

ALGORITMOS Y ESTRUCTURA DE DATOS

1

Módulo I, guía de ejercicios

Ejercicio 1

Dados dos valores numéricos enteros calcular e informar la suma, la diferencia y el producto entre dichos valores.

Ejercicio 2

Dada una terna de números naturales que representan el día, el mes y el año de una determinada fecha informarla como un solo número natural de 8 dígitos (*aaaammdd*).

Ejercicio 3

Dado valor numérico entero que se ingresa por teclado, se pide informar:

- La quinta parte de dicho valor,
- el resto de la división por 5 y
- la séptima parte de la quinta parte.

Ejercicio 4

Dados dos valores numéricos diferentes entre si, informar cual es el mayor.

Ejercicio 5

Dados dos valores numéricos, informar cual es el mayor y cual es el menor o, si ambos valores son iguales emitir un mensaje.

Ejercicio 6

Dadas dos fechas, informar cual es la más reciente. Determinar cuales deben ser los datos de entrada y en que formato los debe ingresar el usuario.

Ejercicio 7

Dado un triángulo representado por sus lados *lado1*, *lado2* y *lado3*, determinar e indicar según corresponda: “equilátero”, “isósceles” o “escálenos”.

Ejercicio 8

Dados un mes y el año al que corresponde, informar cuantos días tiene el mes.

Ejercicio 9

Se ingresa una edad, mostrar por pantalla alguna de las siguientes leyendas:

- “menor” si la edad es menor o igual a 12,
- “cadete” si la edad está comprendida entre 13 y 18,
- “juvenil” si la edad es mayor que 18 y no supera los 26, y
- “mayor” si no cumple ninguna de las condiciones anteriores.

Ejercicio 10

Dados dos valores numéricos enteros, calcular e informar su producto mediante sumas sucesivas.

Ejercicio 11

Dados 50 números enteros, informar el promedio de los mayores que 100 y la suma de los menores que -10.

Ejercicio 12

En un torneo de fútbol participan k equipos. El torneo se juega con el sistema de “todos contra todos”. Por cada partido en el que participa un equipo se dispone de la siguiente información:

- código de equipo (codEq) y
- código de resultadp (“P” = perdido, “E” = empatado, “G”= ganado).

Se dispone de un lote de datos conteniendo todos los resultados del torneo, agrupados por *codEq*.

Desarrollar el programa que imprima, por cada equipo, su código y el puntaje total que obtuvo considerando que suma 3 puntos por cada partido ganado y 1 punto por cada partido empatado.

Ejercicio 13

Se ingresa un conjunto de valores reales, cada uno de los cuales representan el sueldo de un empleado, excepto el último valor que será 0 para indicar el fin del ingreso de datos. Se pide desarrollar un programa que determine e informe:

- Cuántos empleados ganan menos \$2000.
- Cuántos ganan \$2000 o más pero menos de \$3000.
- Cuántos ganan \$3000 o más pero menos de \$5000.
- Cuántos ganan \$5000 o más.

Ejercicio 14

Dado un valor numérico entero m , determinar e imprimir un listado con los m primeros múltiplos de 3 que no sean múltiplos de 5.

Ejercicio 15

Dados 10 valores numéricos, informar cual es el mayor.

Ejercicio 16

Dados n valores numéricos, informar el mayor, el menor y en que posición del conjunto fueron ingresados cada uno de ellos.

Ejercicio 17

Dado un conjunto de nombres y fechas de nacimiento con formato *ddmmaaaa*, que finaliza cuando se ingrese un nombre igual a "fin", informar el nombre de la persona con mayor edad y el de la más joven.

Ejercicio 18

Dado un conjunto de valores numéricos que finaliza con el ingreso de un valor nulo, determinar e imprimir:

- El valor máximo negativo.
- El valor mínimo positivo.
- El valor mínimo dentro del rango -17.3 y 26.9.

- El promedio de todos los valores ingresados.

Ejercicio 19

Se dispone de un conjunto de valores enteros positivos que finaliza con el ingreso de un número negativo.

