

**UNIVERSIDAD ABIERTA INTERAMERICANA FACULTAD DE TECNOLOGÍA INFORMÁTICA**

Programación Estructurada

Trabajo práctico obligatorio individual Nro 2 – Año 2020

Alumno: Julian Ryan

Docente: Gastón Matías Weingand y Agustina Gantuz Fecha entrega: 10/07/2020

Objetivo del TP:

Desarrollar programas que incorporen los conceptos de vectores, matrices y funciones dentro del paradigma estructurado.

Resolver los siguientes ejercicios en código C:

1. La empresa [www.ejercicios.com.ar](http://www.ejercicios.com.ar/), de venta de productos por Internet, comercializa 10 productos los cuales se identifican por nros. correlativos a partir del 1. Se debe realizar la carga de los precios de dichos artículos y una vez efectuada averiguar e imprimir:
   1. Precio máximo y nro. de artículo al que corresponde.
   2. Precio mínimo y nro. de artículo al que corresponde.
   3. Cantidad de artículos con precio superior al precio promedio.
   4. Cantidad de artículos con precio superior a $100.
   5. Listar los artículos por precio en orden ascendente.
2. Escribir un programa que permita ingresar una cadena de hasta diez caracteres que representa un número en numeración romana, y por medio de una función **romanoadecimal ()** imprima el número y su equivalente en numeración arábiga. Anexar el pseudocódigo y el diagrama de flujo correspondiente solo para este ejercicio.
3. Realizar un programa que permita realizar y mostrar el resultado de las operaciones de **suma y multiplicación** de 2 matrices 3 x 3 y la función transpuesta de una **matriz 4 x 4** (Los datos de todas las matrices las ingresa el usuario).

**Aclaraciones:**

-El TP se entrega en un archivo comprimido (zip o rar) con el código C de las soluciones realizadas. El pseudocódigo y el diagrama del punto 2 pueden adjuntarse directamente desde el Pseint o en un Word dentro del mismo archivo.

Desarrollo:

1. **Código en C:**

/\*

1. La empresa www.ejercicios.com.ar, de venta de productos por

Internet, comercializa 10 productos los cuales se identifican por nros.

correlativos a partir del 1. Se debe realizar la carga de los precios de

dichos artículos y una vez efectuada averiguar e imprimir:

a. Precio máximo y nro. de artículo al que corresponde.

b. Precio mínimo y nro. de artículo al que corresponde.

c. Cantidad de artículos con precio superior al precio promedio.

d. Cantidad de artículos con precio superior a $100.

e. Listar los artículos por precio en orden ascendente

\*/

#include <stdio.h>

int main ()

{

//Declaración de variables

int i =0;

int j =0;

int numeroAuxiliar1 =0;

int numeroAuxiliar2 = 0;

int posicionProductoMayorPrecio =0;

int posicionProductoMenorPrecio =0;

int productoMayorPrecio = 0;

int productoMenorPrecio = 0;

int productosSuperioresACien = 0;

int productosSuperioresAPromedio = 0;

int sumaDelPromedio = 0;

int vectorNumeroDeProducto[10];

int vectorPrecio[10];

float promedio = 0;

//Solicitud de datos al usuario

for(i=0; i<10; i++)

{

printf("Por favor, ingrese el precio del articulo %i \n", i+1);

scanf("%i", &vectorPrecio[i]);

}

printf("\nLos precios de los articulos ingresados son: \n");

//Impresión de los datos ingresados

for(i=0; i<10; i++)

{

printf("El precio del articulo %i es $ %i\n", i+1, vectorPrecio[i]);

}

//a. Precio máximo y nro. de artículo al que corresponde.

//Obtención del precio máximo y su ubicación

for (i=0; i<10; i++)

{

if(vectorPrecio[i]>productoMayorPrecio){

productoMayorPrecio = vectorPrecio[i];

posicionProductoMayorPrecio = i+1;

}

}

//Impresión del precio máximo

printf("\n\nEl precio mayor es: $ %i y el numero de articulo es %i \n\n", productoMayorPrecio,posicionProductoMayorPrecio);

//b. Precio mínimo y nro. de artículo al que corresponde.

