
- b) Copiar el vector VecA en otro llamado VecB.
- c) Generar un vector VecC correspondiente a la suma de VecA y VecB.
- d) Copiar VecA en orden inverso en otro vector llamado VecD.
- e) Listar los cuatro vectores simultáneamente informando en la primera columna el número de orden de los elementos.
- f) Informar las posiciones de elementos pares del vector VecA.
- g) Informar los elementos de posiciones impares del vector VecA.
- h) Informar la suma de los elementos del vector VecA.
- i) Informar el valor promedio de los elementos del vector VecA.
- j) Ingresar por teclado un valor entero N e informar la cantidad de elementos iguales a N en el vector VecA.

7.2 Para una carrera de automóviles, se toman los tiempos de clasificación (real, mayor que 0 en segundos) 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:

- El número de auto que clasificó en primer lugar.
- El número de auto que obtuvo el peor tiempo de clasificación.
- La cantidad de autos que superaron el promedio de tiempo de clasificación.
- El o los números de autos que no participan de la carrera actual.

7.3 Dados los legajos y sueldos de los empleados de una empresa (máximo 50 empleados), se pide determinar el / los empleado/s de máximo sueldo. El fin de carga de sueldos está dado por un legajo igual a cero.

- Nota 1: Los números de legajo son números correlativos de 1 a 50, pero no se sabe si están los 50 empleados.

- Nota 2: Realizar el mismo ejercicio considerando que los números de legajo son números NO correlativos de 4 cifras. No se evalúa si se repite algún legajo.

7.4 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 (entero, de 10001 a 10010)
- Cantidad de unidades solicitadas (entero, mayor a cero)
- Precio unitario del producto (real, mayor a cero)

Se puede recibir más de un pedido por producto. Como se piden desde distintos lugares del país, el precio unitario puede variar de un pedido a otro aunque sea del mismo producto.

La carga finaliza cuando se ingresa un producto con código igual a cero.

Al finalizar, informar:

- Un listado con código y cantidad de unidades solicitadas de cada producto.
- Indicar el porcentaje de productos que se pidieron en más de 1000 unidades.
- Informar qué producto vendió el menor importe (puede haber más de uno).

7.5 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 de 101 a 112).
- 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 de facturación por cada **vendedor** y Cantidad de facturas emitidas **por vendedor**, con el siguiente formato:

Venta del Día: XX/XX/XXXX

Nro. de Vendedor	Importe vendido	Cantidad de facturas emitidas
xxx	XXXXXX, XX	xx
xxx	XXXXXX, XX	xx
...
xxx	XXXXXX, XX	xx
TOTAL	XXXXXXXX, XX	xxx

- El porcentaje de vendedores que no vendieron nada durante el día.
- El vendedor que tuvo la mayor cantidad de facturas durante el día.
- El promedio de ventas por factura realizada en el día.

7.6 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 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, en el mismo día o distintos días) durante el mes.

Determinar e informar:

- a) La cantidad de cada tipo de mueble fabricado durante el mes.
- b) Tipo de mueble del cual se ha fabricado la mayor cantidad. (puede haber más de uno).
- c) Listado de cantidad de muebles fabricados por operario (sin importar el tipo), indicando:

<u>Código de operario</u>	<u>Cantidad de muebles fabricada</u>
---------------------------	--------------------------------------

- d) Día o días hábil/es en los cuales no hubo nada de producción.

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

- Legajo (entero, mayor a cero)
- Sección (carácter, desde 'A' hasta 'H')
- Sueldo (real, mayor a cero)

El fin de datos es cuando aparece un legajo igual a cero.

Se pide:

- a. Emitir un informe que indique por sección, el total de sueldos de la misma.

SECCIÓN	TOTAL DE SUELDOS POR SECCIÓN
A	XXXXX.XX
B	XXXXX.XX
....
 TOTAL SUELDOS	 XXXXXX.XX

- b. El legajo del empleado con mayor sueldo consignando legajo, sección y sueldo.
- c. El promedio de sueldo por empleado.