El conjunto está dividido en subconjuntos mediante valores 0 (cero). Se pide Desarrollar un programa que determine e informe:

- Por cada subconjunto, el promedio de valores que lo componen.
- El total de subconjuntos procesados.
- El valor máximo del conjunto, indicando en qué subconjunto se encontró y su posición relativa dentro del mismo.
- El valor mínimo de cada subconjunto.

Nota: el lote puede estar vacío (el primer valor ingresado será negativo), o bien puede contener uno, varios o todos los sublotos vacíos (ceros consecutivos).

Ejercicio 20

Dada una serie de m pares (color, número) que corresponden a los tiros de una ruleta. Se pide informar:

- Cuántas veces salió el número 0 y, por cada 0, cual fue el número anterior.
- Cuántas veces seguidas llegó a repetirse el color negro.
- Cuántas veces seguidas llegó a repetirse el mismo número y cuál fue.
- El mayor número de veces seguidas que salieron alternados los colores rojo y negro.
- El mayor número de veces seguidas que se negó la segunda docena.

Nota: recuerde que la ruleta tiene números comprendidos entre 0 y 36. “Negar la segunda docena” significa que el número en cuestión no está comprendido entre 12 y 23.

Ejercicio 21

Un buque de carga traslada 100 contenedores a 3 diferentes puertos del país identificados con los números 1, 2 y 3.

Por cada uno de los contenedores trasladados por el buque se registran los siguientes datos:

- Identificación del contenedor: idCont.
- Peso del contenedor en (en kilos): peso.
- Puerto de arribo (un valor de 1 a 3): idPuerto.

Se pide calcular e informar:

- El peso total que el buque debe trasladar.
- El contenedor de mayor peso.
- La cantidad de contenedores que se trasladarán a cada puerto.

Ejercicio 22

Se realiza una inspección en una fábrica de pinturas, y se detectaron algunas infracciones. De cada infracción se tomó nota de los siguientes datos:

- Tipo de Infracción (1, 2, 3, ó 4): tInfr.
- Motivo de la infracción: motivo.
- Valor de la multa: valor.
- Gravedad de la infracción ('L', 'M', 'G'): gravedad.

Los datos finalizan cuando se ingrese un `tInfr` igual a cero. Se pide informar al final del proceso:

- Los valores totales de la multa a pagar de acuerdo a la gravedad de las infracciones registradas.
- La leyenda "Clausurar Fábrica" si la cantidad de infracciones 3 y 4 con gravedad "G" es mayor a 3.
- Informar el motivo correspondiente la infracción de menor valor.

Ejercicio 23

Dada una serie de caracteres que conforman una oración, donde cada palabra está separada de la siguiente por un carácter blanco y la oración finaliza con un punto. Se pide informar:

- Cantidad de veces que apareció cada vocal.
- Cantidad de palabras que contiene la oración.
- Cantidad de letras que posee la palabra más larga.

Ejercicio 24

Dado un número entero positivo entre 1 y 3999 informar su correspondiente número Romano.

Ejercicio 25

Desarrollar un programa tal que, dado un valor entero positivo que se ingresa por teclado indique si se trata de un número primo o no.

Nota: los números primos son aquellos valores enteros positivos que solo son divisibles por sí mismos y por la unidad.

Ejercicio 26

Dado un valor entero positivo n que se ingresa por teclado, desarrollar un programa que muestre por consola los primeros n números primos.

Ejercicio 27

Desarrollar un programa tal que, dado un valor entero positivo que se ingresa por teclado y que indica la posición de un término dentro de la sucesión de Fibonacci, muestre por consola el valor del término ubicado en dicha posición.

Nota: la sucesión de Fibonacci es una serie numérica cuyos primeros dos términos son 1 y los siguientes términos se calculan como la suma de los dos anteriores.

Los primeros términos de la serie son los siguientes:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ..., t_i , t_{i+1} , t_i+t_{i+1} , ...

Ejercicio 28

Desarrollar un programa que muestre por pantalla los primeros n términos de la serie numérica de Fibonacci, donde n es un valor que se ingresa por consola.

