

CI-2125 Computación I

Practica 6

Contenido: Arreglos de caracteres. Matrices

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

```
main()
{
    int c;
    char nombre[4][7]={'F','u','l','a','n','o','\0',
                       'J','u','a','n','\0',' ',' ',
                       'P','e','d','r','o','\0',' ',' ',
                       'E','l','i','s','a','\0',' ',' '};
    printf ("Esto es un mensaje personalizado. \n\n");
    for (c = 0; c < 4; c++){
        printf("Hola %s \n\n", nombre[c]);
    }
}
```

SOLUCION: Escribe **Hola Fulano**, salta dos líneas, escribe **Hola Juan**, salta 2 líneas, escribe **Hola Pedro**, salta 2 líneas, finalmente **Hola Elisa** y salta 2 líneas.

2. Diseñe un algoritmo que dada una cadena de caracteres fija, imprima la cadena completa, imprima el tercer carácter de la cadena y la subcadena a partir del cuarto carácter. Escriba el programa en C equivalente.

Solucion:

ENTRADAS:

ninguna

PRECONDICIÓN:

ninguna

SALIDAS:

cadena (string)

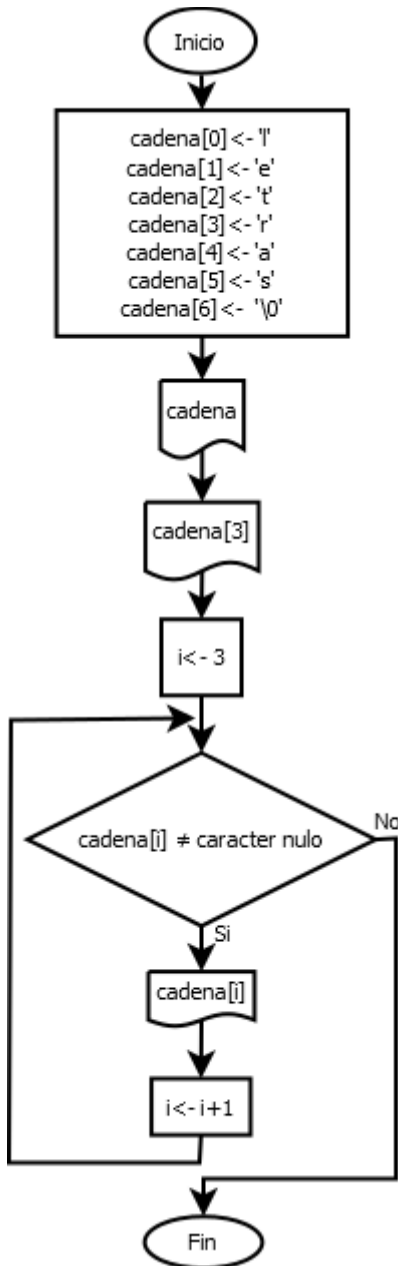
c (char)

subcadena (string)

POSTCONDICIÓN:

Se imprime en pantalla la cadena, c y subcadena, tal que

c=cadena[2] y subcadena[i] = cadena[i+4] para i entre 0 y strlen(cadena)-4



3. Se desea escribir un programa que calcula el número de días de un mes. Para ello, implemente las siguientes funciones y tipos.
- Defina un arreglo cuyos valores sean string de caracteres. Cada string corresponde a un mes del año, empezando con la posición 0 es “enero”, 1 es “febrero” y así sucesivamente.
 - Escriba una función que dado un string con el nombre del mes, busque en el arreglo y retorne el entero correspondiente a ese mes. Ejemplo, si introduce “enero”, devuelve 1, si introduce “diciembre” devuelve 12.
 - Escriba una función que dado el nombre de un mes, retorne el número de días de ese mes.
 - Qué debería hacer para que su programa considere los años bisiestos.