7.8 Se ingresan DNI (entero, entre 1000000 y 99999999), código de sector (entero, 1 a 5) y sueldo de los empleados (real, mayor a cero) de una empresa. Se sabe que como máximo la empresa tiene 100 empleados. Se desea:

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

7.9 Una empresa de venta de artículos de limpieza trabaja con 300 artículos diferentes. Al comenzar el programa se ingresa el estado del stock hasta el momento. De cada artículo se ingresa los siguientes datos:

- Código de artículo (entero, mayor a cero).
- Número de depósito donde se encuentra el artículo (entero, entre 1 y 20).
- Stock (entero, entre 0 y 2000).
- Punto de reposición (entero, entre 500 y 1000).

Luego se procesan los movimientos del mes en el stock: egresos (resta al stock) e ingresos (suma al stock).
Por cada movimiento se ingresa:

- Código del artículo
- Código del movimiento (carácter, 'I': ingreso a stock - suma; 'E': egreso de stock - resta)
- Cantidad ingresada / egresada (entero, mayor a cero)

La carga de movimientos finaliza con un código de artículo igual a cero.

Se pide informar lo siguiente:

- a. El detalle de la cantidad de artículos en stock, informando código y existencia.
- b. El porcentaje de artículos que hayan quedado con el stock igual a cero.
- c. Los códigos de artículos que hayan quedado con stock por debajo del punto de reposición.
- d. Cuántos depósitos tuvieron movimientos de egreso durante el mes.

NOTA: Se permite que el stock pueda quedar negativo, de artículos vendidos pero pendiente de ingreso al stock.

7.10 Una empresa de alquiler de autos tiene una flota de 30 autos de alta gama, identificados cada uno con código. Como primera acción, se ingresan por teclado los siguientes datos:

- Código del auto (entero, mayor a cero)
- Precio diario del alquiler en dólares (real, mayor que cero)

Al comenzar el procesamiento de los alquileres, se ingresa la fecha del día y la cotización del dólar de ese día. A continuación, se ingresan los siguientes datos correspondiente a cada alquiler realizado en el día:

- Código del auto
- Cantidad de días de alquiler (entero, mayor que cero)

Para finalizar la carga del día, se ingresa un código de auto igual a cero.

Determinar e informar:

- Porcentaje de vehículos alquilados durante el día.
- Informar la siguiente planilla:

ALQUILERES DE AUTOS DEL DÍA XX / XX / XXXX

COTIZACIÓN DEL DÓLAR: \$ XXXX.XX

<u>CÓDIGO VEHÍCULO</u>	<u>DÍAS DE ALQUILER</u>	<u>PRECIO DEL ALQUILER EN PESOS</u>	<u>FECHA DE DEVOLUCIÓN DEL VEHÍCULO</u>
XXXXXX	XX	XXXX.XX	XX / XX / XXXX
XXXXXX	XX	XXXX.XX	XX / XX / XXXX
...
XXXXXX	XX	XXXX.XX	XX / XX / XXXX
<u>TOTAL RECAUDADO EN ALQUILERES (EN PESOS)</u>			XXXXX.XX
<u>TOTAL RECAUDADO EN ALQUILERES (EN DÓLARES)</u>			XXXX.XX

7.11 La empresa de turismo *CIVITATIS* desea analizar las ventas de los 80 paquetes turísticos que comercializa, efectuada durante el período de vacaciones.

Para ello, en un primer lote de datos, por cada paquete turístico, se ingresa:

- ✓ Código de paquete turístico (números enteros positivos no consecutivos de 4 cifras)
- ✓ Precio unitario (real, mayor a cero)

A continuación, ingresar la fecha del día de proceso (día, mes, año).

Luego, procesar los datos de las ventas del período, que se ingresan por teclado, con el siguiente detalle:

- ✓ Número de cliente (entero, mayor a cero)
- ✓ Código de paquete turístico
- ✓ Cantidad vendida (entero, mayor a cero)

Estos datos no se ingresan ordenados, los números de clientes **no** se repiten y la lectura debe detenerse cuando se ingrese un número de cliente igual a cero.

