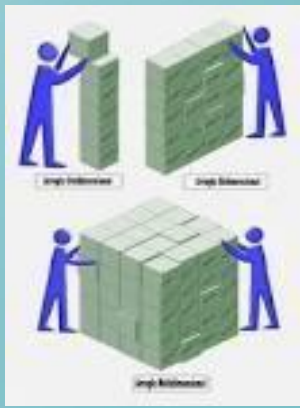
Universidad Tecnológica Nacional
Facultad Regional Resistencia
Técnico Superior en Programación

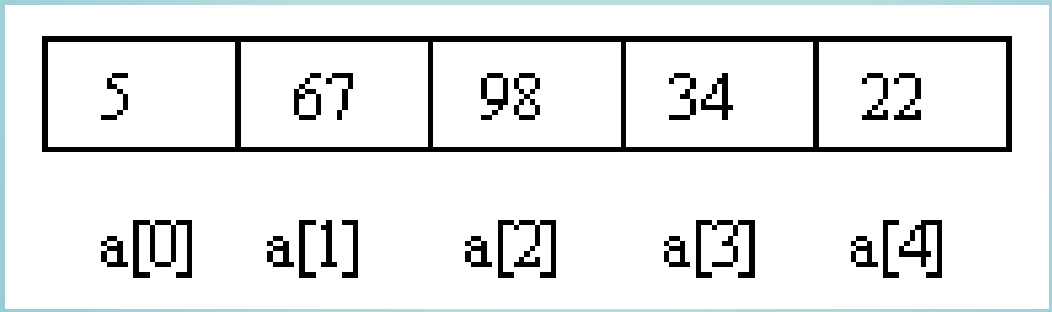
Laboratorio de Computación I

ARREGLOS



Definición

Los arreglos son una colección de variables del mismo tipo que se referencian utilizando un nombre común.



Definición

Se puede definir de una forma abstracta como ***"un conjunto finito ordenado de elementos homogéneos"*.**

Por **finito,** entendemos que hay un número especifico de elementos en el arreglo; número que debe ser grande o pequeño pero debe existir.

Por **ordenado,** entendemos que los elementos están dispuestos de tal manera que hay un elemento cero, un elemento primero, un segundo, un tercero y así sucesivamente.

Por **homogéneo,** entendemos que todos los elementos del arreglo son del mismo tipo.

Operaciones con Arreglos

- Cargar un arreglo.
- Recorrer un arreglo.
- Buscar un elemento en particular.
- Acceder a un elemento en una posición determinada.
- Acceder a una posición determinada y mostrar su contenido.
- Insertar un nuevo elemento.
- Eliminar un elemento.
- Ordenar un arreglo

Operaciones con Arreglos

- **Cargar un arreglo.**
  - **Mediante declaración del arreglo.**
  - **Usando una estructura repetitiva**

- **Recorrer un arreglo para mostrar sus elementos.**

Forma de inicializar un arreglo es la siguiente:

tipo nombre_arreglo[tamaño] = {lista-valores};

Ejemplo:

```
int x[7] = {3,5,7,-2,0,1,4};

int y[30] = {2,4,5};  // el resto de los espacios vacíos es cero

int z[3] = {1,2,3,4}  //esto es un desbordamiento ERROR
```

Int lista[9]= {0, 4, 78, 5, 32, 9, 77, 1, 23}
Posición → 0 1 2 3 4 5 6 7 8 = (9 posiciones)

```
/*Utilizacion  de arreglos*/

#include <stdio.h>
#include <conio.h>

int main()
{

        int lista[9]= {0, 4, 78, 5, 32, 9, 77, 1, 23};
        int i;

        for(i = 0; i < 9; i++)
                printf("Digito %d:%d\n",i,lista[i]);
return 0;
getch();
}
```

Cargar elementos en la declaración del arreglo

Cargar elementos en el arreglo usando estructura repetitiva.

