



Guía TP primer parcial

martes, 16 de marzo de 2021 18:58



Fdl - Guía
de



FACULTAD DE INGENIERIA Y CIENCIAS EXACTAS
Departamento de Tecnología Informática

FUNDAMENTOS DE INFORMÁTICA

Guía de Trabajos Prácticos
v18.10.17

Objetivos: Que el alumno pueda:

- Entender y saber construir un programa
- Expresar un programa en términos estructurados
- Diferenciar conceptualmente los distintos tipos de estructuras de control
- Plantear estrategias de resolución de problemas
- Identificar módulos que puedan resolverse mediante procedimientos
- Utilizar estructuras de datos básicas

Bibliografía sugerida:

Básica

- Joyanes Aguilar, Luis: Fundamentos de Programación. Editorial Mc.Graw-Hill. ISBN 978-84-481-6111-8. EAN 9788448161118.

Complementaria

- **Tutorial de Python 3** [En línea] Python Software Foundation. Disponible gratuitamente bajo la licencia PSF. Buscar links en Google.
- Pilgrin, Mark: **Inmersión en Python 3** [En línea] Traducido por José Miguel González Aguilera. Disponible gratuitamente (bajo licencia Creative Commons 3.0). . Buscar links en Google.
- Marzal Varó, Andres - Gracia Luengo, Isabel, García Sevilla, Pedro: **Introducción a la programación con Python 3** [En línea] Disponible gratuitamente bajo licencia Creative Commons. Buscar links en Google.

Autor:
Ricardo Thompson

Trabajo Práctico 1: Algoritmos elementales y cálculo de expresiones

Ejercicio 1: Describir la secuencia de acciones necesarias para:

- Enviar un mensaje por Whatsapp.
- Tomar un colectivo.
- Llegar a la vereda desde la posición en la que se encuentra.
- Lavarse las manos.
- Preparar un mate.
- Recargar la tarjeta SUBE en una terminal automática de recarga.
- Destapar una botella de cerveza.
- Poner una alarma en el reloj del celular.
- Cruzar una calle en una esquina con semáforos.
- Pintar una pared.
- Comprar una entrada de cine por Internet.
- Separar el cuatro de copas de un conjunto de naipes.
- Buscar la menor carta de un mazo de naipes, que puede no estar completo.

Ejercicio 2: Calcular el valor de las siguientes expresiones, respetando el orden de operaciones establecido.

- $12 * 4 + 4 * 5$
- $(12 * (1 - 5) + 4) * 3$
- $12 * 1 - 5 + 4 * 3$
- $(17 - 2) / 5$
- $3 + 2 * 2 - (8 * 4 + 1 / 2.0) * 3$
- $5 * 4 / 2$
- $5 * (4 / 2)$
- $24 / 2 ** 2$
- $(24 / 2) ** 2$
- $3 + 4 * 6 / 2 - 5$
- $3 + 4 * 6 / (2 - 5)$
- $(- 0.1) * 3$
- $- 9 ** 2$
- $(- 9) ** 2$
- $10 / 3$
- $10 // 3$
- $12 \% 5$

Trabajo Práctico 2: Estructura secuencial

Ejercicio 1: Indicar el valor de la variable X después de la ejecución de las siguientes secuencias de comandos:

- | | | |
|--------------------------------------|--|---|
| a) $m = 2$
$x = 4$
$y = x + m$ | b) $a = 3$
$b = 4$
$x = a * a - b$ | c) $x = 3$
$h = x * (-4) + 2$
$x = y$ |
|--------------------------------------|--|---|

Ejercicio 2: Indicar si es posible alterar el orden de las siguientes secuencias de sentencias sin provocar cambios en los valores finales de las variables:

- | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|
| a) $X = Y$
$Y = Z$ | b) $X = Y$
$Z = X$ | c) $X = Z$
$Y = Z$ | d) $X = Y$
$X = Z$ |
|-----------------------|-----------------------|-----------------------|-----------------------|

Ejercicio 3: Desarrollar un programa que permita ingresar dos números enteros A y B a través del teclado. Imprimir su suma y su diferencia.