//Obtención del precio minimo e intento de su ubicación

productoMenorPrecio = productoMayorPrecio;

for (i=0; i<10; i++)

{

if(vectorPrecio[i]<productoMenorPrecio){

productoMenorPrecio = vectorPrecio[i];

posicionProductoMenorPrecio = i+1;

}

}

//Impresión del precio minimo

printf("El precio menor es: $ %i y el numero de articulo es %i \n\n", productoMenorPrecio,posicionProductoMenorPrecio);

//c. Cantidad de artículos con precio superior al precio promedio.

//Obteción del promedio

for(i=0; i<10; i++)

{

sumaDelPromedio += vectorPrecio[i];

}

promedio = (sumaDelPromedio)/10;

//Impresión del promedio

printf("El promedio de los precios es de: $ %.2f \n\n" , promedio);

//Contador para los productos que superan al promedio

for(i=0; i<10; i++)

{

if(vectorPrecio[i]>promedio)

productosSuperioresAPromedio++;

}

//Impresión de la cantidad de productos que superan al promedio

printf ("La cantidad de articulos con precio superior al promedio es de: %i \n\n", productosSuperioresAPromedio);

//d. Cantidad de artículos con precio superior a $100

//Contador para los productos con precio superior a $100

for(i=0; i<10; i++)

{

if(vectorPrecio[i]>100)

productosSuperioresACien++;

}

//Impresión de la cantidad de productos que superan a $100

printf ("La cantidad de articulos con precio superior a $100 es de: %i \n\n", productosSuperioresACien);

//e. Listar los artículos por precio en orden ascendente

//Generación de un vector con las posiciones del producto

for (i=0; i<10; i++)

{

vectorNumeroDeProducto[i]=i;

}

//Ordenamiento ascendente por sustitución con un auxiliar de los valores del vector

for (i=0; i<10; i++)

{

for (j=i+1; j<10; j++)

{

if(vectorPrecio[j]<vectorPrecio[i])

{

numeroAuxiliar1=vectorPrecio[i];

vectorPrecio[i]=vectorPrecio[j];

vectorPrecio[j]=numeroAuxiliar1;

numeroAuxiliar2=vectorNumeroDeProducto[i];

vectorNumeroDeProducto[i]=vectorNumeroDeProducto[j];

vectorNumeroDeProducto[j]=numeroAuxiliar2;

}

}

}

//Impresión de los resultados

printf("Los articulos ordenados de manera ascendente segun su precio son: \n\n");

for (i=0; i<10; i++)

{

printf("El precio del articulo %i es $ %i \n", vectorNumeroDeProducto[i]+1, vectorPrecio[i]);

}

return 0;

}

1. **Código en C:**

/\*

2) Escribir un programa que permita ingresar una cadena de hasta diez

caracteres que representa un número en numeración romana, y por

medio de una función romanoadecimal () imprima el número y su

equivalente en numeración arábiga. Anexar el pseudocódigo y el

diagrama de flujo correspondiente solo para este ejercicio.

\*/

#include <stdio.h>

#include <string.h>

//Declaracion de función

void romanoADecimal (int n,char romano[], int valor []);

int main () {

//Declaración de variables

char romano [10];

int valor [100];

int n=0;

//Solicitud de datos al usuario

printf("Ingrese el numero romano que desea transformar a decimal: \n");

gets (romano);

n=strlen (romano);

//función de romanos a decimal

romanoADecimal (n,romano,valor);

return 0;

}

void romanoADecimal (int n,char romano[], int valor []){

//Declaración de variables internas a la función

int i;

int numero;