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

int main()
{
        int lista[10];
        int i;



        for(i = 0; i < 10; i++)
                {
                        printf("Ingresar el elemento %d:\n",i);
                        scanf("%d",&lista[i]);
                }

        /*Mostrar los elementos del arreglo*/

       for(i = 0; i < 10; i++)
                printf("Elemento %d:%d\n",i+1,lista[i]);

        return 0;
       getch();
       }
```

Operaciones con Arreglos

- Buscar un elemento en particular.
- Acceder a un elemento en una posición determinada.
- Acceder a una posición determinada y mostrar su contenido.

Ejercicios de Repaso - Arreglos

- Realizar un programa que declare tres arreglos, ‘vector1’, ‘vector2’ y ‘vector3’ de cinco enteros cada uno, pida valores por teclado para ‘vector1’ y ‘vector2’ y calcule vector3=vector1+vector2. Mostrar los tres arreglos por pantalla.

- Se quiere realizar un programa que lea por teclado las 5 notas obtenidas por un alumno (comprendidas entre 0 y 10). A continuación debe mostrar todas las notas, la nota media, la nota más alta que ha sacado y la menor.

- Crear un vector de 5 enteros, inicializa el vector en etapa de declaración. Copia los elementos del vector en otro vector pero en orden inverso, y mostrar por la pantalla ambos arreglos.

Búsqueda

Un algoritmo de búsqueda es aquel que está diseñado para localizar un elemento concreto dentro de un arreglo.

Tipos de Búsqueda:

- Arreglo desordenado sin elementos repetidos.
- Arreglo desordenado con elementos repetidos.
- Arreglo ordenado sin elementos repetidos: BÚSQUEDA BINARIA
- Arreglo ordenado con elementos repetidos.

Búsqueda en un arreglo sin elementos repetidos

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

int main()
{
        int lista[9]= {0, 4, 5, 9,3, 50, 30, 81, 70};
        int i,b,enc;

        printf("Ingrese nro. a buscar\n");
        scanf("%d",&b);
        i=0;
        enc=0;

        while (i<9 && enc==0)
        {
                if (b== lista[i]) enc=1;
                else i++;   }

        if (enc==1) printf("Número encontrado \n");
        else printf("Número no encontrado \n");

        return 0;
        getch();
}
```

En el caso del programa anterior, los elementos del arreglo están desordenado.

Pregunta: Si los elementos del arreglo estarían ordenados, ¿funcionaría el programa de búsqueda?

Búsqueda en un arreglo ordenado sin elementos repetidos

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

int main()
{
        int lista[9]= {0, 4, 5, 9,32, 50, 77, 81, 93};
        int i,b,enc;

        printf("Ingrese nro. a buscar\n");
        scanf("%d",&b);
        i=0;
        enc=0;

        while (i<9 && enc==0)
        {
                if (b== lista[i]) enc=1;
                else i++;     }

        if (enc==1) printf("Número encontrado \n");
        else printf("Número no encontrado \n");

        return 0;
        getch();
}
```

ACTIVIDAD

Modificar el programa anterior para que una función realice la búsqueda.

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

int buscar (int v[],int a)
{
		int i,enc;
		i=0;
		enc=0;

		while (i<9 && enc==0)
		{
				if (a== v[i]) enc=1;
				else i++;
		}
		return enc;}
int main()
{

		int lista[9]= {0, 4, 5, 9,32, 50, 77, 81, 93};
		int b,bus;

		printf("Ingrese nro. a buscar\n");
		scanf("%d",&bus);
		b=buscar(lista,bus);

		if (b==1) printf("Número encontrado \n");
		    else printf("Número no encontrado \n");

		return 0;
		getch();
}
```

¿Qué hace este programa?

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

int main()
{

        int lista[9]= {0, 4, 5, 9,3, 50, 4, 81, 4};
        int i,b,enc;

        printf("Ingrese nro. a buscar\n");
        scanf("%d",&b);
        i=0;
        enc=0;

        while (i<9)
        {
                if (b== lista[i]) enc++;
                i++;          }

        if (enc!=0)printf("Número encontrado %d veces \n",enc);
        else printf("Número no encontrado \n");

        return 0;
        getch();
}
```

ACTIVIDAD

Modificar el programa anterior para que funcione en un arreglo ordenado con elementos repetidos como por ejemplo:

A={1,2,3,3,5,6,77,8}

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

int main()
{

        int lista[9]= {0, 4, 4, 9,10, 50, 50, 50, 94};
        int i,b,enc;

        printf("Ingrese nro. a buscar\n");
        scanf("%d",&b);
        i=0;
        enc=0;

        while (i<9)
        {
                while (lista[i]==b){
                  if (b== lista[i]) enc++;
                i++;
                }
                i++;          }

        if (enc!=0)printf("Número encontrado %d veces \n",enc);
        else printf("Número no encontrado \n");

        return 0;
        getch();
}
```