Ejercicio 4: Ingresar la longitud del radio de un círculo. Calcular e imprimir:

- La superficie del círculo ($\text{Sup} = \pi * r^2$)
- El perímetro de la circunferencia ($\text{Per} = \pi * d$)
- La superficie de la esfera ($\text{Sup} = 4 * \pi * r^2$)
- El volumen de la esfera ($\text{Vol} = 4/3 * \pi * r^3$)

Ejercicio 5: Escribir un programa que permita ingresar la edad de una persona en años y la convierta a días, imprimiendo el resultado. Considerar que todos los años tienen 365 días.

Ejercicio 6: Leer una medida en metros e imprimir esta medida expresada en centímetros, pulgadas, pies y yardas. Los factores de conversión son:

- 1 pie = 12 pulgadas
- 1 yarda = 3 pies
- 1 pulgada = 2,54 cm.
- 1 metro = 100 cm.

Ejercicio 7: Ingresar tres números enteros, calcular su promedio y mostrarlo por pantalla.

Ejercicio 8: Leer un período en segundos e imprimirlo expresado en días, horas, minutos y segundos. Por ejemplo, 200000 segundos equivalen a 2 días, 7 horas, 33 minutos y 20 segundos.

Ejercicio 9: Una inmobiliaria paga a sus vendedores un salario de \$800, más una comisión de \$50 por cada venta realizada, más el 5% del valor de esas ventas. Realizar un programa que imprima el número del vendedor y el salario que le corresponde en un determinado mes. Se leen el número del vendedor, la cantidad de ventas que realizó y el valor total de las mismas.

Ejercicio 10: Un banco necesita para sus cajeros automáticos un programa que lea una cantidad de dinero e imprima a cuántos billetes equivale, considerando que existen billetes de \$100, \$50, \$10, \$5 y \$1. Desarrollar dicho programa de tal forma que se minimice la cantidad de billetes entregados por el cajero.

Ejercicio 11: Desarrollar un programa que solicite una temperatura expresada en grados centígrados y la imprima convertida a grados Fahrenheit, tal que:

$$^{\circ}\text{C} = \frac{5}{9} * (^{\circ}\text{F} - 32)$$

Ejercicio 12: Escribir un programa para convertir un número binario de 4 cifras en un número decimal. El número binario se ingresa como un solo número entero de cuatro dígitos.

Trabajo Práctico 3: Estructura alternativa

Ejercicio 1: Ingresar dos números A y B e imprimir el mayor, o cualquiera si son iguales.

- Ejercicio 2:** Leer un número entero A e imprimir un mensaje indicando si es par o impar.
- Ejercicio 3:** Leer un número entero N y determinar si es un número natural (positivo y distinto de 0). Si lo es, imprimirlo junto con su doble. En caso contrario, imprimirlo junto con su triple.
- Ejercicio 4:** Ingresar dos números enteros A y B. Desarrollar un programa que determine si A es múltiplo de B y si B es múltiplo de A. Imprimir mensajes aclaratorios.
- Ejercicio 5:** Desarrollar un programa para leer la base y la altura de un triángulo e imprimir su superficie. El algoritmo debe validar los datos de entrada, verificando que éstos sean números positivos. En caso contrario debe imprimirse el dato erróneo junto con una leyenda aclaratoria. Se recuerda que $\text{Sup} = (\text{Base} * \text{Altura}) / 2$.
- Ejercicio 6:** Una editorial determina el precio de un libro según la cantidad de páginas que contiene. El costo básico del libro es de \$125, más \$2,20 por página con encuadernación rústica. Si el número de páginas supera las 300 la encuadernación debe ser en tela, lo que incrementa el costo en \$80. Además, si el número de páginas sobrepasa las 600 se hace necesario un procedimiento especial de encuadernación que incrementa el costo en \$136. Desarrollar un programa que calcule el costo de un libro dado el número de páginas.
- Ejercicio 7:** Una empresa aplica el siguiente procedimiento en la comercialización de sus productos:
- Aplica el precio base a la primera docena de unidades.
 - Aplica un 10% de descuento a todas las unidades entre 13 y 100.
 - Aplica un 25% de descuento a todas las unidades por encima de las 100.
- Por ejemplo, supongamos que vende 230 unidades de un producto cuyo precio base es 100. El cálculo resultante sería:
- $100 * 12 + 90 * 88 + 75 * 130 = 18870$, y el precio promedio será $18870 / 230 = 82,04$*
- Desarrollar un programa que lea la cantidad solicitada de un producto y su precio base, e imprima el valor total de la venta y el precio promedio.
- Ejercicio 8:** Una empresa factura a sus clientes el último día de cada mes. Si el cliente paga su factura dentro de los primeros 10 días del mes siguiente, tiene un descuento de \$120 o del 2% de la factura, lo que resulte menor. Si paga en los siguientes 10 días del mes deberá pagar el importe original de la factura, mientras que si paga después del día 20 deberá abonar una multa de \$150 o del 10% de su factura, lo que resulte mayor. Desarrolle un programa que lea el número del cliente y el total de la factura, y emita un informe donde conste el número del cliente y los tres importes que podrá abonar según la fecha de pago.
- Ejercicio 9:** Leer un número correspondiente a un año e imprimir un mensaje indicando si es bisiesto o no. Se recuerda que un año es bisiesto cuando es divisible por 4. Sin embargo, aquellos años que sean divisibles por 4 y también por 100 no son bisiestos, a menos que también sean divisibles por 400. Por ejemplo, 1900 no fue bisiesto pero sí el 2000.