//Transformación de caracteres romanos a numeros decimales

for(i=0;i<n;i++){

if (romano [i] == 'I' || romano [i] =='i')

valor[i]=1;

if (romano [i] == 'V' || romano [i] =='v')

valor[i]=5;

if (romano [i] == 'X' || romano [i] =='x')

valor[i]=10;

if (romano [i] == 'L' || romano [i] =='l')

valor[i]=50;

if (romano [i] == 'C' || romano [i] =='c')

valor[i]=100;

if (romano [i] == 'D' || romano [i] =='d')

valor[i]=500;

if (romano [i] == 'M' || romano [i] =='m')

valor[i]=1000;

}

//Transformación de los valores negativos

for(i=0;i<n;i++){

if(i==n-1){

numero+=valor[i];

}

else

if(valor[i]>=valor [i+1]){

numero+=valor[i];

}

else{

numero-=valor[i];

}

}

//Impresión del resultado

printf ("El numero es: %i", numero);

}

-**Pseudocódigo**:

Funcion ROMANOADECIMAL ()

// Declaración de variables internas a la función

Definir i Como Entero

Definir num Como Entero

Dimension nRomano[10]

Dimension valorNumerico[100]

// Transformación de caracteres romanos a numeros decimales

Para i<-0 Hasta n Hacer

Si (nRomano[i]==i O nRomano[i]==i) Entonces

valorNumerico[i] <- 1

FinSi

Si (nRomano[i]==v O nRomano[i]==v) Entonces

valorNumerico[i] <- 5

FinSi

Si (nRomano[i]==x O nRomano[i]==x) Entonces

valorNumerico[i] <- 10

FinSi

Si (nRomano[i]==l O nRomano[i]==l) Entonces

valorNumerico[i] <- 50

FinSi

Si (nRomano[i]==c O nRomano[i]==c) Entonces

valorNumerico[i] <- 100

FinSi

Si (nRomano[i]==d O nRomano[i]==d) Entonces

valorNumerico[i] <- 500

FinSi

Si (nRomano[i]==m O nRomano[i]==m) Entonces

valorNumerico[i] <- 1000

FinSi

FinPara

// Transformación de los valores negativos

Para i<-0 Hasta n Hacer

Si (i==n-1) Entonces

num <- num+valorNumerico[i]

SiNo

Si (valorNumerico[i]>=valorNumerico[i+1]) Entonces

num <- num+valorNumerico[i]

SiNo

num <- num-valorNumerico[i]

FinSi

FinSi

FinPara

// Impresión del resultado

Escribir 'El numero es: ',num

FinFuncion

Algoritmo Ejercicio2

// Declaración de variables

Definir nRomano Como Caracter

Definir n Como Entero

Dimension nRomano[10]

// Solicitud de datos al usuario

Escribir 'Ingrese la cantidad de caracteres que tiene el numero romano que desea transformar a decimal: '

Leer n

Escribir 'Ingrese el numero romano que desea transformar a decimal: '

