

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en Software y Tecnologías  
Emergentes Lenguaje C

práctica 6. Punteros

ALUMNO: IVAN FERNANDO HERNANDEZ CESEÑA

MATRICULA: 373077

GRUPO: 932

PROFESOR: Yulith Vanessa Altamirano

Flores 08 de noviembre del 2023

## Apuntadores y Argumentos de Funciones.

Clase	Lenguaje C
-------	------------

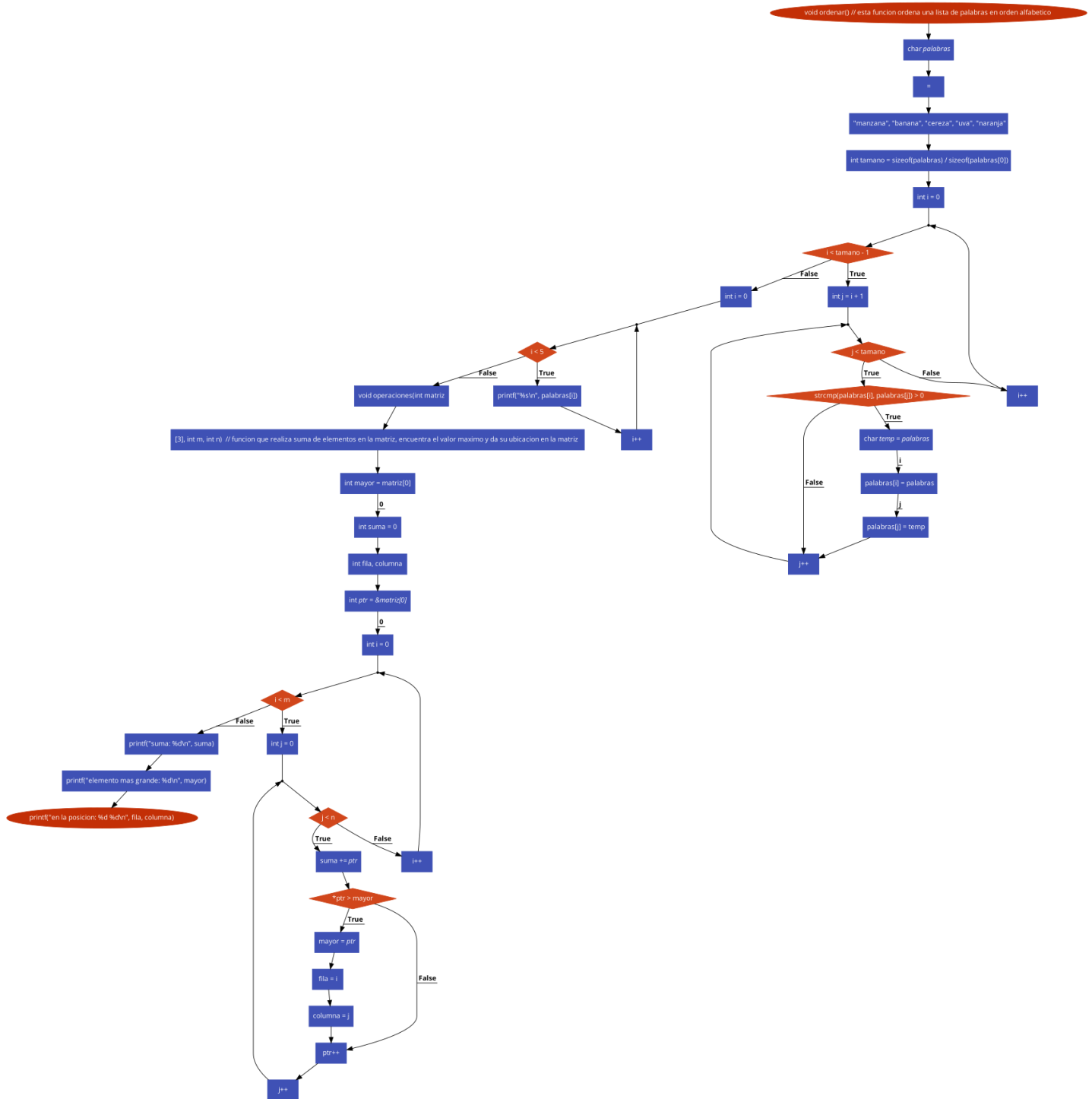
### Instrucciones

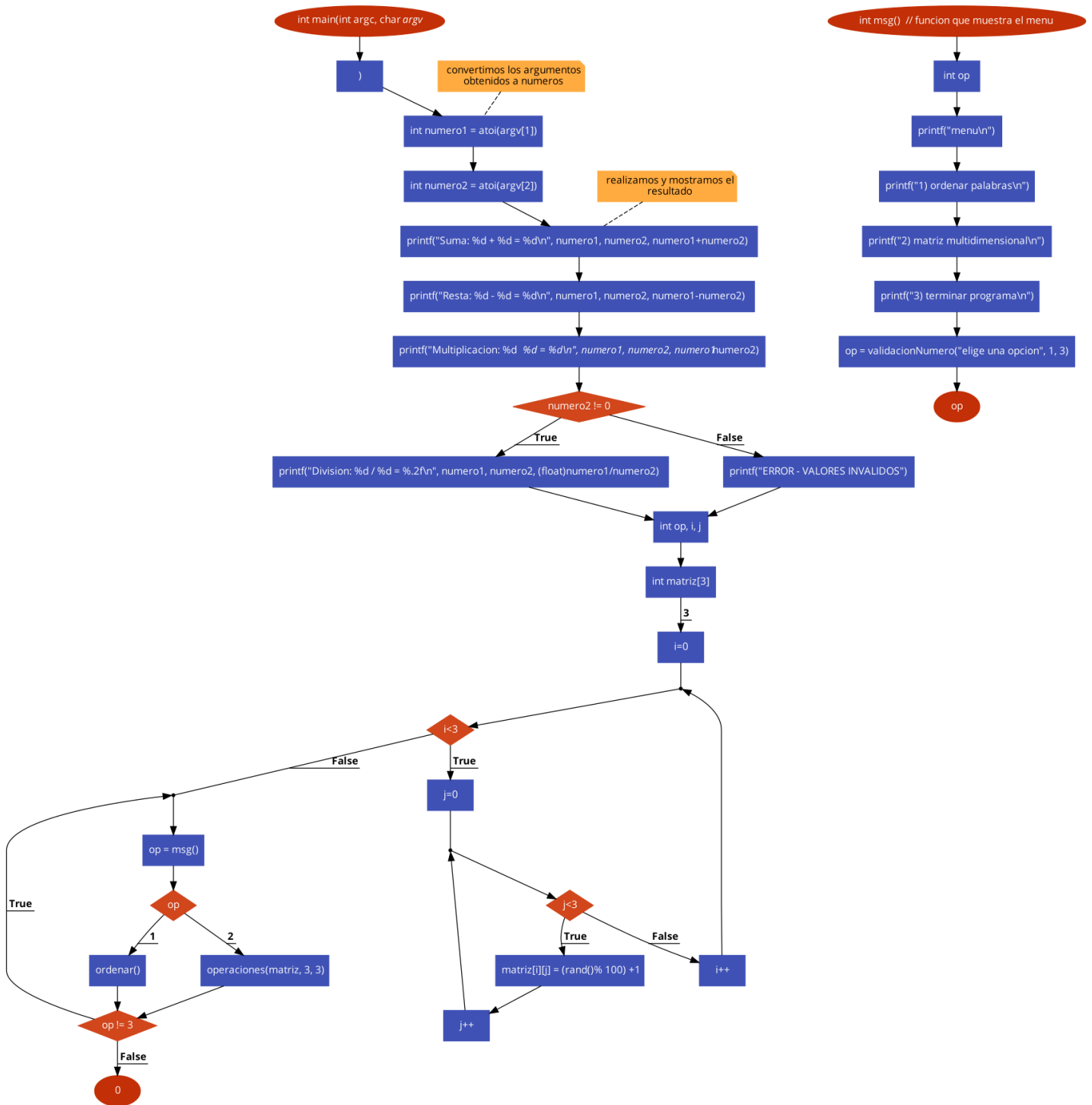
Desarrollen el código en lenguaje C y elaboren el diagrama de flujo correspondiente para los ejercicios. Será suficiente con un archivo .cpp que contenga todos los ejercicios organizados en un menú.