- Ejercicio 10:** Desarrollar un programa para leer las longitudes de los tres lados de un triángulo L1, L2, L3 y determinar qué tipo de triángulo es según la siguiente clasificación:
- Si $A \geq B + C$ no se trata de un triángulo.
 - Si $A^2 = B^2 + C^2$ se trata de un triángulo rectángulo.
 - Si $A^2 > B^2 + C^2$ se trata de un triángulo obtusángulo.
 - Si $A^2 < B^2 + C^2$ se trata de un triángulo acutángulo.

Tener en cuenta que A denota el mayor de los lados L1, L2 y L3, mientras que B

y C corresponden a los dos lados restantes.

Ejercicio 11: La fecha de Pascua para un año cualquiera **X** puede determinarse mediante el siguiente algoritmo:

- Calcular el resto de dividir X sobre 19 y llamarlo A.
- Calcular el resto de dividir X sobre 4 y llamarlo B.
- Calcular el resto de dividir X sobre 7 y llamarlo C.
- Multiplicar A por 19, sumarle 24 y calcular el resto de dividir este resultado por 30. Este resto recibirá el nombre D.
- Multiplicar B por 2, C por 4 y D por 6. Sumar los tres resultados y sumarle 5. Calcular el resto de dividir este último resultado por 7 y llamarlo E.
- La fecha de Pascua se obtiene sumando los valores D y E, más la constante 22.
- El resultado se expresa como una fecha dentro del mes de Marzo. Si el número es mayor que 31, entonces ese año Pascua se celebrará en el mes de Abril. Ejemplo: Un resultado 35 significa que Pascua cae el 4 de Abril (35 menos los 31 días del mes de Marzo es igual a 4).

Preparar un programa que permita ingresar el año y calcule la fecha de Pascua.

Ejercicio 12: Leer tres números correspondientes al día, mes y año de una fecha e imprimir un mensaje indicando si la fecha es válida o no.

Ejercicio 13: Diseñar un programa que calcule y muestre el sueldo neto de un empleado en base a su sueldo básico y su antigüedad en años. Si es soltero se le incrementa el sueldo en 5% del salario bruto por cada año de antigüedad, mientras que si es casado se le incrementa el sueldo en 7% del bruto por cada año de antigüedad. También se le realizan los siguientes descuentos: Jubilación: 11%, Obra Social: 3%, Sindicato: 3%

Como datos de entrada se ingresa por teclado el sueldo básico, antigüedad y estado civil ('s' o 'c'). Se debe informar: (reemplazar los 9 por los valores que correspondan)

Estado Civil: Soltero/Casado

Sueldo básico	Antigüedad	Descuentos	Importe
\$ 999.99	99 años		+ 999.99
	Jubilación	- 999,99	
	Obra Social	- 999,99	
	Sindicato	- 999,99	

	Sueldo Neto		999,99

FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS
Departamento de Tecnología Informática
Fundamentos de Informática



Trabajo Práctico 4: Estructura iterativa

Ejercicio 1: Realizar un programa para ingresar desde el teclado un conjunto de números. Mostrarlos a medida que se los ingresa. Finalizar la lectura de datos con el valor -1.