- Acumular la cantidad vendida por cada paquete turístico. Si se ingresa un código de paquete turístico no ofertado, rechazar la venta (los 3 datos) con una leyenda aclaratoria.
- Al finalizar el proceso, informar:
 - Las cantidades e importes vendidos de cada paquete turístico, con el siguiente diseño:

VENTAS AL XX/XX/XXXX		
<u>CÓDIGO DEL PAQUETE TURÍSTICO</u>	<u>CANTIDAD VENDIDA</u>	<u>IMPORTE VENDIDO</u>
XXXX	XXX	XXXXXXXXXX.XX
XXXX	XX	XXXXXXXXXX.XX
.....
<u>TOTALES</u>	XXXX	XXXXXXXXXXXXXX.XX

- El promedio de ventas de los paquetes turísticos por cliente.
- Los códigos de los paquetes turísticos de los que no se han realizado ninguna venta en el período.
- El número de cliente que hizo la menor compra durante el período.

Ejercicio resuelto

Una tienda de mascotas almacena un máximo de 80 productos diferentes en distintos puntos de venta, identificados por códigos (entero,

positivo de 4 cifras), únicos que no son correlativos. Cada producto tiene un precio unitario real (mayor que cero), y una cantidad en existencia

(entero, mayor o igual a cero), que se ingresa junto con el código por única vez al comienzo del programa. Cuando no hay más productos que ingresar

se ingresa producto con código cero.

Luego se reciben los datos de los movimientos realizadas durante la semana, con los siguientes datos:

- Código del producto
- Código del punto de venta (carácter, de 'A' a 'J')
- Cantidad ingresada o egresada (entero, mayor a cero)
- Código del movimiento (entero, 1 para el ingreso - suma; 2 para el egreso - resta)

No pueden egresar por venta los productos de los cuales no haya la totalidad pedida. En ese caso, informar la situación y desechar el movimiento.

En el caso que el producto no exista, informar número de producto y la palabra INEXISTENTE. Los datos finalizan con un producto igual a cero.

Se solicita determinar:

- a) El código del producto con mayor cantidad en inventario al finalizar la operación.

- b) El punto de venta que registró el mayor número de operaciones (ingresos y egresos) en el día (puede haber más de uno).
- c) El importe (en dinero) de cada producto en inventario al finalizar el día.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <stdbool.h>

int LeeYValidaEnt(int);
float LeeYValidaReal(float);
int CargaVecParParcial(int[], int[], float[]);
char IngrDatoValCH(char, char);
int IngrDatoValEnt(int, int);
int Busqueda(int[], int, int);
int BusquedaCH(char[], int, char);
int PosicionMayor(int[], int);
int Maximo(int[], int);
void MostrarParIgualA(int[], char[], int, int);
void MostrarVecParal(int[], int[], float[], int);

int main ()
{
    int VProd[80], VCant[80], VPuntos[10]={0}, cant, prod, cant_mov, opcion, pos, pos2, max, maxi;
    float VPUnt[80];
    char pvta, VPtoV[10]='A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J';

    printf("\nIngresar codigos de producto, cantidad en existencia y precios unitarios\n");
    cant=CargaVecParParcial(VProd, VCant, VPUnt);
    printf("\nIngreso de movimientos semanales\n");
    printf("\nIngresar el codigo del producto - mayor que cero - Cero para terminar: \t");
    prod=LeeYValidaEnt(0);
    while(prod !=0)
    {
        printf("\nIngrese el punto de venta. ");
```



```
pvtA=IngrDatoValCH('A', 'J');
printf("\nIngresar la cantidad ingresada o egresada - mayor que cero : \t");
cant_mov=LeeYValidaEnt(0);
printf("\nIngresar el tipo de movimiento - 1: Ingreso - 2: Egreso: \t");
opcion=IngrDatoValEnt(1, 2);
pos=Busqueda(VProd, cant, prod);
if(pos != -1)
{
    if(opcion==2 && VCant[pos]< cant_mov)
        printf("\nNo se puede procesar el movimiento de egreso porque no hay existencia de productos\n");
    else
    {
        if(opcion==1)
            VCant[pos]+=cant_mov;
        else
            VCant[pos]-=cant_mov;