Solucion:

Programa Principal:

ENTRADAS: mes (string)

PRECONDICIÓN: mes debe ser válido (estar en el conjunto {"enero",...,"diciembre"})

SALIDAS: numdias (int) // número de días del mes introducido

POSTCONDICIÓN: numdias es el número de días correspondiente a mes

Funcion Buscarmes(Meses,mes)

ENTRADAS:

Meses (arreglo [12] de string), parámetro por referencia

mes (string) , parámetro por referencia

PRECONDICIÓN: Meses almacena el conjunto {"enero",...,"diciembre"} y mes debe estar en ese conjunto

SALIDA: retorna un entero

POSTCONDICIÓN: Buscarmes devuelve el número correspondiente a mes

Funcion diasmes(Meses,días,mes)

ENTRADAS:

Meses (arreglo de string), parámetro por referencia

días (arreglo [12] de enteros), parámetro por referencia

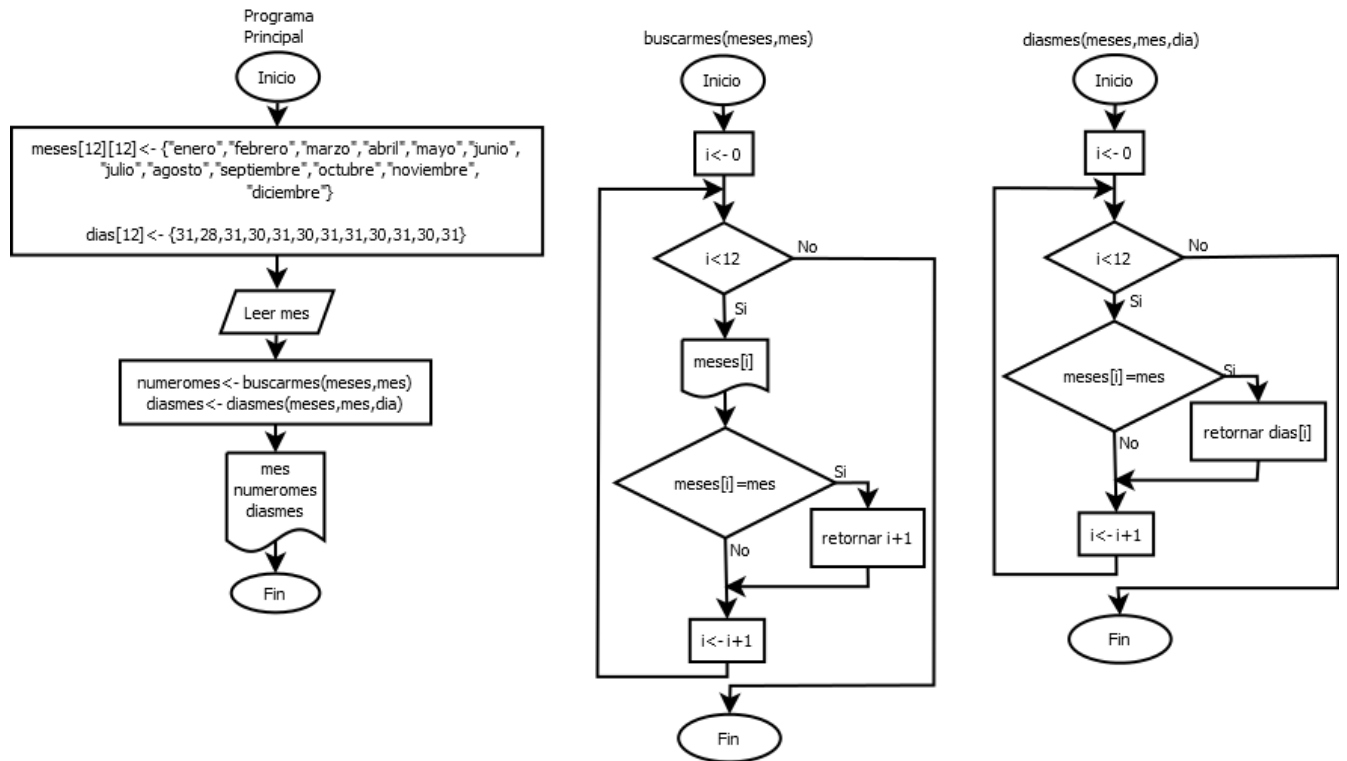
mes (string) , parámetro por referencia

PRECONDICIÓN: Meses almacena el conjunto {"enero",...,"diciembre"}, días almacena el número de días por cada mes tal que para todo i $28 \leq \text{días}[i] \leq 31$ y mes debe ser válido

SALIDA: retorna un entero

POSTCONDICIÓN: diasmes devuelve el número de días correspondientes a mes

DIAGRAMA DE FLUJO:



Si se desea hacer que el programa considere años bisiestos se puede hacer una función que determine si un año ingresado es bisiesto y hacer dicha comprobación en el cálculo de días para que en el caso de ser bisiesto devuelva un día más

4. Usando diagramas de flujo diseñe un programa que lea dos matrices y calcule la suma de ellas, luego imprime el resultado en notación matricial. Para ello, diseñe un subprograma que lea una matriz, otro subprograma que calcule la suma de dos matrices dadas y devuelva una matriz con el resultado, y un subprograma que imprima en pantalla una matriz dada en notación matricial.