Ejercicio 2: Realizar un programa para ingresar desde el teclado un conjunto de números. Al finalizar mostrar por pantalla el primer y último elemento ingresado. Finalizar la lectura con el valor -1.

Ejercicio 3: Realizar un programa para ingresar desde el teclado un conjunto de números e informar si la cantidad de elementos es impar o par, sin utilizar contadores. Finalizar la lectura de datos con el valor -1.

Ejercicio 4: Realizar un programa para ingresar desde el teclado un conjunto de números e informar el último elemento ingresado en una posición par. Finalizar la lectura de datos con el valor -1.

datos con el valor -1.

Ejemplos:

Si la secuencia es 3 7 4 5 6 7 9 -1 el valor a informar es 7

Si la secuencia es 3 7 4 5 -1 el valor a informar es 5

- Ejercicio 5:** Realizar un programa para ingresar desde el teclado un conjunto de números e informar los elementos ingresados menores a un valor ingresado previamente. Finalizar la lectura de datos con el valor -1.
- Ejercicio 6:** Realizar un programa para ingresar desde el teclado un conjunto de números e informar en forma separada la cantidad de elementos pares e impares ingresados. Finalizar la lectura de datos con el valor -1.
- Ejercicio 7:** Realizar un programa para ingresar desde el teclado un conjunto de números y mostrar por pantalla el menor y el mayor de ellos. Finalizar la lectura de datos con un valor -1.
- Ejercicio 8:** Desarrollar un programa que imprima los números naturales comprendidos entre 1 y 300.
- Ejercicio 9:** Leer cien números e imprimir su promedio.
- Ejercicio 10:** Desarrollar un programa que imprima los números impares comprendidos entre 1 y 1000.
- Ejercicio 11:** Leer diez números enteros e imprimir el mayor.
- Ejercicio 12:** Leer dos números A y B enteros positivos. Calcular el producto $A * B$ por sumas sucesivas e imprimir el resultado. Ejemplo: $4 * 3 = 4 + 4 + 4$ (4 sumado 3 veces).
- Ejercicio 13:** Leer diez números e imprimir el menor de ellos, indicando además el número de orden con que fue leído.
- Ejercicio 14:** Escribir un programa que imprima los múltiplos de 7 hasta el 3000.
- Ejercicio 15:** Leer diez números e imprimir el mayor, el menor y el rango del conjunto (El rango de un conjunto se calcula restando el mayor menos el menor).

FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS
Departamento de Tecnología Informática
Fundamentos de Informática



- Ejercicio 16:** Dadas las siguientes series numéricas, desarrollar programas que muestren los primeros 20 términos de cada una de ellas.
- a. 1, 1, 2, 3, 4, 9, 8, 27, 16...
es decir $2^0, 3^0, 2^1, 3^1, 2^2, 3^2, 2^3, 3^3...$
 - b. -1, 2, -3, 4, -5, 6, -7...
es decir $1*(-1)^1, 2*(-1)^2, 3*(-1)^3, 4*(-1)^4, 5*(-1)^5, 6*(-1)^6, 7*(-1)^7...$
 - c. 1, 2, 4, 7, 11, 16, 22, 29...
es decir 1, (1+1), (2+2), (4+3), (7+4), (11+5), (16+6), (22+7)...
(el primer término es 1 y cada nuevo término se obtiene sumando el término anterior más el número de orden del término actual)
- Ejercicio 17:** Para las mismas series del ejercicio anterior, informar el término de mayor valor que sea menor a un tope ingresado previamente. Ejemplo: Para las series a), b) y c), si se ingresa el valor 10 como tope se deberá informar 9, 10 y 7 respectivamente.
- Ejercicio 18:** Para las mismas series y topes del ejercicio anterior, informar el resultado de la sumatoria de los términos calculados hasta cada tope.