Ejercicio 29

Sea el conjunto A de valores enteros, descripto más abajo; se pide calcular e informar lo siguiente:

- cuántos valores 0 hubo,
- promedio de valores positivos y
- la sumatoria de valores negativos.

Resolver el ejercicio considerando que:

- El conjunto A se compone de 167 valores enteros.
- El conjunto A se compone de n valores numéricos, donde el valor de n debe ser leído previamente.
- El conjunto A se compone de varios valores (no se sabe cuantos), cuyo ingreso finaliza con la llegada de un valor igual al anterior.
- El conjunto A se compone de n valores, pero el proceso deberá finalizar si se procesan todos los valores o si la cantidad de ceros supera a cuatro.
- El conjunto A se compone de n , pero el proceso deberá finalizar si el promedio de positivos resulta mayor que seis.

Ejercicio 30

El gobierno de la Ciudad de Buenos Aires realiza una encuesta en casas de familia. De cada familia se conoce: domicilio, tipo de vivienda ('C': casa, 'D': departamento), y cantidad de integrantes.

De cada integrante de la familia se conoce: nombre, apellido, edad, sexo ('F', 'M'), nivel de estudios alcanzados ('N': no posee, 'P': primario, 'S': secundario, 'T': terciario, 'U': universitario), y un indicador ('I': incompleto, 'C': completo) que se refiere al ítem anterior.

Los datos finalizan cuando la cantidad de integrantes sea igual a cero. Se pide emitir un listado con los resultados:

- Los datos de los encuestados que hayan completado los estudios primarios.
- El porcentaje de analfabetismo en la ciudad (se considera analfabetos a los mayores de 10 años que no posean estudios).
- El domicilio de la familia con mayor cantidad de integrantes que viven en departamento.
- La edad promedio de cada familia y de la ciudad.
- La cantidad de encuestados en cada tipo de nivel de estudios alcanzados incompletos.
- El porcentaje de encuestados de sexo femenino y masculino.

Ejercicio 31

Una compañía aérea desea emitir un listado con los movimientos mensuales de sus m vuelos al exterior. Para ello cuenta con la siguiente información:

- De cada vuelo realizado el número de vuelo, destino, y cantidad de asientos.
- De cada pasajero el número de pasaporte y el importe que abonó por el pasaje (en dólares).

La información finaliza con un número de pasaporte igual a cero. Se pide emitir el siguiente listado:

Nro. de Vuelo: 9999, Destino: xxxxxxxxxxxxxxxxx

<u>Nro. de Pasaporte</u>	<u>Importe en u\$s</u>
--------------------------	------------------------

99999999	999.99
----------	--------

99999999	999.99
----------	--------

99999999	999.99
----------	--------

Total recaudado del vuelo: \$99999.99

% de Asientos Libres del vuelo: 999.99

% de Asientos Ocupados del vuelo: 999.99

: : :

Total recaudado en el mes: \$999999.99

Cantidad de veces seguidas que se dieron vuelos completos: 99

El número de vuelo que más recaudó: 9999

Ejercicio 32

Desarrollar una función que calcule el máximo común divisor de dos números enteros a y b con el siguiente algoritmo:

1. Dividir a por b , y calcular el resto ($0 < r < b$)
2. Si $r==0$ entonces el MCD es b , si no seguir en (3)
3. Reemplazar a por b , b por r , y volver al paso (1)

La función debe tener el siguiente prototipo:

```
int mcd(int a,int b);
```

Ejercicio 33

Desarrollar una función que reciba un valor entero y retorne su factorial. El prototipo de la función debe ser el siguiente:

```
int factorial(int n);
```


Analice las siguientes situaciones:

1. Compruebe el comportamiento de la función factorial cuando recibe valores pequeños como parámetro.
2. Compruebe el comportamiento de la función cuando recibe valores grandes (por ejemplo mayores que 100).
3. Si en (2) experimentó algún tipo de error, ¿cual considera que puede ser la causa del problema y cómo lo solucionaría?