Programa Principal:

```
ENTRADAS:
    M,N (enteros) // Dimensiones de las matrices
    A, B (arreglos [MAX][MAX] de reales)
PRECONDICIÓN:
    0<M<MAX y 0<N<MAX
SALIDAS:
    C (arreglo [MAX][MAX] de reales) // la matriz suma
POSTCONDICIÓN:
    Para todo i,j se cumple  $C[i][j]=A[i][j]+B[i][j]$ 
```

Funcion LeerMatriz(M,N)

```
ENTRADAS:
    M,N (enteros), parámetro por valor
PRECONDICIÓN:
    0<M<MAX y 0<N<MAX
SALIDA:
    retorna un arreglo de arreglo de reales
POSTCONDICIÓN:
    LeerMatriz devuelve una matriz leida de tamaño MxN
```

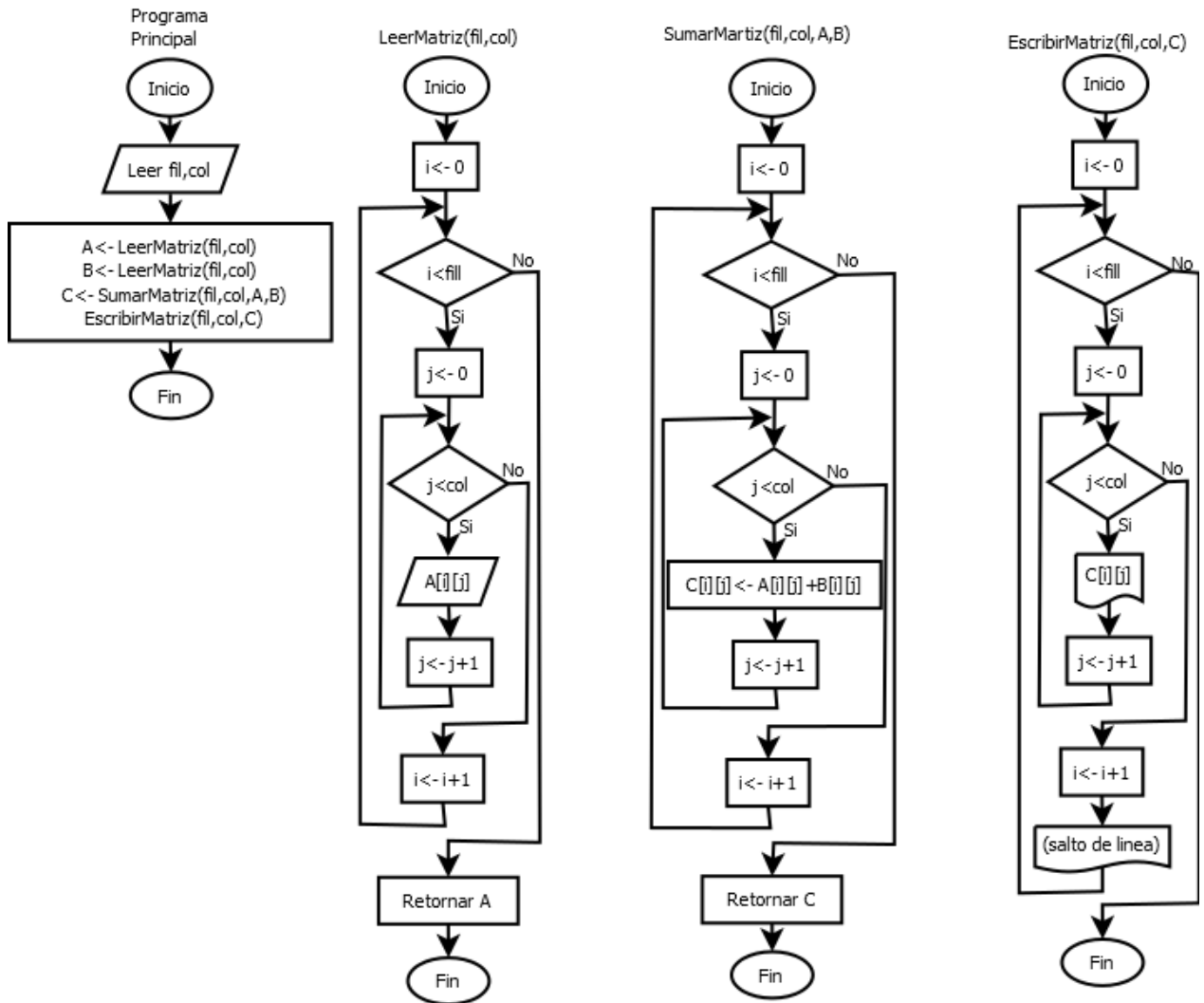
Funcion SumarMatriz(M,N,A,B)

```
ENTRADAS:
    M,N (enteros), parámetro por valor
    A,B (arreglo [MAX][MAX] de reales), parámetro por referencia
PRECONDICIÓN:
    0<M<MAX y 0<N<MAX
SALIDA:
    retorna un arreglo de arreglo de reales
POSTCONDICIÓN:
    SumarMatriz devuelve la matriz suma de A+B
```