### Repositorio

<https://github.com/trentivan/lenguajeC.git>

# Diagrama de Flujo





# compilado

```
PS C:\Users\elvan\Downloads\uabc_3_sem\lenguaje c\lenguajeC> cd .\practica7\  
PS C:\Users\elvan\Downloads\uabc_3_sem\lenguaje c\lenguajeC\practica7> cd output  
PS C:\Users\elvan\Downloads\uabc_3_sem\lenguaje c\lenguajeC\practica7\output> ./practica7 12 5  
Suma: 12 + 5 = 17  
Resta: 12 - 5 = 7  
Multiplicacion: 12 * 5 = 60  
Division: 12 / 5 = 2.40  
menu  
1) ordenar palabras  
2) matriz multidimensional  
3) terminar programa  
elige una opcion  
1  
banana  
cereza  
manzana  
naranja  
uva  
menu  
1) ordenar palabras  
2) matriz multidimensional  
3) terminar programa  
elige una opcion  
2  
suma: 442  
elemento mas grande: 79  
en la posicion: 2 0  
menu  
1) ordenar palabras  
2) matriz multidimensional  
3) terminar programa  
elige una opcion  
3  
PS C:\Users\elvan\Downloads\uabc_3_sem\lenguaje c\lenguajeC\practica7\output> |
```

## documentado (en código)

```
// taller8.c
// Hernandez Ceseña Ivan Fernando
// 08/11/2023
// este código recibe argumentos desde la terminal, ordena un vector de
cadena en orden alfabético
// realiza suma de elementos en la matriz, encuentra el valor máximo y da su
ubicación en la matriz

#include "ivan.h"

int msg();
void ordenar();
void operaciones(int matriz[][3], int m, int n);

int main(int argc, char *argv[])
{
    // convertimos los argumentos obtenidos a números
    int numero1 = atoi(argv[1]);
    int numero2 = atoi(argv[2]);

    // realizamos y mostramos el resultado
    printf("Suma: %d + %d = %d\n", numero1, numero2, numero1+numero2);
    printf("Resta: %d - %d = %d\n", numero1, numero2, numero1-numero2);
    printf("Multiplicación: %d * %d = %d\n", numero1, numero2,
numero1*numero2);
    if (numero2 != 0)
    {
        printf("División: %d / %d = %.2f\n", numero1, numero2,
(float)numero1/numero2);
    }
    else
    {
        printf("ERROR - VALORES INVALIDOS");
    }

    int op, i, j;
```

```

int matriz[3][3];

for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
    {
        matriz[i][j] = (rand()% 100) +1;
    }
}

do{
    op = msg();
    switch(op)
    {
        case 1:
            ordenar();
            break;

        case 2:
            operaciones(matriz, 3, 3);
            break;
    }
}while(op != 3);

return 0;
}

int msg() // funcion que muestra el menu
{
    int op;
    printf("menu\n");
    printf("1) ordenar palabras\n");
    printf("2) matriz multidimensional\n");
    printf("3) terminar programa\n");
    op = validacionNumero("elige una opcion", 1, 3);
    return op;
}

```

```

}

void ordenar() // esta funcion ordena una lista de palabras en orden
alfabetico
{
    char *palabras[] = {"manzana", "banana", "cereza", "uva", "naranja"};
    int tamano = sizeof(palabras) / sizeof(palabras[0]);

    for (int i = 0; i < tamano - 1; i++)
    {
        for (int j = i + 1; j < tamano; j++)
        {

            if (strcmp(palabras[i], palabras[j]) > 0)
            {

                char *temp = palabras[i];
                palabras[i] = palabras[j];
                palabras[j] = temp;

            }

        }

    }

    for (int i = 0; i < 5; i++)
    {
        printf("%s\n", palabras[i]);
    }

}

void operaciones(int matriz[][3], int m, int n) // funcion que realiza suma
de elementos en la matriz, encuentra el valor maximo y da su ubicacion en la
matriz
{
    int mayor = matriz[0][0];
    int suma = 0;
    int fila, columna;
    int *ptr = &matriz[0][0];
    for (int i = 0; i < m; i++)
    {

```



```
    for (int j = 0; j < n; j++)
    {
        suma += *ptr;
        if (*ptr > mayor)
        {
            mayor = *ptr;
            fila = i;
            columna = j;
        }
        ptr++;
    }
}

printf("suma: %d\n", suma);
printf("elemento mas grande: %d\n", mayor);
printf("en la posicion: %d %d\n", fila, columna);
}
```