        pos2=BusquedaCH(VPtoV, 10, pvtA);
        if(pos2 != -1)
            VPuntos[pos2]++;
    }
}
else
    printf("\n%d INEXISTENTE\n", prod);

printf("\nIngresar el codigo del producto - mayor que cero - Cero para terminar: \t");
prod=LeeYValidaEnt(0);
}

maxi=PosicionMayor(VCant, cant);
printf("\nEl producto con mayor cantidad de productos es %d", VProd[maxi]); //Punto a
max=Maximo(VPuntos, 10);
printf("\nLa cantidad maxima de operaciones en puntos de ventas es %d y se realizo en los siguientes puntos de venta:", max);
MostrarParIguales(VPuntos, VPtoV, 10, max); //Punto b
printf("\nPRODUCTO\tIMPORTE EN STOCK\n");
```

```
MostrarVecParal(VProd, VCant, VPUnt, cant); // Punto c

printf("\n\n\n");
system("pause");
return 0;
}

///////////////////////////////// FUNCIONES ///////////////////////////////////

int LeeYValidaEnt(int lim)
{
    int dato;

    scanf("%d", &dato);
    while (dato<lim)
    {
        printf("\nERROR - El dato ingresado tiene que ser mayor o igual a %d: \t", lim);
        scanf("%d", &dato);
    }
    return dato;
}

/////////////////////////////////

float LeeYValidaReal(float lim)
{
    float dato;

    scanf("%f", &dato);
    while (dato<lim)
    {
        printf("\nERROR - El dato ingresado tiene que ser mayor o igual a %d: \t", lim);
        scanf("%f", &dato);
    }
    return dato;
}

/////////////////////////////////

int CargaVecParParcial(int P[], int C[], float I[])
{
    int i=0, prod, cant;
```

```
float precio;

printf("\nIngrese el codigo del producto - mayor a cero - Cero para terminar: \t");
prod=LeeYValidaEnt(0);
while(prod != 0)
{
    printf("\nIngrese la cantidad en existencia del codigo %d - mayor o igual a cero: \t", prod);
    cant=LeeYValidaEnt(0);
    printf("\nIngrese el precio unitario del codigo %d - mayor o igual a cero: \t", prod);
    precio=LeeYValidaReal(0.0);
    P[i]=prod;
    C[i]=cant;
    I[i]=precio;
    i++;
    printf("\nIngrese el codigo del producto - mayor a cero - Cero para terminar: \t");
    prod=LeeYValidaEnt(0);
}
return i;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int IngrDatoValEnt(int linf, int lsup)
{
    int dato;
    bool OK=true;

    do{
        if(OK)
        {
            printf("\nIngrese un entero entre %d y %d: \t", linf, lsup);
            scanf("%d", &dato);
            OK=false;
        }
        else
        {
            printf("\nERROR - Ingrese un caracter entre %d y %d: \t", linf, lsup);
            scanf("%d", &dato);
        }
    }
}
```

```
    }
    }while(dato<linf || dato>lsup);

    return dato;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
char IngrDatoValCH(char li, char ls)
{
    char dato;
    bool OK=true;

    do{
        if(OK)
        {
            printf("\nIngrese un caracter entre %c y %c: \t", li, ls);
            fflush(stdin);
            scanf("%c", &dato);
            dato=toupper(dato);
            OK=false;
        }
        else
        {
            printf("\nERROR - Ingrese un caracter entre %c y %c: \t", li, ls);
            fflush(stdin);
            scanf("%c", &dato);
            dato=toupper(dato);
            OK=false;
        }
    }while(dato<li || dato>ls);

    return dato;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int Busqueda (int V[], int ce, int dato)
{
    int i=0;

    while(V[i] != dato && i<ce)
```

```
        i++;
    if(i==ce)
        return -1;
    else
        return i;
}

/////////////////////////////////////////////////////////////////
int BusquedaCH(char V[], int ce, char dato)
{
    int i=0;

    while(V[i] != dato && i<ce)
        i++;
    if(i==ce)
        return -1;
    else
        return i;
}