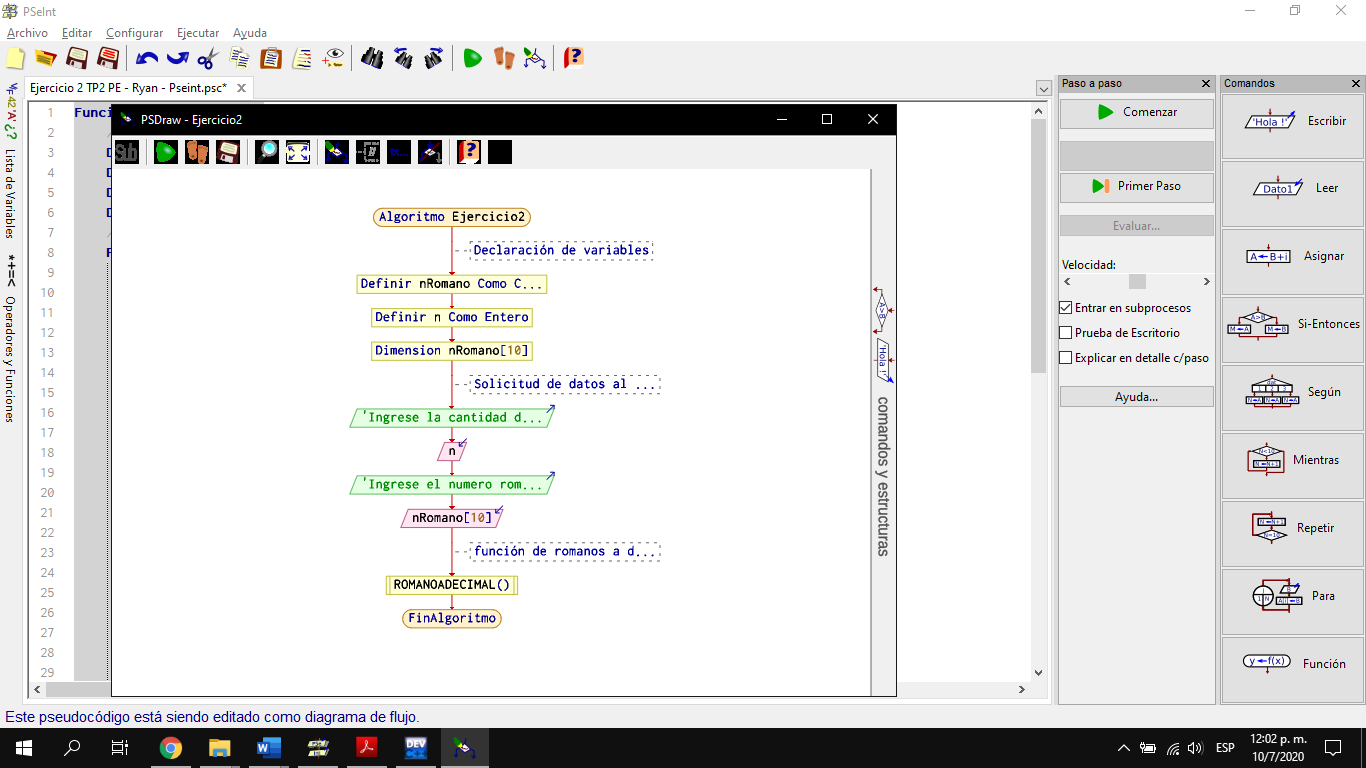
Leer nRomano[10]