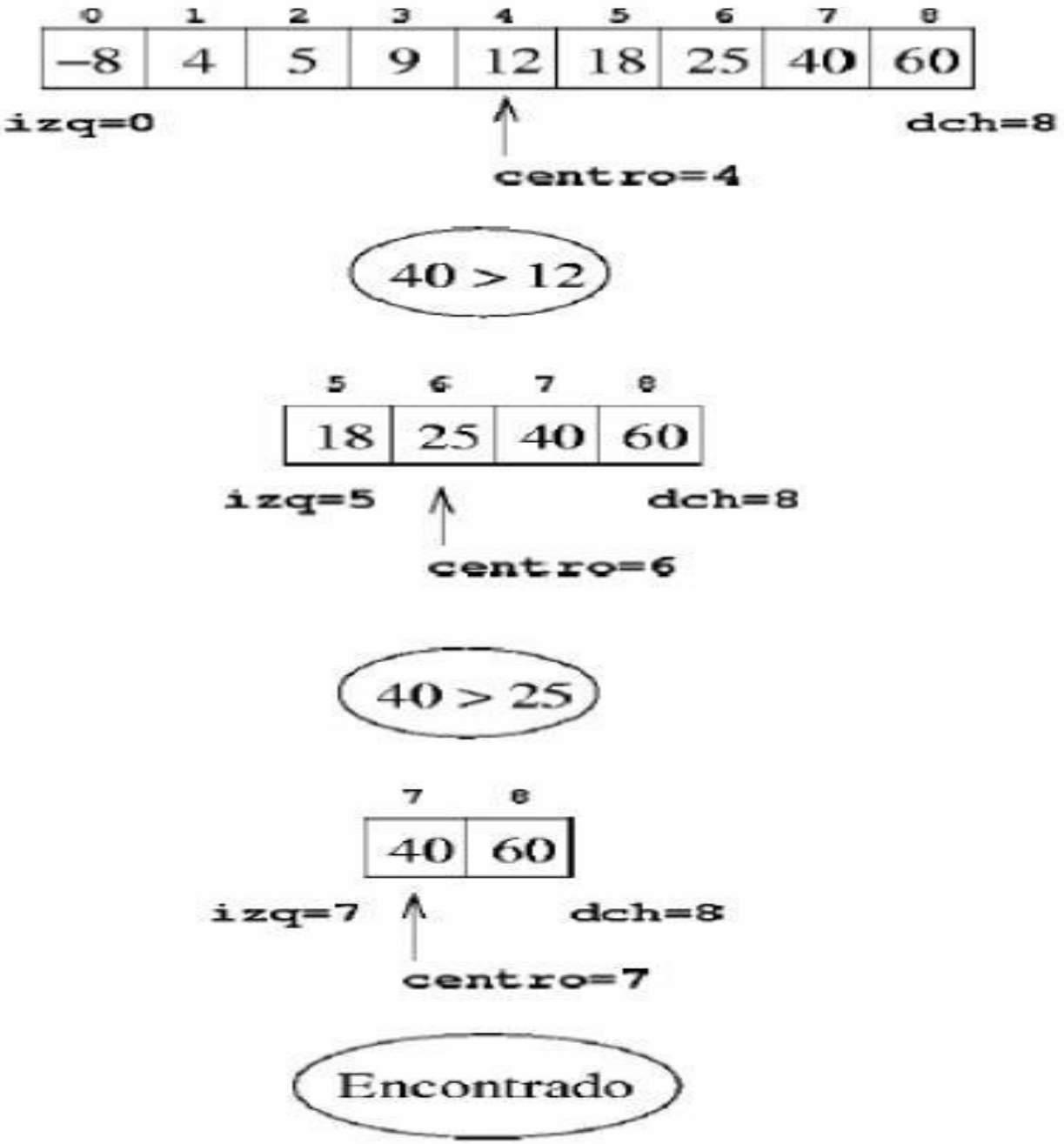
Búsqueda Binaria

La **búsqueda binaria** o **búsqueda dicotómica** es un algoritmo de búsqueda.