/////////////////////////////////////////////////////////////////
int PosicionMayor(int V[], int ce)
{
    int p, i, max;

    for(i=0; i<ce; i++)
        if(i==0 || V[i]>max)
        {
            max=V[i];
            p=i;
        }
    return p;
}

/////////////////////////////////////////////////////////////////
int Maximo(int V[], int ce)
{
    int m, i;
```

```
    for(i=0; i<ce; i++)
        if(i==0 || V[i]>m)
            m=V[i];

    return m;
}

/////////////////////////////////////////////////////////////////
void MostrarParIgualesA(int C[], char P[], int ce, int dato)
{
    int i;

    for(i=0; i<ce; i++)
        if(C[i]==dato)
            printf("\n%c", P[i]);
}

/////////////////////////////////////////////////////////////////
void MostrarVecParal(int P[], int C[], float U[], int ce)
{
    int i;

    for(i=0; i<ce; i++)
        printf("\n%5d\t%.2f", P[i], C[i]*U[i]);
}
```

Práctica para el Segundo Parcial

P2.1 La empresa *El Clavel* que comercializa flores exóticas por mayor, desea analizar las ventas efectuadas por sus 110 sucursales mensualmente a lo largo del país, en las principales ciudades.

Para ello, se tiene una lista para ingresar desde teclado con los datos de las 12 variedades de flores que comercializa. Al comienzo del programa, por cada tipo de flor, se ingresa:

- Código de la variedad de la flor (entero, de 101 a 999)
- Precio de la flor por unidad (real, mayor que cero)

A continuación, ingresar el mes y año correspondiente a los datos a procesar.

Luego se ingresan los datos de las ventas del mes por teclado, con el siguiente detalle:

- Número de sucursal (entero, de 5001 a 5110)
- Código de la flor
- Cantidad vendida (entero, mayor que cero)

Estos datos no se ingresan ordenados y la lectura debe detenerse cuando se ingrese un número de sucursal igual a 5000.

Se pide:

- a. Informar los importes vendidos por cada sucursal con el siguiente diseño:

<u>VENTAS DEL MES XX DE XXXX</u>	
Sucursal	Importe vendido
xxx	xxxx.xx
xxx	xxxx.xx
.....
.....

- b. Código de la variedad de flor de la cual se vendieron más cantidad de flores. Si hubiere más de una, informar la primera que aparezca en esa condición.
- c. Informar cuántas sucursales vendieron flores por un importe superior al promedio de ventas de todas las sucursales.
- d. Informar el porcentaje de las flores de las cuales no se vendió nada en ninguna sucursal.

P2.2 La Empresa Corrientes Transportadora S.A. se dedica a la prestación de servicios de mensajería, manipulación de correspondencia, entrega de documentos y paquetería. Desea controlar los viajes realizados por sus 25 vehículos, para lo cual se ingresan los datos de dichos vehículos:

- Código de vehículo (entero, mayor que cero).
- Capacidad máxima en kilos (real, entre 500 y 32000).

En un segundo lote de datos se ingresan por cada viaje realizado en el mes, el siguiente detalle:

- Código de vehículo
- Peso total de la mercadería transportada en un viaje (real, mayor a 0).

Se debe rechazar el viaje cuando se supere el tope máximo en kilos del vehículo informado. Dentro del mes un vehículo puede hacer varios viajes, uno o ninguno. La finalización de la carga de este segundo lote se da por un código de vehículo igual a -1.

Se pide:

- a) Informar, al finalizar el mes, el porcentaje de kilos transportado por cada uno de los vehículos respecto del total.
- b) Informar la cantidad de kilos transportado por cada vehículo con el siguiente diseño:

<u>NÚMERO DE PATENTE</u>	<u>KILOS TRANSPORTADOS</u>
--------------------------	----------------------------

- c) Informar el Código de vehículo, que ha tenido la mayor cantidad de viajes rechazados por exceso de kilos (si hay varios, informar el último que aparezca).