Ejercicio 34

Desarrolle la función `unificarFecha` que recibe tres parámetros que representan el día, mes y año de una fecha y retorna la fecha unificada en un entero largo de 8 dígitos con el siguiente formato: `aaaammdd`. El prototipo de la función debe ser el siguiente:

```
long unificarFecha(int dia, int mes, int anio);
```

Ejercicio 35

Desarrollar la función `esMultiplo` que recibe dos valores a y b y retorna `true` o `false` según a sea o no múltiplo de b . El prototipo será:

```
bool esMultiplo(int a, int b);
```

Ejercicio 36

Dada una serie de números naturales, indicar para cada uno su factorial e informar cuantos son múltiplos de 3, de 5, de 7 y cuantos son múltiplos de 3 y de 5. Invocar las funciones desarrolladas anteriormente.

Ejercicio 37

Dada la fracción P/Q , siendo P y Q números naturales, informar la mayor cantidad de simplificaciones posibles. Desarrolle y utilice una función que reciba dos números naturales y retorne el menor factor común. Ej: $360/60 = 180/30 = 90/15 = 30/5 = 6/1$.

Ejercicio 38

Desarrolle la función `calcularPorcentajeDiferencia` que recibe dos valores a , b y retorna el siguiente cálculo: $(b-a) * 100 / (a+b)$. El prototipo será:

```
float calcularPorcentajeDiferencia(long a, long b);
```

Ejercicio 39

Desarrollar la función `continuar` de forma tal que imprima un mensaje en la pantalla con el texto "Presione una tecla para continuar...".

El prototipo debe ser:

```
void continuar();
```

Ejercicio 40

Desarrollar la función `pow` que recibe una base y un exponente y retorne la potencia. El prototipo será:

```
double pow(double x, double y);
```

Ejercicio 41

Dada la siguiente codificación, se pide desarrollar la representación gráfica del algoritmo, el seguimiento, escribir un enunciado, especificando cuales son las variables locales y los parámetros por valor y por dirección.

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

using namespace std;

void raiz(double a, double b, double c, bool& ok)
{
    double x1, x2;
    double discr;

    discr = b*b-4*a*c;

    if( discr>=0 )
    {
        discr = sqrt(discr);
        x1 = (-b+discr)/(2*a);
        x2 = (-b-discr)/(2*a);
        ok = true;
    }
    else
    {
        ok = false;
    }
}
```

Ejercicio 42

Desarrollar una función `void` tal que dada una hora (*hhmmss*) y un tiempo también en formato *hhmmss* devuelva la nueva hora que surge de sumar el tiempo a la hora inicial, considere también un posible cambio el día. El prototipo de la función debe ser el siguiente:

```
void sumarHora(long h, long t, long& res);
```

Ejercicio 43

Desarrollar y probar la función `esPrimo` que, dado un valor numérico que recibe como parámetro retorna true o false indicando si se trata o no de un número primo.

El prototipo de la función debe ser:

```
bool esPrimo(int n);
```

Ejercicio 44

Dado un valor entero que representa una suma de dinero, desarrollar un programa que indique qué cantidad de billetes de \$100, \$50, \$20, \$10, \$5, \$2 y \$1 se deben utilizar para alcanzarla, dando prioridad a los billetes de mayor denominación.

Por ejemplo: Si la suma de dinero fuese: \$2579 entonces se necesitarán 25 billetes de \$100, 1 billete de \$50, 1 billete de \$20, 1 billete de \$5 y 2 billetes de \$2.

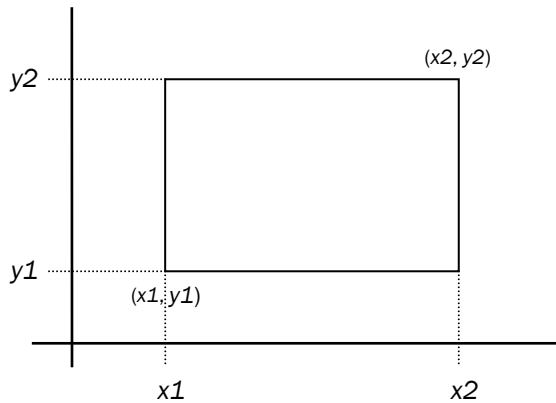
Ejercicio 45

Dada la función `calcularBilletes` cuyo código se muestra a continuación, rehacer el ejercicio anterior invocandola donde corresponda.