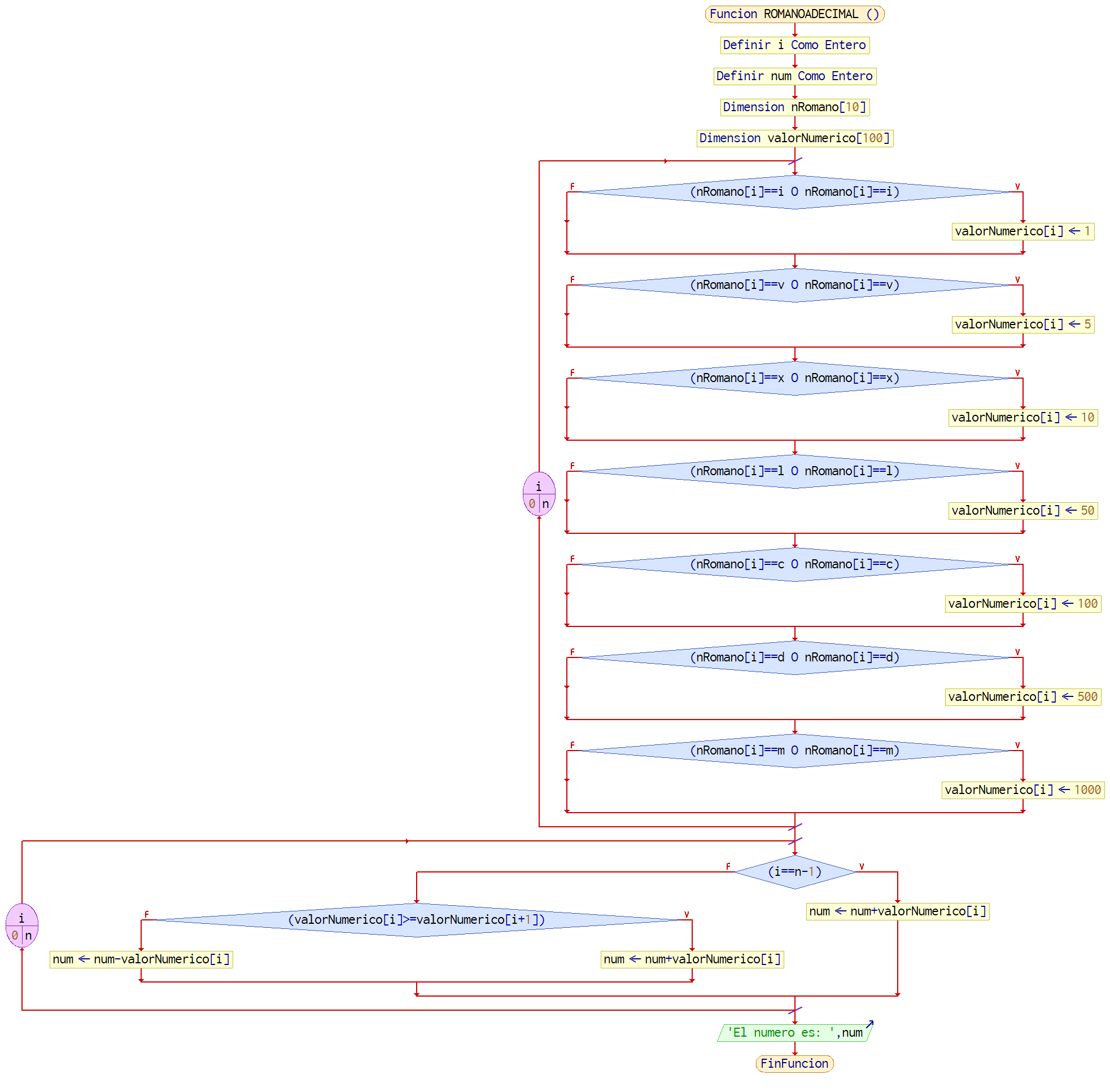
// función de romanos a decimal

ROMANOADECIMAL()

FinAlgoritmo

**-Diagrama de flujo:**





1. **Código en C:**

/\*

Realizar un programa que permita realizar y mostrar el resultado de

las operaciones de suma y multiplicación de 2 matrices 3 x 3 y la

función transpuesta de una matriz 4 x 4 (Los datos de todas las

matrices las ingresa el usuario).

\*/

#include <stdio.h>

#define fil 3

#define col 3

#define fila 4

#define colu 4

/\*Funciones que no logro hacer que anden

void valoresMatrices (int matriz1 [] [col], int matriz2 [] [col]);

void sumaMatrices (int matriz1 [] [col], int matriz2 [] [col], int matriz3 [] [col]);

void multiplicacionMatrices (int matriz1 [] [col], int matriz2 [] [col], int matriz4 [] [col]);

void valorMatrizT ()

\*/

int main () {

//Declaración de variables

int i=0;

int j=0;

int k=0;

int l=0;

int opcion = 0;

int matriz1 [fil] [col];

int matriz2 [fil] [col];

int matriz3 [fil] [col];

int matriz4 [fil] [col];

int matriz5 [fila] [colu];

int matriz6 [fila] [colu];

//Menu Inicial - Solicitud de datos al usuario

printf("Seleccione una de las siguientes opciones para operar con matrices: \n");

printf("1: Suma de dos matrices 3x3 \n");

printf("2: Multiplicacion de dos matrices 3x3 \n");

printf("3: Funcion transpuesta de una matriz 4x4 \n");

printf("0: Salir \n");

scanf("%i", &opcion);