P2.3 La Subsecretaría de Políticas Universitarias (SPU) recibe de forma anual el detalle de la actividad académica de las casi 2800 Instituciones de Educación Superior Nacionales (universitarios y no universitarios) que existen en la actualidad. Para ello cuenta con un primer lote de información con los siguientes datos de cada Institución:

- Código de la Institución (entero, entre 1000 y 9999).
- Tipo de Institución (carácter, 'U': Universitaria, 'I': Instituto no Universitario)

Este lote de información termina con un código de Institución igual a cero.

En un segundo lote de datos se ingresan los datos enviados por las Instituciones durante el período en estudio, con el siguiente detalle:

- Código de la Institución.
- Código de la materia-comisión (entero, mayor a cero)
- Cantidad de alumnos que aprobaron la cursada (entero, mayor o igual a 0).

La finalización de la carga de este segundo lote se da por un Código de Institución igual a cero.

Se pide lo siguiente:

- a) Informar, al finalizar el período estudiado, el promedio de notas de Universidades y de Institutos no Universitarios.
- b) Informar por Código de Institución, la cantidad de alumnos que aprobaron la cursada en cualquiera de los tipos de Institución con el siguiente diseño:

<u>CÓD. DE INSTITUCIÓN</u>	<u>CANT. DE ALUMNOS QUE APROBARON</u>
----------------------------	---------------------------------------

- c) Informar el porcentaje de instituciones que tuvieron menos de 100 aprobados respecto del total de instituciones.

Ejemplo de Parcial 2 resuelto

Para poder controlar las elecciones nacionales del País, la Justicia Electoral decide realizar un sistema para poder evaluar algunos resultados rápidamente, con el objetivo de realizar algunas consideraciones estadísticas. Para ello al comienzo del programa se ingresan códigos de 3 cifras que representan a las 24 provincias que pueden presentarse a las elecciones como distritos electorales, según el siguiente detalle:

- Código de la provincia (entero, de 101 a 999)
- Cantidad de ciudadanos habilitados para votar en el Padrón Provincial (entero, mayor a cero)

Al finalizar los comicios, comienzan a llegar los datos totales de las personas que se han presentado a votar en las distintas provincias, según el siguiente detalle:

- Código de la provincia
- Número de Mesa (entero, mayor a cero)
- Cantidad de votos (entero, mayor a cero)

Estos datos no se ingresan ordenados, no se repiten los datos de las mesas, pero sí se puede repetir la provincia con distintas mesas y terminan cuando se ingrese un código de provincia igual a cero. Asumimos que todas las provincias se presentan a desarrollar las elecciones el mismo día.

Determinar:

- a. Informar cuántos ciudadanos que estaban habilitados para votar NO votaron, es decir, estuvieron ausentes en la jornada cívica, con el siguiente diseño:

<u>CIUDADANOS QUE NO VOTARON</u>	
<u>PROVINCIA</u>	<u>CANTIDAD DE CIUDADANOS QUE NO VOTARON</u>
XXX	XXXXXXXX
.....
XXX	XXXXXXXX
TOTAL	XXXXXXXXXX

- b. Indicar el porcentaje de provincias que tienen más de 100000 personas del padrón de la provincia que votaron.
- c. Informar en qué mesa hubo la máxima cantidad de votos, indicando la provincia a la que pertenece.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <stdbool.h>
```

```
#define NUM_PROVINCIAS 24
```

```
// Declaración de funciones
```

```
int IngrDatoValCF(int, int, int);
int LeeYValidaEnt(int);
void registrarPadrones(int[], int[], int);
int Busqueda(int[], int, int);
void MostrarVecPar(int[], int[], int[], int);
int ContarMayorA(int[], int, int);

int main()
{
    int codigosProvincias[NUM_PROVINCIAS], codigo, mesa, votos, pos, max, mesa_max, prov_max;
