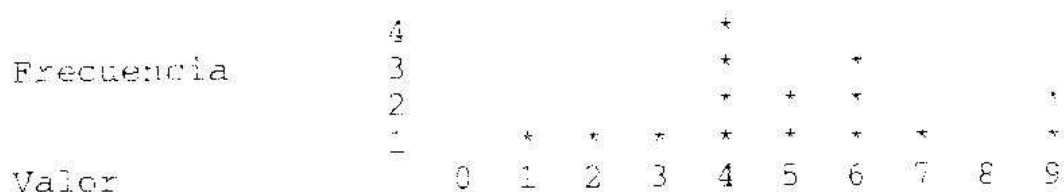


7.3.11. LABORATORIO 6: LISTAS

Objetivo

1. Familiarizarse con el uso de las listas para manipular objetos o tipos de datos
1. Realizar un programa que lea 20 números (entre el 1 y el 10), se almacenen en una lista y muestre aquel o aquellos que hayan aparecido más veces
2. Un histograma es un gráfico que muestra la frecuencia con que aparecen en una lista o arreglo dado valores dentro de subintervalos especificados de su intervalo. Por ejemplo, si un arreglo unidimensional de enteros tiene elementos de tipo 0..9 y contiene los siguientes valores: 6 4 4 1 9 7 5 6 4 2 3 9 5 6 4. Esto indica que los valores 0 y 8 no aparecen en el arreglo, los valores 1, 2, 3 y 7 aparecen una vez, el valor 5 aparece dos veces, el valor 6 tres veces y el valor 4 aparece 4 veces . El histograma sería:



Escribir un programa que, tras leer las notas de los alumnos en una asignatura, genere y visualice el histograma de las notas redondeadas a valores enteros: 0, 1, 2,...10

3. Diseñe un programa que dada una lista x cualquiera, sustituya cualquier elemento negativo por cero.
4. Diseñe un programa que lea una lista de n valores numéricos. A partir de esta lista debe generar otra solamente con los valores impares. Mostrar las dos listas.

5. Diseñe un programa que pida una lista de valores numéricos. A partir de esta lista se deben generar dos listas más: una con los valores de índice par y otra con los valores de índice impar. Mostrar las tres listas
6. Duplicaciones. Se debe pedir al usuario que digite " n " palabras y se deben insertar en una lista. Las palabras que se encuentren repetidas solo se deben insertar una vez. " n " es un número que va de 1 a 10. Se debe imprimir la lista.

Luego se le debe pedir al usuario que digite " m " frases. Se debe volver a imprimir todas las palabras de la lista junto con el número que represente la cantidad de apariciones de cada una de ellas dentro de las " m " frases. " m " es un número que va de 1 a 5.

7. Diseñe un programa que dados " n " números enteros, determine cuál de los " $n-1$ " últimos números es más cercano al primero. Por ejemplo, si el usuario introduce los números 2,6,4,1 y 10, el programa responderá que el número más cercano al 2 es el 1.
8. Diseñe un programa que dados " n " puntos en un plano, determine cuál de los " $n-1$ " últimos puntos es más cercano al primero. Un punto se representa con dos variables: una para la abcisa y otra para la ordenada. La distancia entre dos puntos $(x1, y1)$ y $(x2, y2)$ es

$$\sqrt{(x1-x2)^2+(y1-y2)^2}$$

9. Diseñe un programa que almacene en una variable x la lista obtenida con la función "`range (1,4)`" y, a continuación, la modifique para que cada componente sea igual al cuadrado del componente original. El programa mostrará la lista resultante por pantalla.
10. Diseñe un programa que almacene en una variable x la lista obtenida con la función "`range (1, n)`" , donde " n " es un entero que

se pide al usuario, y modifique dicha lista para que cada componente sea igual al cuadrado del componente original. El programa mostrará la lista resultante por pantalla.

11. Diseñe un programa que construya una lista con los “ n ” primeros números primos. No confunda con los primos entre 1 y “ n ”.

12. Búsqueda. El ejercicio consiste en solicitarle al usuario que inserte un conjunto de “ n ” nombres. Cada nombre debe irse colocando en una lista junto con un número al inicio, formando de esta manera otra lista. Se debe imprimir la lista final.

Los números inician en 1 y se van incrementando consecutivamente con cada nuevo nombre.

Por ejemplo:

[(1, “José”), (2, “María”), (3, “Moisés”), (4, “David”)]

Al terminar de crear la lista se debe realizar la búsqueda. Para ello se solicita un número y se debe devolver el nombre que representa. Si el número no corresponde a ningún nombre se le debe avisar al usuario.

Existen varias técnicas para resolver búsquedas, investigue la búsqueda binaria y otro tipo de búsqueda e implemente ambas técnicas.

13. Pulpería. Una pulpería posee pocos productos y su dueño desea un programa muy simple para poder conocer cuáles productos tiene en existencia.

Para esto se van a crear tres funciones

- **Consulta:** verifica si el producto sobre el cual se desea consultar está en existencia y si está, se devuelve la cantidad que hay.

Posee como parámetro de entrada el nombre del producto y devuelve la cantidad que haya en la pulpería. Se le debe

mostrar al usuario la cantidad en existencia o si el producto no existe dentro de la pulpería entonces el mensaje respectivo.

- **Agregar productos:** consiste en agregar productos nuevos al stock de la pulpería o comprar más productos de alguno ya existente.

Posee como parámetros de entrada el nombre del producto más la cantidad de entrada que va a ser parte del stock de la pulpería. Si el producto ya existe sólo se suma la cantidad a la que ya hay, pero si no existe entonces se agrega el producto.

Se debe devolver y mostrarle al usuario la nueva cantidad del producto junto con el nombre del mismo.

- **Vender productos:** consiste en disminuir del stock la cantidad de un producto que un cliente compra.

Su parámetro de entrada es el nombre del producto que el cliente va a comprar más la cantidad. Si el producto no existe en la pulpería se le avisa al usuario, si ya no existe más en stock también se le avisa, si existe menos cantidad de la pedida se le avisa al usuario y se le vende todo lo que hay en stock (el stock para este producto queda en 0 y también se le avisa al usuario de la cantidad comprada). Por último si se puede comprar la cantidad deseada sin ningún inconveniente se le dice al dueño de la tienda la cantidad de stock que quedó para ese producto respectivo.

Sería conveniente poder crear un menú sobre consola para que el usuario no tenga que digitar el nombre de las funciones para correr el programa.