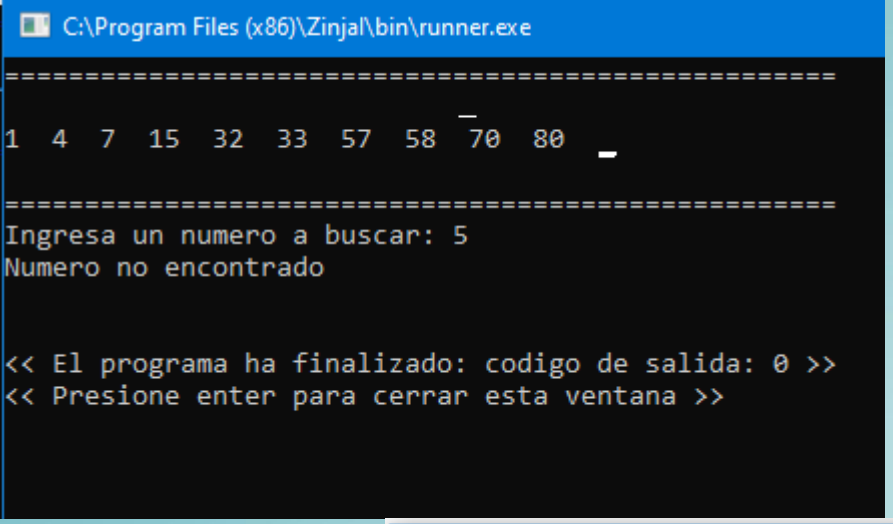
Para realizarla, es necesario contar con un arreglo ordenado.

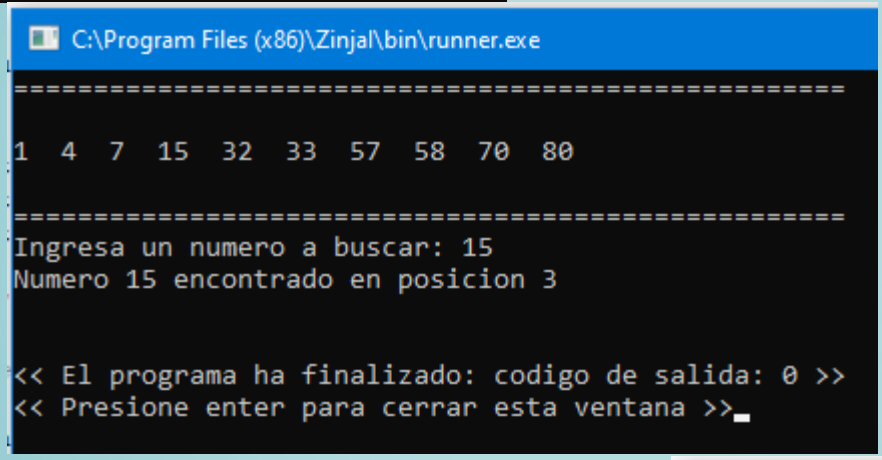
Se toma un elemento central, normalmente el elemento que se encuentra a la mitad del arreglo, y se lo compara con el elemento buscado. Si el elemento buscado es menor, se toma el intervalo que va desde el elemento central al principio, en caso contrario, se toma el intervalo que va desde el elemento central hasta el final del intervalo.