    int padronProvincias[NUM_PROVINCIAS]={0}, VVotos[NUM_PROVINCIAS]={0}, cont100K;
    bool primero=true;

    registrarPadrones(codigosProvincias, padronProvincias, NUM_PROVINCIAS);
    printf("\nIngreso de votos por provincia\n");
    printf("\nIngrese el codigo de provincia - entero 101 y 999 - Cero para terminar: \t");
    codigo=IngrDatoValCF(101, 999, 0);
    while(codigo != 0)
    {
        printf("\nIngrese el número de mesa - mayor a cero: \t");
        mesa=LeeYValidaEnt(0);
        printf("\nIngrese la cantidad de votos de la mesa %d - mayor a cero: \t", mesa);
        votos=LeeYValidaEnt(0);
        pos=Busqueda(codigosProvincias, NUM_PROVINCIAS, codigo);
        if(pos != -1)
        {
            VVotos[pos]+=votos;
            if(primero || votos > max)
            {
                max=votos;
                mesa_max=mesa;
                prov_max=codigo;
                primero=false;
            }
        }
    }
}
```

```
}  
  
else  
  
    printf("\nEl código de provincia ingresado no existe\n");  
  
  
    printf("\nIngrese el código de provincia - entero 101 y 999 - Cero para terminar: \t");  
    codigo=IngrDatoValCF(101, 999, 0);  
}  
printf("\nCIUDADANOS QUE NO VOTARON \n\n");  
printf("PROVINCIA\tCANTIDAD DE CIUDADANOS QUE NO VOTARON\n");  
MostrarVecPar(codigosProvincias, VVotos, padronProvincias, NUM_PROVINCIAS); //Punto a  
cont100K=ContarMayorA(VVotos, NUM_PROVINCIAS, 100000);  
printf("\nEl porcentaje de provincias que votaron mas de 100000 personas es: %.2f",  
(float)cont100K/NUM_PROVINCIAS*100); //Punto b  
printf("\nLa mayor cantidad de votos por mesa es %d en la mesa %d de la provincia %d\n", max, mesa_max,  
prov_max); //Punto c  
  
  
printf("\n\n\n");  
system("pause");  
return 0;  
}  
  
////////////////////////////////////  
void registrarPadrones(int C[], int P[], int size)  
{  
    int i, codigo, padron;  
  
    printf("\nIngrese los datos de las provincias:\n");  
    for (i = 0; i < size; i++)  
    {  
        printf("\nIngrese el Código de la provincia - entre 101 y 999): \t");  
        codigo=IngrDatoValCF(101, 999, 999);  
  
        printf("\nIngrese la cantidad de ciudadanos habilitados para votar: \t");  
        padron= LeeYValidaEnt(0);  
        C[i] = codigo;  
        P[i] = padron;  
    }  
}
```

```
}  
////////////////////////////////////  
int IngrDatoValCF(int li, int ls, int cf)  
{  
    int dato;  
    bool OK=true;  
  
    do{  
        if(OK)  
        {  
            scanf("%d", &dato);  
            OK=false;  
        }  
        else  
        {  
            printf("\nERROR - Ingrese un valor entre %d y %d o %d: \t", li, ls, cf);  
            scanf("%d", &dato);  
        }  
    }while((dato<li || dato>ls) && dato != cf);  
  
    return dato;  
}  
////////////////////////////////////  
int LeeYValidaEnt(int lim)  
{  
    int dato;  
  
    scanf("%d", &dato);  
    while (dato<lim)  
    {  
        printf("\nERROR - El dato ingresado tiene que ser mayor o igual a %d: \t", lim);  
        scanf("%d", &dato);  
    }  
    return dato;  
}  
////////////////////////////////////
```

```
int Busqueda (int V[], int ce, int dato)
{
    int i=0;

    while(V[i] != dato && i<ce)
        i++;
    if(i==ce)
        return -1;
    else
        return i;
}

/////////////////////////////////////////////////////////////////

void MostrarVecPar(int Cod[], int Cant[], int Pad[], int ce)
{
    int i;

    for(i=0; i<ce; i++)
        printf("\n%5d\t%10d", Cod[i], Pad[i] - Cant[i]);
}

/////////////////////////////////////////////////////////////////

int ContarMayorA(int V[], int ce, int dato)
{
    int i, c=0;

    for(i=0; i<ce; i++)
        if(V[i]>dato)
            c++;
    return c;
}
```