

Trabajo práctico complementario (repaso para parcial)

Consejos para la resolución

- Pensar el algoritmo antes de empezar a programar. Por ejemplo: “para resolver este problema se necesita hacer una repetición y por cada iteración debo hacer determinado procesamiento”.
- Anotar, dibujar, bosquejar los elementos, variables, estructuras que tendrá el programa, para hacerse una idea más concreta y visual.
- Una vez hecha parte de un algoritmo, seguirla como si fuéramos el intérprete, viendo qué parte del código se ejecuta en cada momento, dándole valores a las variables, probando con casos límite (por ejemplo, evaluando qué pasa con el elemento final de una lista).
- Escribir el código por partes (que tengan sentido), dejando temporalmente de lado lo que pueda resolverse con posterioridad. Por ejemplo: “aquí habrá una selección múltiple y, por cada opción, se llamará a distintas funciones que se desarrollarán más tarde, pero primero escribo la sentencia de selección con sus opciones”.
- A veces resulta más fácil plantear una iteración si primero se escriben las instrucciones para lo que se quiere hacer en cada paso de la iteración, sin preocuparse en cómo empieza o cuáles son las condiciones para la terminación de la misma. Una vez planteado el código para cada paso de la iteración, las condiciones de corte deberían ser más evidentes, al igual que lo que se necesita hacer antes de comenzar a iterar. Por ejemplo: “para encontrar el mínimo valor de una secuencia, necesito comparar el mínimo que encontré hasta el momento con el valor actual: si es menor, reemplazo al mínimo con el valor actual. Corto cuando no tenga más elementos y, para comenzar, debo tener un mínimo valor inicial (que podría ser el primer elemento de la secuencia)”.
- Utilizar funciones cuando sea necesario. Además, si un mismo algoritmo se escribe en dos o más partes diferentes del programa, dicho algoritmo es candidato a separarse en una nueva función. Por ejemplo: “para saber si debo procesar un dato, primero necesito saber si tiene determinadas características, entonces hago una función que me devuelva True si el dato tiene esas características”.

Ejercicios

- 1) Pedir al usuario que ingrese una secuencia de strings de cualquier longitud, una por una (es decir, cada string por separado). Los strings ingresados por el usuario pueden contener letras (mayúsculas o minúsculas), números o símbolos. La lectura debe finalizar cuando el usuario ingrese un string *que contenga* al número 0. Los strings no deben almacenarse en ningún contenedor sino que se procesan con cada ingreso.
 - a) Por cada string que se procese, se deberá informar si contiene algún símbolo (carácter que no es un número ni una letra). De ser así, se debe informar una sola vez por cada string.
 - b) Informar cuántos strings en total ingresó el usuario.
 - c) Informar la cantidad de letras mayúsculas que aparecieron en todos los strings en total, incluyendo las

repetidas.

2) Una empresa de telefonía tiene planes identificados con un código alfanumérico y un precio por minuto. Necesita un programa que permita cargar la información de cada uno de esos planes y también la información de sus clientes. Los clientes están identificados con un código numérico y de ellos se deben almacenar: apellido, código del plan que contrató y minutos consumidos en el último mes. Se pide informar:

- a) Apellido del cliente que más minutos consumió en el último mes.
- b) Listado de todos los clientes, mencionando código identificador y precio a pagar por lo consumido en el último mes. Si un cliente de la lista tiene asignado un código de plan inválido, excluirlo del listado.

3) El bar de Moe, en Springfield, requiere un programa que registre los consumos y pagos de los clientes. Tiene 10 mesas, identificadas con números del 1 al 10. Cada vez que se atiende una mesa, se registran todas las bebidas que pidieron en esa mesa. Por otro lado, el bar tiene un listado de cada bebida con su precio. Se requiere hacer un menú con opciones para que Moe elija lo que desea hacer:

- a) Cargar el detalle de bebidas disponibles y el precio de cada una en un diccionario.
- b) Cargar los consumos de cada mesa en un diccionario cuyas claves sean los números de mesa y sus valores sean listas que contienen los nombres de las bebidas consumidas.
- c) Visualizar listado de bebidas y sus precios.
- d) Cerrar mesa: cuando los clientes de una mesa se retiran, debe imprimirse el total a pagar por todo lo consumido en la mesa y luego eliminar el detalle de consumo para que quede listo para recibir a nuevos clientes.
- e) Salir del programa.

4) Leer de teclado una secuencia de números enteros (la cantidad de dígitos de cada número puede variar), positivos o negativos. La lectura finaliza cuando se hayan leído 5 números positivos. Los números no deben almacenarse en ningún contenedor.