```
int calcularBilletes(int& monto, int denom)
{
    int cant = monto/denom;
    monto = monto%denom;
    return cant;
}
```

Ejercicio 46

Dado un conjunto de 20 elementos (*x1, y1, x2, y2*), donde cada uno representa los vértices opuestos de un rectángulo tal como se ilustra en la figura:



Se pide:

1. Informe el perímetro de cada rectángulo que verifique ser: cuadrado.
2. Al final del algoritmo informe la cantidad de rectángulos ingresados que no son cuadrados.

Utilice la función `abs` incluida en la biblioteca `cmath`, que retorna el valor absoluto del argumento que recibe como parámetro.

Ejercicio 47

De un censo realizado en una población se conocen los siguientes datos:

1. Día de nacimiento (2 dígitos)
2. Mes (2 dígitos)
3. Año (4 dígitos)
4. Sexo ('M'=masculino, 'F'=femenino)

Con estos datos de cada habitante se forma un lote, finalizado su ingreso con un día igual a 0.

Se pide desarrollar el programa que determine e imprima:

1. Cuántos nacimientos hubo en el mes de octubre de todos los años.
2. Cuántos nacimientos hubo antes del 9 de julio de 1990.
3. Cuántos nacimientos de mujeres hubo en la primavera del 1982.
4. Sexo de la persona más vieja (solo existe una).

Ejercicio 48

Desarrollar una función que reciba un número natural de 4 dígitos en formato de *hhmm* que corresponde a un tiempo determinado en horas y minutos, y retorne ese tiempo en minutos. El prototipo debe ser el siguiente:

```
long toMin(long t);
```

Desarrollar una función `void` que reciba el costo (expresado en pesos) de un abono telefónico, la cantidad de minutos libres que incluye el abono, el cargo (expresado en pesos) por minuto excedente y la cantidad de minutos utilizados por un abonado y retorne la cantidad de minutos excedidos y el monto (también expresado en pesos) que se debe abonar (costo del abono más minutos excedidos por el costo de minutos excedidos) más el 21% del valor del IVA.

El prototipo y posterior desarrollo de esta función queda a cargo del alumno.

Utilizando lo anterior, se pide desarrollar un algoritmo que resuelva la siguiente situación problemática:

Todos los fines de mes, una empresa de telefonía celular debe confeccionar las facturas con los consumos de todos sus abonados, que se realizan en tres turnos de trabajo: Mañana, Tarde y Noche.

Para ello se ingresará por teclado la siguiente información de cada celular:

- Número de celular 9 dígitos (0 indica cambio de turno).
- Nombre del abonado 20 caracteres.
- Dirección del abonado 25 caracteres.
- Tiempo utilizado 4 dígitos en formato *hhmm*.
- Tipo de abono (carácter, A, B, C, D o E).

Dependiendo del tipo de abono que se tenga, el usuario tiene cierta cantidad de minutos libres, por los cuales no abona cargo extra, pero por cada minuto que se exceda debe abonar una suma extra según la siguiente tabla:

Plan	A	B	C	D	E
Costo	\$70	\$55	\$40	\$28	\$19
Minutos Libres	300	200	100	60	40
Cargo Min Exced.	\$0.09	\$0.15	\$0.21	\$0.29	\$0.37

Emitir el siguiente listado por cada turno:

Turno: Mañana

<u>Nombre Abonado</u>	<u>Dirección</u>	<u>Minutos libres</u>	<u>Minutos Excedidos</u>	<u>Monto Total a Abonar</u>
xxxxxxxxxxx	xxxxxxxxx	99999	99999	\$ 9999.99
xxxxxxxxxxx	xxxxxxxxx	99999	99999	\$ 9999.99
xxxxxxxxxxx	xxxxxxxxx	99999	99999	\$ 9999.99

.....
Turno: Tarde

.....
Turno: Noche