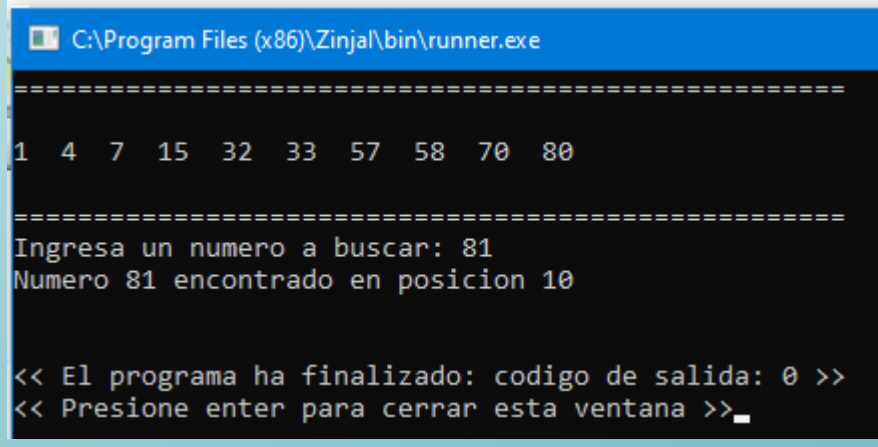
Se procede de esta manera con intervalos cada vez menores hasta que se llega a un intervalo indivisible, en cuyo caso el elemento no está en el vector, o el elemento central sea el elemento buscado.


0 1 2 3 4 5 6 7 8
−8 4 5 9 12 18 25 40 60
izq=0
dch=8
centro=4
40 > 12
5 6 7 8
18 25 40 60
izq=5
dch=8
centro=6
40 > 25
7 8
40 60
izq=7
dch=8
centro=7
Encontrado

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

int main(){
int i,buscar,mitad;  int a = 0;
int b = 10;
int contadorA = 0;
int contadorB = 0;
int lista[10]= {1, 4, 7, 15, 32,33,57,58,70,80};

printf("==================================================\n\n");
//se muestra el arreglo en pantalla
for(i = 0; i < 10; i++){
printf("%d ", lista[i]);
}
printf("\n\n==================================================\n");
printf("Ingresa un numero a buscar: ");
scanf("%d", &buscar);

while (a <= b){
                contadorA++;  mitad = (a + b) / 2;

                if(buscar == lista[mitad]){
                                printf("Numero %d encontrado en posicion %d\n", lista[mitad], mitad);  break;
                }
                else if(buscar < lista[mitad]){
                                b = mitad -1;
                }
                else{
                                a = mitad + 1;
                }
                contadorB++;
}//fin while

if(contadorA == contadorB){  printf("Numero no encontrado\n");
}
return 0;
}
```







¿Cuál es el problema en el código?
¿Pueden corregirlo?

Operaciones con Arreglos

- **Ordenar un arreglo**

Es la operación de ordenar un arreglo en algún orden de acuerdo a un criterio de ordenamiento.

Operaciones de Ordenamiento

Algoritmos de inserción:

En este tipo de algoritmo los elementos que van a ser ordenados son considerados uno a la vez.

Cada elemento es INSERTADO en la posición apropiada con respecto al resto de los elementos ya ordenados.

Algoritmos de intercambio:

En este tipo de algoritmos se toman los elementos de dos en dos, se comparan y se INTERCAMBIAN si no están en el orden adecuado. Este proceso se repite hasta que se ha analizado todo el conjunto de elementos y ya no hay intercambios.

Algoritmos de selección:

En este tipo de algoritmos se SELECCIONA o se busca el elemento más pequeño (o más grande) de todo el conjunto de elementos y se coloca en su posición adecuada. Este proceso se repite para el resto de los elementos hasta que todos son analizados.

Algoritmos de enumeración:

En este tipo de algoritmos cada elemento es comparado contra los demás. En la comparación se cuenta cuántos elementos son más pequeños que el elemento que se está analizando, generando así una ENUMERACION. El número generado para cada elemento indicará su posición.

```
#include<stdio.h>
#include<conio.h>
#define max 5

int a[5]={10,34,3,2,1};
int i,j,aux,n=max;
int main(){

		for(i=0;i<=n;i++){
				for(j=0;j<n-1;j++){
						if(a[j]>a[j+1]){
								aux=a[j];
								a[j]=a[j+1];
								a[j+1]=aux;
						}
				}
		}
		for(i=0;i<max;i++)
		{
				printf("%d-",a[i]);
		}
		return 0;
		getch();
}
```

La **Ordenación de burbuja** (**Bubble Sort** en inglés). Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada. También es conocido como el **método del intercambio directo**. Dado que solo usa comparaciones para operar elementos, se lo considera un algoritmo de comparación, siendo el más sencillo de implementar.