- Ejercicio 19:** Leer un conjunto de números que representan edades de un grupo de personas, finalizando la lectura cuando se ingrese el número 999. Imprimir cuántos son menores de 18 años, cuántos tienen 18 años o más y el promedio de edad de ambos grupos.
- Ejercicio 20:** Leer A y B enteros positivos y verificar que $A \geq B$. Una vez hecha esta verificación, dividir A sobre B mediante restas sucesivas, es decir sin utilizar el operador de división. Ejemplo 5 dividido 2:
- Restamos el dividendo menos el divisor: $5 - 2 = 3$
 - Repetimos esta resta tantas veces como sea posible: $3 - 2 = 1$
 - Al ser el resultado (1) menor que el divisor (2), detenemos el proceso.
 - Este resultado es el resto de la división, mientras que el cociente se obtiene contando la cantidad de restas que se efectuaron. Imprimir diviendo, divisor, cociente y resto.
- Ejercicio 21:** Leer un número natural N. Calcular e imprimir los números naturales pares menores que N.
- Ejercicio 22:** Leer un número natural N. Calcular e imprimir los primeros N números naturales impares.
- Ejercicio 23:** Leer un número natural M. Calcular e imprimir:
- La sumatoria de los números naturales menores o iguales que M.
 - La productoria de los números naturales mayores o iguales que M y menores que $M*2$.
- Ejercicio 24:** El factorial de un número entero N mayor que cero se define como el producto de todos los enteros X tales que $0 < X \leq N$. Desarrollar un programa para calcular el factorial de un número dado. Deberán rechazarse las entradas inválidas (menores que 0).

- Ejercicio 25:** Leer dos números naturales A y B. Calcular A^B mediante productos sucesivos y mostrar el resultado. Tener en cuenta que A y B pueden ser nulos.
- Ejercicio 26:** Calcular e imprimir la suma de los números enteros comprendidos entre dos números enteros A y B ingresados por teclado. Tener en cuenta que A puede ser mayor, menor o igual que B.
- Ejercicio 27:** Leer un número entero y mostrar un mensaje informando cuántos dígitos contiene. Ejemplo: Si se ingresa 12345 debe imprimir 5.
- Ejercicio 28:** Leer un número entero e invertir las cifras que contiene. Imprimir por pantalla el número invertido. Por ejemplo, si se ingresa 1234 debe mostrar 4321.
- Ejercicio 29:** Juancito está descontento con su rendimiento en las clases de Programación. En su primer programa cometió un error, en el segundo dos, en el tercero cuatro, en el cuarto ocho y así sucesivamente. Las clases duran S semanas y debe realizar dos programas semanales. Diseñar un programa que lea S y calcule el número de errores que Juancito debe esperar cometer en su último programa, si mantiene constante su rendimiento actual. Resolver este problema utilizando dos estrategias distintas.
- Ejercicio 30:** Leer un número natural N. Calcular e imprimir los primeros N términos de la sucesión geométrica de razón 3, cuyos primeros términos son 1, 3, 9, 27, 81.... es decir $3^0, 3^1, 3^2, 3^3, \dots$
- Ejercicio 31:** Realizar un programa que lea un número natural H e imprima un mensaje indicando si H es primo o no. Se dice que un número es primo cuando sólo es divisible por sí mismo y por la unidad.
- Ejercicio 32:** La sucesión de Fibonacci es una sucesión de números enteros donde cada térmi-

Ejercicio 32: La sucesión de Fibonacci es una sucesión de números enteros donde cada término no se obtiene como suma de los dos anteriores, siendo los dos primeros 1 y 1. Por lo tanto, Fib=1, 1, 2, 3, 5, 8, 13, 21.... Realizar un programa que lea N e imprima los N primeros términos de esta sucesión, como así también la suma de los mismos.