//Switch segun opciones para operar con matricies

switch (opcion){

case 1: printf("Ingrese los datos para la suma de dos matrices 3x3: \n");

// Introducción de los valores de la primera matriz

for(i=0; i<fil; i++){ // i corresponde a filas

printf("Coloque los datos de la fila numero %i: \n", i+1);

for(j=0; j<col; j++){ // j corresponde a columnas

scanf("%i", &matriz1[i][j]);

}

}

//Impresion de los valores almacenados en la primera matriz

printf("Primera matriz:\n");

for(k=0; k<fil; k++ ){

for( l=0; l<col; l++ ){

printf( " %i ", matriz1[k][l]);

}

printf( "\n");

}

// Introducción de los valores de la segunda matriz

for(i=0; i<fil; i++){ // i corresponde a filas

printf("Coloque los datos de la fila numero %i: \n", i+1);

for(j=0; j<col; j++){ // j corresponde a columnas

scanf("%i", &matriz2[i][j]);

}

}

//Impresion de los valores almacenados en la segunda matriz

printf("Segunda matriz:\n");

for(k=0; k<fil; k++ ){

for( l=0; l<col; l++ ){

printf( " %i ", matriz2[k][l] );

};

printf( "\n");

};

// Se calcula el resultado:

for( i=0; i<fil; i++ ){

for( j=0; j<col; j++ ){

matriz3[i][j] = matriz1[i][j] + matriz2[i][j];

};

};

//Se imprime la matriz resultante de la suma

printf( "Matriz resultante de la suma: \n");

for( i=0; i<fil; i++ ){

for( j=0; j<col; j++ ){

printf( " %i ", matriz3 [i][j] );

};

printf( "\n");

};

break;

case 2: printf("Ingrese los datos para la multiplicacion de dos matrices 3x3 \n");

// Introducción de los valores de la primera matriz

for(i=0; i<fil; i++){ // i corresponde a filas

printf("Coloque los datos de la fila numero %i: \n", i+1);

for(j=0; j<col; j++){ // j corresponde a columnas

scanf("%i", &matriz1[i][j]);

}

}

//Impresion de los valores almacenados en la primera matriz

printf("Primera matriz:\n");

for(k=0; k<fil; k++ ){

for( l=0; l<col; l++ ){

printf( " %i ", matriz1[k][l]);

}

printf( "\n");

}

// Introducción de los valores de la segunda matriz

for(i=0; i<fil; i++){ // i corresponde a filas

printf("Coloque los datos de la fila numero %i: \n", i+1);

for(j=0; j<col; j++){ // j corresponde a columnas

scanf("%i", &matriz2[i][j]);

}

}

//Impresion de los valores almacenados en la segunda matriz

printf("Segunda matriz:\n");

for(k=0; k<fil; k++ ){

for( l=0; l<col; l++ ){

printf( " %i ", matriz2[k][l] );

};

printf( "\n");

};

// Se calcula el resultado:

for( i=0; i<fil; i++ ){

for( j=0; j<col; j++ ){

matriz3[i][j] = matriz1[i][j] \* matriz2[i][j];

};

};

//Se imprime la matriz resultante de la multiplicacion

printf( "Matriz resultante de la multiplicacion: \n");

for( i=0; i<fil; i++ ){

for( j=0; j<col; j++ ){

printf( " %i ", matriz3 [i][j] );

};

printf( "\n");

};

break;

case 3: printf("Ingrese los datos para la funcion transpuesta de una matriz 4x4 \n");

int i,j,k,l;

int matriz5 [fila] [colu];

int matriz6 [fila] [colu];

// Introducción de los valores de la matriz

for (i=0; i<fila; i++){

printf("Coloque los datos de la fila numero %i: \n", i+1);

for(j=0; j<colu; j++){

scanf("%d", &matriz5 [i] [j]); //Lectura de los valores iniciales

}

printf("\n");

