

CI-2125 Computación I

Práctica 7: Arreglos

Nota: Recuerde que en las prácticas se deben realizar las preguntas teóricas y desarrollar la mayor cantidad de diagramas de flujo, para que puedan realizar los laboratorios más rápido y completos. Se recomienda que adelanten los códigos por su cuenta para que los errores y las dudas sean corregidas de manera fácil en los laboratorios.

1. Realice la corrida en frío del siguiente programa y explique lo que pasa

```
main (){
    int a[4] = {0,1,2,3};
    int b[4] = {4,5,6,7};
    int i;
    for(i=0; i<8; i++) printf("a[%d] = %d \n", i, a[i]);
    return 0;
}
```

2. Diga cual será el resultado de correr los siguientes trozos de programa:

```
main (){
    int j,k;
    int primero[21] = {7,1,8,2,9,3,10,4,-1,5,-2,6};
    int segundo[21];
    for (j=0; j<6; j++) segundo[j] = primero[2*j] + j;
    for (k=3; k<7; k++) printf("%d %d\n", primero[k+1], segundo[k-1]);
}
```

3. Realice la corrida en frío de este programa y diga qué hace la función bubble.

```
#include<stdio.h>
#define N 10

void bubble(int v[]){
    int i, temp, pass;

    for ( pass = 0; pass < N - 1; pass++ ) //pases
        for ( i = 0; i < N - 1; i++ ) // un pase
            if ( v[ i ] > v[ i + 1 ] ) { // una comparación
                temp = v [ i ]; // un intercambio
                v[ i ] = v[ i + 1 ];
                v[ i + 1 ] = temp;
            }
}

int main(){
    int a[ N ] = { 2, 6, 4, 8, 10, 12, 89, 68, 45, 37 };
    int i, temp;

    printf("\nEntrada\n");
    for ( i = 0; i < N; i++ ) printf("%4d",a[ i ]);

    bubble(a);

    printf("\nSalida\n");
    for ( i = 0; i < N; i++ ) printf("%4d",a[ i ]);

    printf("\n");
    return 0;
}
```

Parte de laboratorio, recuerde que deben adelantar los diagramas de flujo.

4. Diseñe un algoritmo que llene un vector de a lo sumo 20 elementos reales e imprima la posición y el valor del elemento mayor almacenado en el vector. ¿Qué debe modificarle al algoritmo para que también calcule el menor elemento? Escriba el programa equivalente en C.
5. Diseñe e implemente un programa que permita leer dos vectores A y B de 20 números reales cada uno y muestre por pantalla como resultado el vector C que contiene la suma de A y B. Para el diseño debe usarse la técnica de análisis descendente y para la implementación deben usarse funciones.
6. Diseñe e implemente un programa que permita ingresar un número N de enteros los cuales se almacenan en un arreglo A. El número N es dado por el usuario y no puede ser mayor a 20. Luego el programa debe permitir al usuario dar un número entero B y decirle si el número B está o no contenido en el arreglo A. Debe darse la opción de buscar otro entero B siempre que el usuario lo desee. Para el diseño debe usarse la técnica de análisis descendente y para la implementación deben usarse funciones.
7. En muchas ocasiones, cuando se esta trabajando con grandes cantidades de datos almacenadas en arreglos, se podría necesitar saber si dado un elemento este se encuentra o no en dicho conjunto. Existen diferentes métodos de búsqueda en un arreglo de elementos, y dependerá si el conjunto de elementos está ordenado o no. Por ejemplo, que pasaría si las guías telefónicas no estuvieran ordenadas yuviésemos que buscar el teléfono de Pedro Guaicaipuro, en el peor de los casos, debemos buscar de principio a fin sin tener garantía de encontrarlo, esta técnica utilizada se le conoce como la técnica simple de búsqueda (búsqueda lineal) donde compara cada uno de los elementos del arreglo con el valor buscado. Esta técnica en promedio deberá comparar el valor buscado con la mitad de los elementos del arreglo y sirve para arreglos pequeños y no ordenados. Si el arreglo está ordenado, se puede utilizar una técnica de alta velocidad: la búsqueda Binaria, la cual consiste en:
 - i. consigue el índice del elemento **medio** del arreglo ,el arreglo original lo divide en dos partes, delimitadas por los índices **Inferior-Medio y Medio- Superior**. Es así como se consiguen dos subarreglos.
 - ii. Compara el valor buscado con el valor del arreglo indexado por **Medio**.
 - iii. Si el valor buscado es igual al elemento medio entonces fin del proceso, sino en función que el valor buscado sea mayor que el valor del medio o menor del valor del medio, la búsqueda se realizará en la mitad inferior o superior, y vuelve nuevamente a dividir el subarreglo en dos partes, en la parte donde se supone debe encontrarse.
 - iv. Esto se repite hasta que el subarreglo sólo tenga un elemento o no se halla encontrado

En un arreglo de 1024 elementos (2^{10}) este método en el peor de los escenarios realizará 10 comparaciones. En el método lineal en promedio realizara 512 comparaciones.

Se pide haga la función BusqBin en C, que recibe un arreglo ordenado de n posiciones y el valor a buscar, y retorne el índice del arreglo donde se encuentra y -1 si no lo consigue.