Ejercicio 33: La raíz cuadrada de un número positivo n puede calcularse mediante la siguiente fórmula de Newton:

$$\sqrt{n} \approx \frac{\frac{n}{a} + a}{2}$$

donde a es una aproximación a \sqrt{n} . Al aplicar repetidamente esta fórmula reemplazando a por la aproximación obtenida en el paso anterior, se obtiene cada vez una aproximación mejor. Desarrollar un programa que calcule la raíz cuadrada aproximada de un número entero positivo n , utilizando como primera aproximación a $n/2$. Detener el proceso cuando la diferencia entre dos cálculos sucesivos sea menor a 0,0001.

Ejercicio 34: Leer tres números D, M y A correspondientes al día, mes y año de una fecha, y un número entero N que representa una cantidad de días. Realizar un programa que calcule e imprima la nueva fecha que resulta de sumar N días a la fecha dada. Tener en cuenta los años bisiestos tal como se detalla en el ejercicio 9 de la práctica 3.

FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS
Departamento de Tecnología Informática
Fundamentos de Informática



Ejercicio 35: Se realizó una encuesta entre 100 consumidores. Por cada persona interrogada se ingresan dos valores: El primero indica la aceptación o no del producto A, mediante un 1 (acepta) o un 0 (no acepta). El segundo valor del par corresponde la producto B. Por ejemplo, el par (1,0) señala que el encuestado acepta el producto A pero no el B. Se solicita informar el porcentaje de consumidores que aceptan:

- El producto A.
- El producto B.
- El producto A solamente.
- El producto B solamente.
- Ninguno de los dos.
- Ambos productos.

Ejercicio 36: Por cada empleado de una empresa se leen tres datos: N° de legajo, sueldo básico y antigüedad en la empresa. Emitir un listado que informe el número de legajo y el salario de los empleados, calculando el salario de la siguiente forma: Al sueldo básico se le debe sumar un 5% por año de antigüedad, agregando un 25% adicional si la misma supera los 10 años. El lote de datos finaliza cuando se ingresa un 0 (cero) como número de legajo.

Ejercicio 37: Una empresa cuenta con 100 empleados, divididos en tres categorías A, B y C. Por cada empleado se lee su legajo, categoría y salario. Se solicita elaborar un informe que contenga:

- Importe total de salarios pagados por la empresa.
- Cantidad de empleados que ganan más de \$20000.
- Cantidad de empleados que ganan menos de \$5000, cuya categoría sea "C".
- Legajo del empleado que más gana.
- Sueldo más bajo.
- Importe total de sueldos por cada categoría.
- Salario promedio.

Ejercicio 38: Desarrollar un programa que lea un número N entero positivo y genere los ele-

mentos correspondientes a la conjetura de Ullman (en honor al matemático S. Ullman), que consiste en lo siguiente:

- Si N es par, dividirlo por 2
- Si es impar multiplicarlo por 3 y sumarle 1.
- Utilizar este resultado como nuevo número N y repetir el proceso.
- Al final se obtendrá el número 1, independientemente del número inicial. Por ejemplo, cuando el entero inicial es 26 la secuencia queda como 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

El programa deberá informar también la cantidad de términos obtenidos.

FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS
Departamento de Tecnología Informática
Fundamentos de Informática



Trabajo Práctico 5: Funciones

Codifique funciones apropiadas para cumplir con los siguientes objetivos:

Ejercicio 1: Dados dos parámetros numéricos, calcular y devolver el resultado de la multiplicación de ambos utilizando sólo sumas.

Ejercicio 2: Dados dos parámetros enteros A y B, obtener A^B (A elevado a la B) mediante multiplicaciones sucesivas, utilizando la función del ejercicio 1.

Ejercicio 3: Imprimir una columna de asteriscos, donde su altura se recibe como parámetro.

Ejercicio 4: Verificar si un número es par o impar, devolviendo True o False respectivamente.

Ejercicio 5: Devolver el máximo entre dos números recibidos como parámetros.

Ejercicio 6: Devolver True si el número entero recibido como primer parámetro es múltiplo del segundo, o False en caso contrario. Ejemplo: `esmultiplo(40,8)` devuelve True y `esmultiplo(50,3)` devuelve False.

Ejercicio 7: Dado un número entero, calcular su factorial. Ejemplo: `fact(4) = 4*3*2*1 = 24`.

Ejercicio 8: Calcular el Máximo Común Divisor de dos enteros no negativos, basándose en las siguientes fórmulas matemáticas:

- $MCD(X,X) = X$
- $MCD(X,Y) = MCD(Y,X)$
- Si $X > Y \Rightarrow MCD(X,Y) = MCD(X-Y,Y)$.

Ejemplo: `MCD(40,15)` devuelve 5.

Ejercicio 9: Desarrollar la función `signo(n)`, que reciba un número entero y devuelva un 1, -1 o 0 según el valor recibido sea positivo, negativo o nulo.

Ejercicio 10: Escribir la función `comparar(a,b)` que reciba como parámetros dos números enteros y devuelva 1 si el primero es mayor que el segundo, 0 si son iguales o -1 si el primero es menor que el segundo. Ejemplo: `comparar(4,2)` devuelve 1, y `comparar(2,4)` devuelve -1.

Ejercicio 11: Adaptar el programa que utiliza la fórmula de Newton para calcular la raíz cuadrada de un número positivo N (de la práctica anterior) para que trabaje como una función.

- Ejercicio 12:** Extraer un dígito de un número entero. La función recibe como parámetros dos números enteros, uno será del que se extraiga el dígito y el otro indica qué cifra se desea obtener. La cifra de la derecha se considera la número 0. Retornar el valor -1 si no existe el dígito solicitado. Ejemplo: `extraerdigito(12345,1)` devuelve 4, y `extraerdigito(12345,8)` devuelve -1.
- Ejercicio 13:** Devolver los últimos N dígitos de un número entero pasado como parámetro. El valor de N también debe ser pasado como parámetro. Devolver el número completo si N es demasiado grande. Ejemplo: `ultimosdigitos(12345,3)` devuelve 345, y `ultimosdigitos(12345,8)` devuelve 12345.

FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS
Departamento de Tecnología Informática
Fundamentos de Informática



- Ejercicio 14:** Obtener el dígito central de un número entero pasado como parámetro, sólo si la cantidad de dígitos es impar. Si la longitud fuera par devolver -1. Ejemplo: `digitocentral(12345)` devuelve 3, y `digitocentral(123456)` devuelve -1.
- Ejercicio 15:** Escribir una función que reciba como parámetros dos números enteros y devuelva la concatenación de ambos. Por ejemplo `concatenar(123,456)` devuelve 123456.
- Ejercicio 16:** Realizar una función que calcule y devuelva la sumatoria de los términos de la sucesión de Fibonacci entre dos números de término dados, los que se reciben como parámetros.

Trabajo Práctico 6: Listas

Desarrollar los siguientes programas o funciones, según corresponda:

Ejercicio 1: Escribir una función para ingresar desde el teclado una serie de números entre 1 y 20 y guardarlos en una lista. En caso de ingresar un valor fuera de rango el programa mostrará un mensaje de error y solicitará un nuevo número. Para finalizar la carga se deberá ingresar -1. La función no recibe ningún parámetro, y devuelve la lista cargada (o vacía, si el usuario no ingresó nada) como valor de retorno.

En los siguientes ejercicios utilice la función del ejercicio 1 para ingresar datos en una lista y:

Ejercicio 2: Calcular la suma de los números de una lista.

Ejercicio 3: Determinar si una lista es capicúa.

Ejercicio 4: Invertir aquellos valores ubicados en posiciones impares de una lista.

Ejercicio 5: Construir una nueva lista llamada SECUENCIAS a partir de los datos de la lista original, en la que los valores se encuentren separados en secuencias cuya suma no sea mayor que 20. Agregar un elemento de valor 0 para separar cada secuencia de la siguiente y un 0 adicional al final de la lista. Mostrar la nueva lista por pantalla. Ejemplo:

Lista original:

5	2	9	6	4	15	3	19	12	1	5
---	---	---	---	---	----	---	----	----	---	---

Resultado:

5	2	9	0	6	4	0	15	3	0	19	0	12	1	5	0
---	---	---	---	---	---	---	----	---	---	----	---	----	---	---	---

Ejercicio 6: A partir de la lista generada en el ejercicio anterior con secuencias de números, colocar en la lista SECMA SLARGA la secuencia más larga de SECUENCIAS. Si hubiera varias secuencias con la misma longitud máxima bastará con colocar cualquiera de ellas. Imprimir la secuencia más larga obtenida.

Ejercicio 7: Escribir una función para devolver la posición que ocupa un valor pasado como parámetro, utilizando búsqueda secuencial en una lista desordenada. La función debe devolver -1 si el elemento no se encuentra en la lista.

Ejercicio 8: Ídem anterior, utilizando búsqueda binaria sobre una lista ordenada.

Ejercicio 9: Una escuela necesita conocer cuántos alumnos cumplen años en cada mes del año, con el propósito de ofrecerles un agasajo especial en su día. Desarrollar un programa que lea el número de legajo y fecha de nacimiento (día, mes y año) de cada uno de los alumnos que concurren a dicha escuela. La carga finaliza con un número de legajo igual a -1. Emitir un informe donde aparezca -mes por mes- cuántos alumnos cumplen años a lo largo del año. Imprimir también una leyenda que indique cuál es el mes con mayor cantidad de cumpleaños.

Ejercicio 10: Leer una lista de números e imprimir el valor mínimo y el lugar que ocupa. Tener en cuenta que el mínimo puede estar repetido, en cuyo caso deberán mostrarse todas las posiciones que ocupe. La carga de datos termina con -1.

Ejercicio 11: Dado una lista de N números (por ejemplo 5), devolver una lista de N-1 valores booleanos, tal que cada valor de este último arreglo corresponde al resultado de la comparación de los pares de valores consecutivos del primer arreglo. El valor booleano es **True** si el primer elemento del par es menor o igual que el siguiente, y **False** si no lo es. Ejemplo:

Lista original:

4	2	8	8	6
---	---	---	---	---

Resultado:

F	T	T	F
---	---	---	---

Ejercicio 12: Cargar dos listas de números A y B. Se solicita construir e imprimir tres nuevas listas C, D y E que contengan:

- La concatenación de los valores pares de A con los impares de B.*
- La concatenación de los valores pares de A con el reverso de los valores pares de B.*
- La intercalación de los elementos de A y B.

* Tener en cuenta que cuando se indica "valores pares" o "valores impares" se hace referencia a los elementos propiamente dichos y no a sus posiciones.

Ejercicio 13: Dada una lista ordenada de números llamada A y un nuevo número N, desarrollar un programa que agregue el elemento N dentro de la lista A, respetando el ordenamiento existente. El programa deberá detectar automáticamente si el ordenamiento es ascendente o descendente antes de realizar la inserción.

Ejercicio 14: Leer una lista de números V. Luego se solicita:

- Calcular el producto de los elementos de subíndice par y dividirlo por la suma de los elementos de subíndice impar, sólo si esta suma es distinta de cero. Imprimir la lista leída y el resultado calculado, o un mensaje de error en caso de no poder realizar la operación.
- Generar e imprimir otra lista tal que su primer elemento contenga la suma del primero más el último elemento de la lista V; el segundo elemento contenga la suma del segundo más el penúltimo de V, etc. La nueva lista contendrá la mitad de los elementos de la lista original.
- Imprimir un listado de aquellos elementos de V que cumplan con la condición de tener iguales sus dos elementos laterales (el anterior y el siguiente). Si ninguno cumple esta condición, se imprimirá una leyenda aclaratoria. Considerar que los extremos de la lista se encuentran unidos, de modo que el último elemento se encuentra antes que el primero, y que el primer elemento se encuentra después del último.

-
- Ejercicio 15:** Realizar un programa que permita ingresar números en una lista, finalizando la lectura con -1. Informar si la secuencia de elementos ingresada es ascendente, descendente, todos sus valores son iguales o se encuentra desordenada.
- Ejercicio 16:** Eliminar de una lista de números enteros los valores que se encuentren en una segunda lista. Imprimir la lista original, la lista de valores a eliminar y la lista resultante.
- Ejercicio 17:** Leer dos listas de números M y N, ambas ordenadas de menor a mayor. Generar e imprimir una tercera lista que resulte de intercalar los elementos de M y N. La nueva lista también debe quedar ordenada, sin utilizar ningún método de ordenamiento.