}

//Impresion de los resultados

printf( "Matriz original: \n");

for (i=0; i<fila; i++){

for(j=0; j<colu; j++){

printf("%d", matriz5 [i] [j]);

printf(" ");

}

}

//Proceso de transposicion de la matriz

for (i=0; i<fila; i++){

for(j=0; j<colu; j++){

matriz6 [j] [i] = matriz5 [i] [j]; //Proceso de intercambio de los indices de lugar

}

}

//Impresion de los resultados

printf( "Matriz resultante de la transposicion: \n");

for (i=0; i<fila; i++){

for(j=0; j<colu; j++){

printf("%d", matriz6 [i] [j]);

printf(" ");

}

printf("\n");

}

break;

case 0: printf ("Usted a seleccionado salir del programa");

return 0;

break;

default: printf("Ha ingresado una opcion invalida\n");

}

return 0;

}

/\*

void valoresMatrices (int matriz1 [] [col], int matriz2 [] [col]) {

int i,j,k,l;

// Introducción de los valores de la primera matriz

for(i=0; i<f; i++){ // i corresponde a filas

for(j=0; j<c; j++){ // j corresponde a columnas

scanf("%i", &matriz1[i][j]);

}

}

//Impresion de los valores almacenados en la primera matriz

printf("Primera matriz:\n");

for(k=0; k<f; k++ ){

for( l=0; l<c; l++ ){

printf( " %i ", matriz1[k][l]);

}

printf( "\n");

}

// Introducción de los valores de la segunda matriz

for(i=0; i<f; i++){ // i corresponde a filas

for(j=0; j<c; j++){ // j corresponde a columnas

scanf("%i", &matriz2[i][j]);

}

}

//Impresion de los valores almacenados en la segunda matriz

printf("Segunda matriz:\n");

for(k=0; k<f; k++ ){

for( l=0; l<c; l++ ){

printf( " %i ", matriz2[k][l] );

};

printf( "\n");

}

void sumaMatrices (int matriz1 [] [col], int matriz2 [] [col], int matriz3 [] [col]){

int i,j,k,l;

// Se calcula el resultado:

for( i=0; i<f; i++ ){

for( j=0; j<c; j++ ){

matriz3[i][j] = matriz1[i][j] + matriz2[i][j];

};

};

//Se imprime la matriz resultante de la suma

printf( "Matriz resultante: \n");

for( i=0; i<f; i++ ){

for( j=0; j<c; j++ ){

printf( " %i ", matriz3 [i][j] );

};

printf( "\n");

}

}

}

void multiplicacionMatrices (int matriz1 [fil] [col], int matriz2 [fil] [col], int matriz4 [fil] [col]){

int i,j,k,l;

// Se calcula el resultado:

for( i=0; i<fil; i++ ){

for( j=0; j<col; j++ ){

matriz4[i][j] = matriz1[i][j] \* matriz2[i][j];

};

};

//Se imprime la matriz resultante de la multiplicacion

printf( "Matriz resultante: \n");

for( i=0; i<fil; i++ ){

for( j=0; j<col; j++ ){

printf( " %i ", matriz4 [i][j] );

};

printf( "\n");

};

}

void valorMatrizT () {

int i,j,k,l;

int matriz5 [fila] [colu]

int matriz6 [fila] [colu];

// Introducción de los valores de la matriz

for (i=0; i<fila; i++){

for(j=0; j<colu; j++){

scanf("%d", &matriz5 [i] [j]); //Lectura de los valores iniciales

}

printf("\n");

}

//Proceso de transposicion de la matriz

for (i=0; i<fila; i++){

for(j=0; j<colu; j++){

matriz6 [j] [i] = matriz5 [i] [j]; //Proceso de intercambio de los indices de lugar

}

}

//Impresion de los resultados

for (i=0; i<fila; i++){

for(j=0; j<colu; j++){

printf("%d", matriz6 [i] [j]);

printf(" ");

}

printf("\n");

}

}

\*/