- a) Informar cuántos números de los que se leyeron estaban entre -60 y -3
- b) Informar cuántas ocurrencias hubieron de cada dígito del 0 al 9 (en valor absoluto, es decir, sin importar si es positivo o negativo). Por ejemplo, si la secuencia ingresada por el usuario es la siguiente: 432, 562, -27, 70, 9, -4, 9 se debe informar:

El 0 apareció 1 vez

El 1 apareció 0 veces

**INTRODUCCIÓN A LA
PROGRAMACIÓN IMPERATIVA**

Escuela de Tecnología | Área Algoritmos y Lenguajes

El 2 apareció 3 veces

...etc.

c) Informar cuántos números negativos se leyeron en total.

d) Informar si en la secuencia apareció algún número múltiplo de 14 que no es divisible por 3.

5) En los libros/películas “El Señor de los Anillos” hay varios personajes, pertenecientes a distintas razas.

Cada personaje tiene un lugar de procedencia y una edad.

Se desea almacenar esta información de los personajes en una estructura que permita buscar por nombre.

El programa debe comenzar con estos datos ya cargados:

Aragorn: raza humano, proviene de Rivendell, 88 años
Frodo: raza hobbit, proviene de La Comarca, 51 años
Gandalf: raza maiar, proviene de Valinor, 2019 años
Gimli, raza enano, proviene de Erebor, 140 años
Sam: raza hobbit, proviene de La Comarca, 39 años
Pippin: raza hobbit, proviene de La Comarca, 29 años
Arwen: raza elfo, proviene de Rivendell, 2778 años
Gollum: raza hobbit, proviene del Valle de Anduin, 589 años
Merry: raza hobbit, proviene de La Comarca, 37 años
Elrond: raza elfo, proviene de Rivendell, 6517 años

Se debe mostrar esta información al usuario y preguntarle si desea agregar más personajes, cortando la carga con una condición adecuada.

Una vez finalizada la carga, informar:

a) Cuántos personajes tienen más de 1000 años de edad.

b) Mostrar los nombres de los personajes de más de 1000 años de edad.

c)Cuál es el lugar del que proceden más personajes.

d) Cuántos personajes son de raza “hobbit”.

e) Qué edad tiene el personaje llamado “Arwen”.

6) Solicitar al usuario que ingrese por teclado nombres de países junto con el continente al que cada uno pertenece. Indicar una condición de corte apropiada y, opcionalmente, almacenar estos datos en un contenedor adecuado.

a) Informar, por cada una de las letras del abecedario (de la "a" a la "z") cuántos nombres de países

**INTRODUCCIÓN A LA
PROGRAMACIÓN IMPERATIVA**

Escuela de Tecnología | Área Algoritmos y Lenguajes

comienzan con dicha letra (debe contabilizarse indistintamente si el usuario lo escribió con letras mayúsculas o minúsculas). Por ejemplo, si la lista ingresada es: Argentina, América - Colombia, América - Congo, África - España, Europa, se deberá informar que hay 1 país cuyo nombre comienza con "a", 2 que comienzan con "c", 1 con "e" y 0 con el resto de las letras.

b) Informar cuántos países de América se ingresaron.

c) Informar cuántos países se ingresaron por cada continente. Por ejemplo: América: 2, África: 1, Europa: 1, Asia: 0, Oceanía: 0.

7) Solicitar al usuario que ingrese una secuencia de caracteres (strings de longitud 1), verificando esta situación (si el usuario ingresa un string de longitud mayor a 1, informarle este error y volver a pedirle el ingreso de un carácter hasta que el ingreso sea correcto). La lectura de datos finalizará cuando el usuario ingrese un carácter que represente a un dígito numérico (por ejemplo, "4") que no sea múltiplo de 3, el cual no debe procesarse.

a) Informar cuántos de los caracteres ingresados eran letras mayúsculas.

b) Informar qué caracteres se repitieron durante el ingreso. Por ejemplo, si la secuencia es: "g", "9", "%", "U", "3", "U", "u", "3", se informará que "3" y "U" se repitieron (notar que "U" es diferente de "u").

8) Una empresa requiere un programa para mantener un registro de stock y precios. Cada artículo se identifica con un código numérico único. Además, de cada artículo se conoce: descripción, rubro (si es artículo de limpieza, comestible, etc.) precio, stock.

a) Informar qué artículos poseen un stock menor a 10.

b) Informar cuántos artículos tienen precio entre 20.50 y 99.0.

c) Dado un rubro, aumentar el precio de todos los artículos de ese rubro en 15%.

d) Obtener el promedio de precios de todos los artículos de categoría "limpieza".