Procedimiento EscribirMatriz(M,N,A,B)

```
ENTRADAS:
    M,N (enteros), parámetro por valor
    A,B (arreglo [MAX][MAX] de reales), parámetro por referencia
PRECONDICIÓN:
    0<M<MAX y 0<N<MAX
POSTCONDICIÓN:
    SumarMatriz imprime en pantalla la matriz
```

DIAGRAMA DE FLUJO:



5. Se desea escribir un programa que permita visualizar el Triangulo de Pascal de orden n. Ejemplo para n=4:

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1

```

Para ello, implemente las siguientes funciones:

a) Escriba una función que dado n calcule el Triángulo de Pascal en una matriz (n+1)x(n+1). Asuma que la matriz tiene un tamaño máximo MAX y que n<MAX. Ejemplo para n=4, la matriz quedaría

```

 1  0  0  0  0
 1  1  0  0  0
 1  2  1  0  0
 1  3  3  1  0
 1  4  6  4  1

```

Procedimiento TriPascal

ENTRADAS:

n (entero), parámetro por valor

PRECONDICION:

$0 < n < \text{MAX}+1$

SALIDA:

tp (arreglo $[\text{MAX}+1][\text{MAX}+1]$ de enteros), parámetro por referencia

POSTCONDICION: La matriz tp contiene el Triangulo de Pascal de orden n

- b) Escriba una función que dado un entero n y una matriz con el Triángulo de Pascal de orden n, construya una matriz $(n+1) \times (2n+1)$ con la pirámide correspondiente a dicho triángulo.

Ejemplo para n=4:

```
0 0 0 0 1 0 0 0 0
0 0 0 1 0 1 0 0 0
0 0 1 0 2 0 1 0 0
0 1 0 3 0 3 0 1 0
1 0 4 0 6 0 4 0 1
```

Procedimiento Piramide

ENTRADAS:

tp (arreglo $[\text{MAX}+1][\text{MAX}+1]$ de enteros)

n (entero)

PRECONDICION:

$0 < n < \text{MAX}+1$ y la matriz tp contiene el Triangulo de Pascal de orden n

SALIDA:

pir (arreglo $[\text{MAX}+1][2*\text{MAX}+1]$ de enteros)

POSTCONDICION:

la matriz pir contiene la pirámide correspondiente al Triángulo dado

- c) Escriba una función que dado un entero n y la matriz $(n+1) \times (2n+1)$ escriba el triángulo en pantalla

Procedimiento EscribirPiramide

ENTRADAS:

pir (arreglo $[\text{MAX}+1][2*\text{MAX}+1]$ de enteros)

n (entero)

PRECONDICION:

$0 < n < \text{MAX}+1$ y la matriz pir contiene una pirámide de orden n

POSTCONDICION: se imprime la matriz en pantalla

- d) Escriba el programa principal que lee n, lo valida e invoca a las funciones anteriores para imprimir el Triángulo de Pascal de orden n

Programa Principal:

ENTRADAS:

n (entero)

PRECONDICIÓN:

$0 < n < \text{MAX}+1$

POSTCONDICIÓN:

Se imprime la pirámide asociada al triángulo de Pascal

DIAGRAMA DE FLUJO:

