

PROGRAMACIÓN ESTRUCTURADA

# MÓDULO I REALIZACIÓN DE UN PROGRAMA OPTIMIZADO

UTILIZANDO ESTRUCTURAS DE MEMORIA Y FUNCIONES.

## EVALUACIÓN PARCIAL Nº2 PROPUESTA DE INTEGRACIÓN DEL MÓDULO II

**Profesor Titular y autor de contenidos:**

Prof. Ing. Rafael Brizuela

**Diseño Didáctico:**

**Diseño Gráfico:**

Lic. Paula Bruzzese

|  |  |
| --- | --- |
| Nombre y apellido:  DNI: | Julian Ryan  37737401 |

**PRESENTACIÓN**

**El trabajo que usted inicia constituye la** segunda evaluación parcial de la asignatura**. Como tal, su realización es obligatoria y constituye una condición para rendir el examen final.**

Tenga en cuenta que todos los contenidos abordados en el Módulo II son fundamentales para la realización de esta Propuesta de Integración Parcial.

**CONSIGNAS**

## Por favor, realice la solución de este ejercicio en DIAGRAMA DE FLUJO LINEAL, solamente.

**Puede aplicar funciones para hacer más prolijo el diagrama.**

**Si lo resuelve a mano, cuide la legibilidad y la prolijidad del diseño final.**

1. Una ferretería recibe a diario pedidos de sus 200 clientes. Esta ferretería comercializa 1000 tipos distintos de lámparas y posee 10 empleados encargados de la recepción y entrega de los pedidos.

Se sabe que tienen los siguientes lotes de datos:

## Lote 1

* + Nro de lámpara
  + Precio
  + Stock actual

## Lote 2

* + Nro de empleado
  + Comisión sobre las ventas

Cada vez que recibe un pedido tiene los siguientes datos, que finalizan con nro de pedido = 0

* + Nro de pedido
  + Nro de cliente
  + Nro de lámpara
  + Cantidad pedida
  + Empleado que atendió el pedido

Sólo se pueden vender aquellas lámparas que tienen stock suficiente para cumplimentar el pedido, sino se descarta y se pone como pendiente.

Cada vez que se efectúa un pedido correctamente se debe actualizar el stock.

## Se desea saber:

1. ¿Qué cantidad de pedidos atendió cada empleado para cada tipo de lámpara?
2. ¿Cuál es la lámpara más solicitada?
3. Valor de cada pedido satisfecho.
4. Para cada lámpara, qué empleado vendió más cantidad.
5. Stock final actualizado de cada nro. de lámpara.
6. Imprima ordenado el punto 2.
7. Codifique en DEV C++ las siguientes funciones en un mismo programa:

* Que permita cargar una matriz de 4 x 4 por pantalla.
* Que calcule el valor máximo de la matriz, sus duplicados y en qué posición se encuentran.
* Utilizando un solo ciclo FOR, ningún IF, que permita calcular la suma de su diagonal principal.

Calcular el valor promedio de la matriz y cuantos de sus valores son mayores a ese promedio.

**CRITERIOS DE EVALUACIÓN**

*La presentación de los criterios de corrección pretende servirle como una herramienta para que usted anticipe los logros obtenidos y ajuste su producción a los requerimientos de la tarea.*

Para la evaluación de este trabajo tendremos en cuenta los siguientes criterios de evaluación:

* + Correcta resolución del problema
    - Ingreso de los datos
    - Lugar donde se realizan los cálculos
    - Lugar donde se muestran los resultados
    - Elección de las estructuras
    - Elección del tipo de ciclo elegido
  + Eficiencia en la resolución
  + Entrega en tiempo y forma

*Envíe su producción a su tutor/a a través de la herramienta que le ofrece el aula virtual. Consulte la fecha de entrega en el Cronograma de la Asignatura. En el término de unos días recibirá una devolución de su docente, a la que podrá acceder consultando sus*

***CALIFICACIONES****.*

*Gracias y éxitos!*

Desarrollo

1)

Diagrama de flujo (funciones):

*El archivo se encuentra adjunto como imagen en la carpeta del parcial*

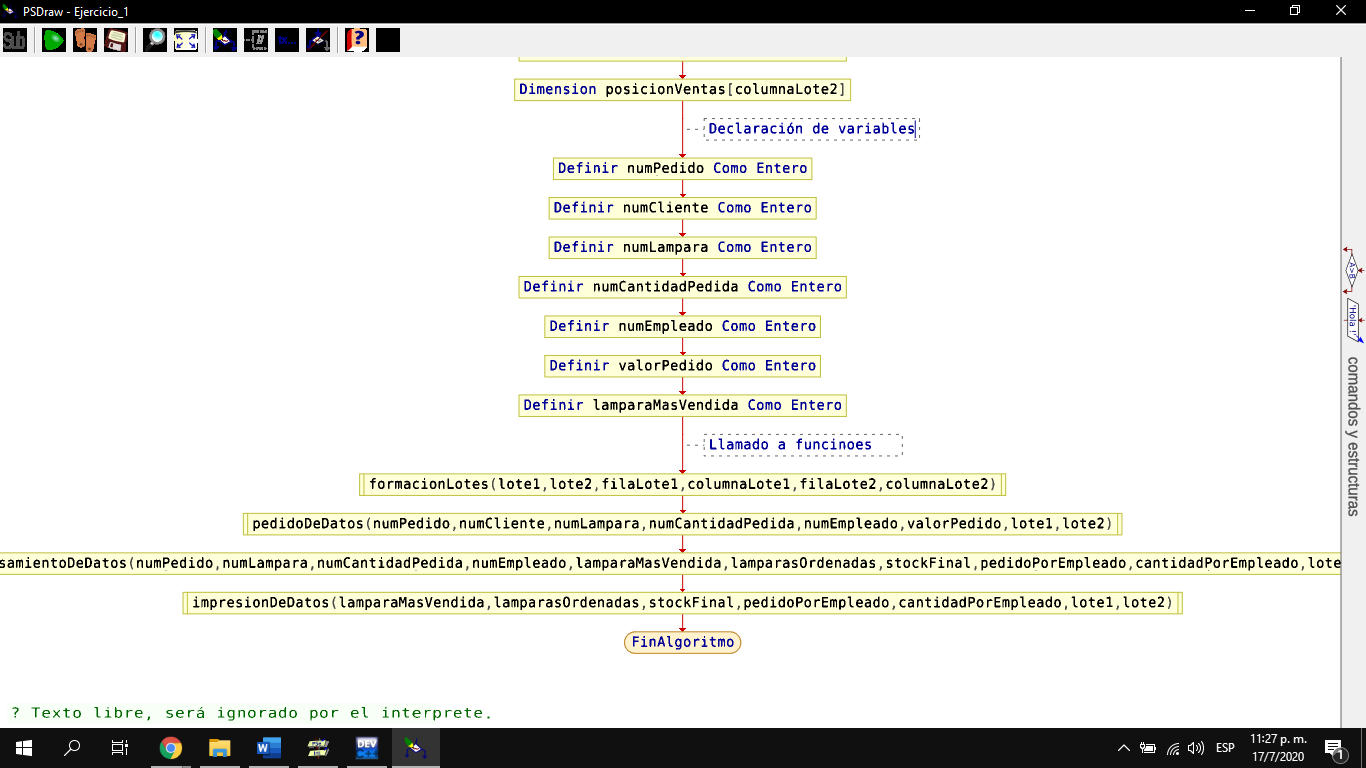
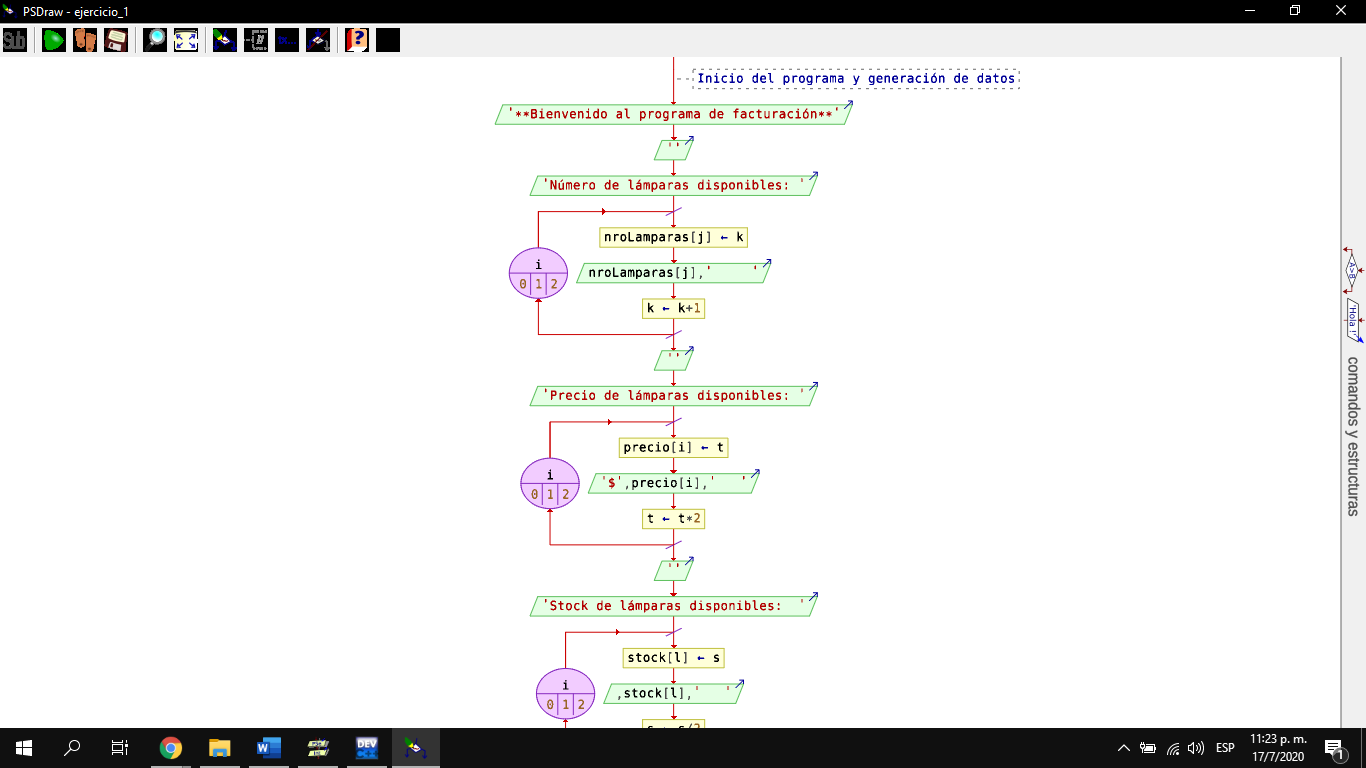
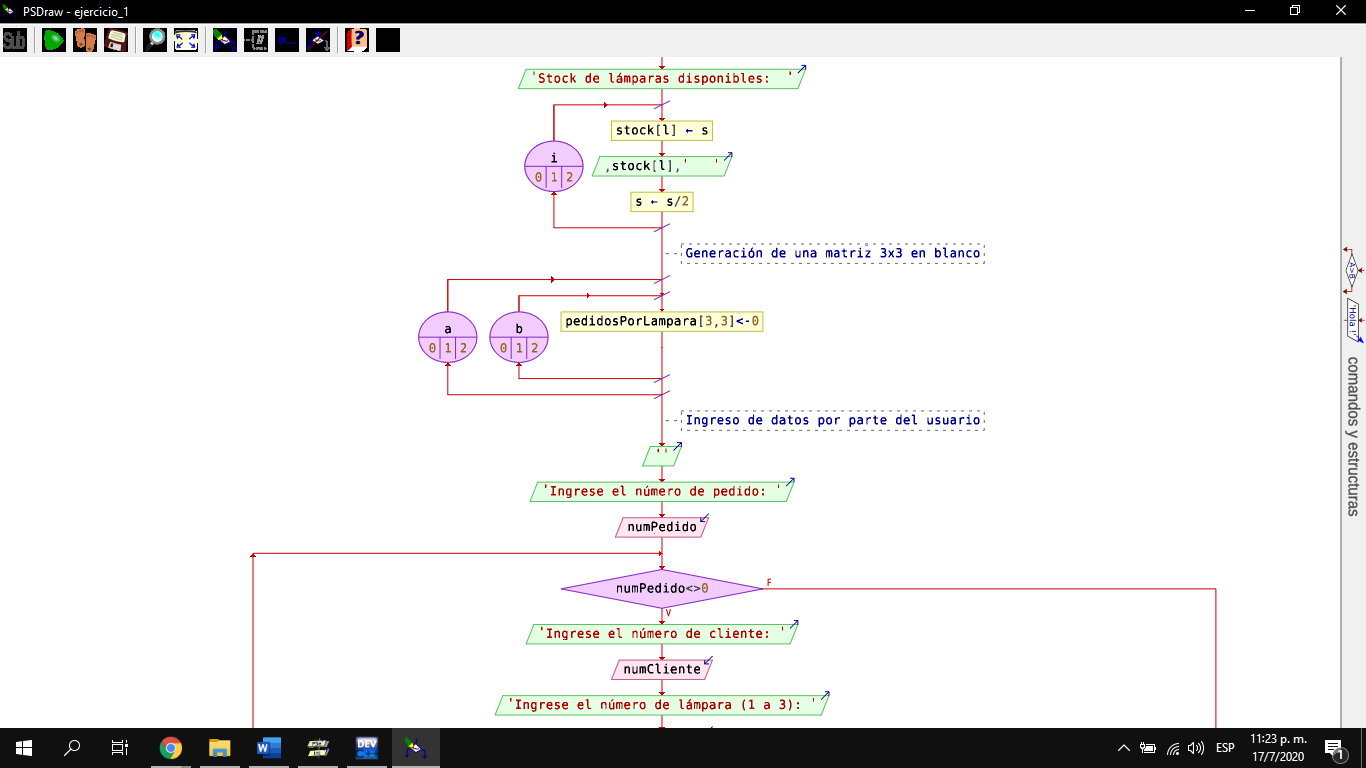
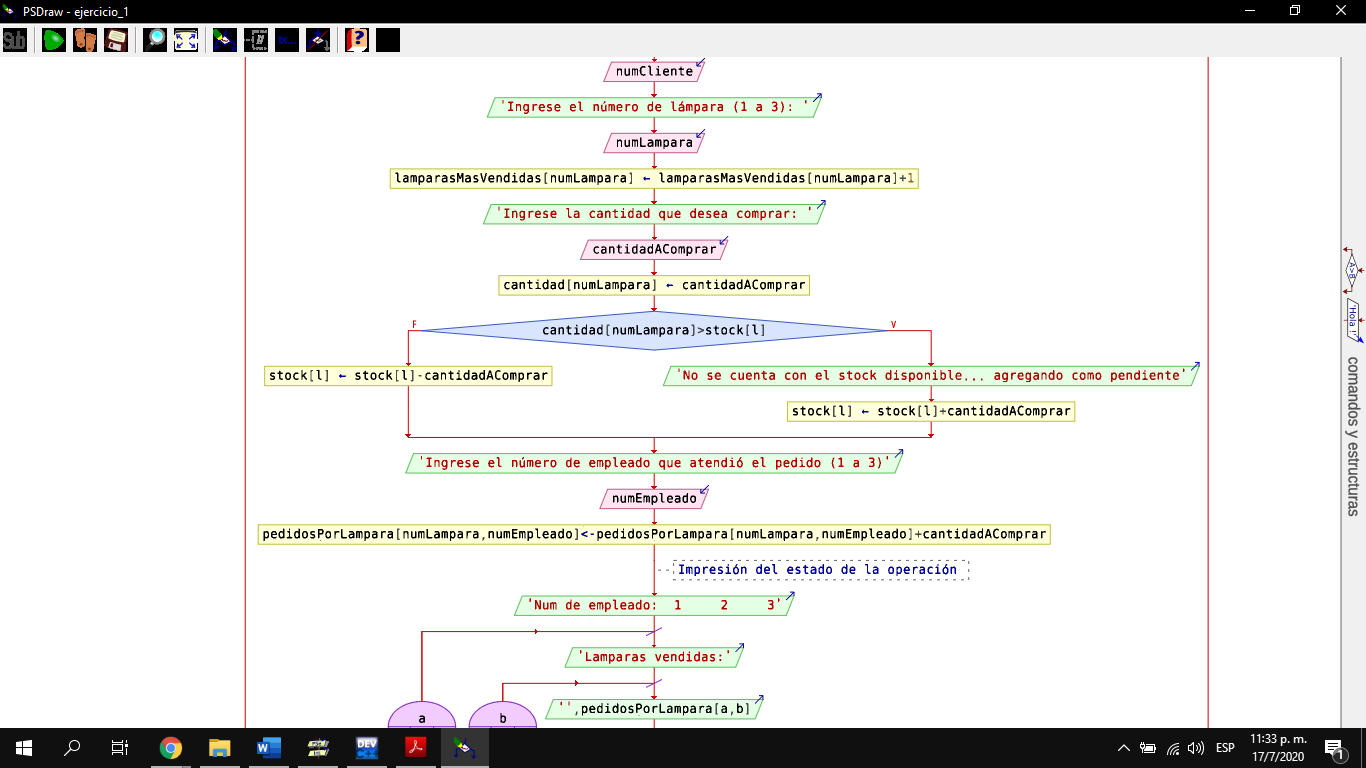


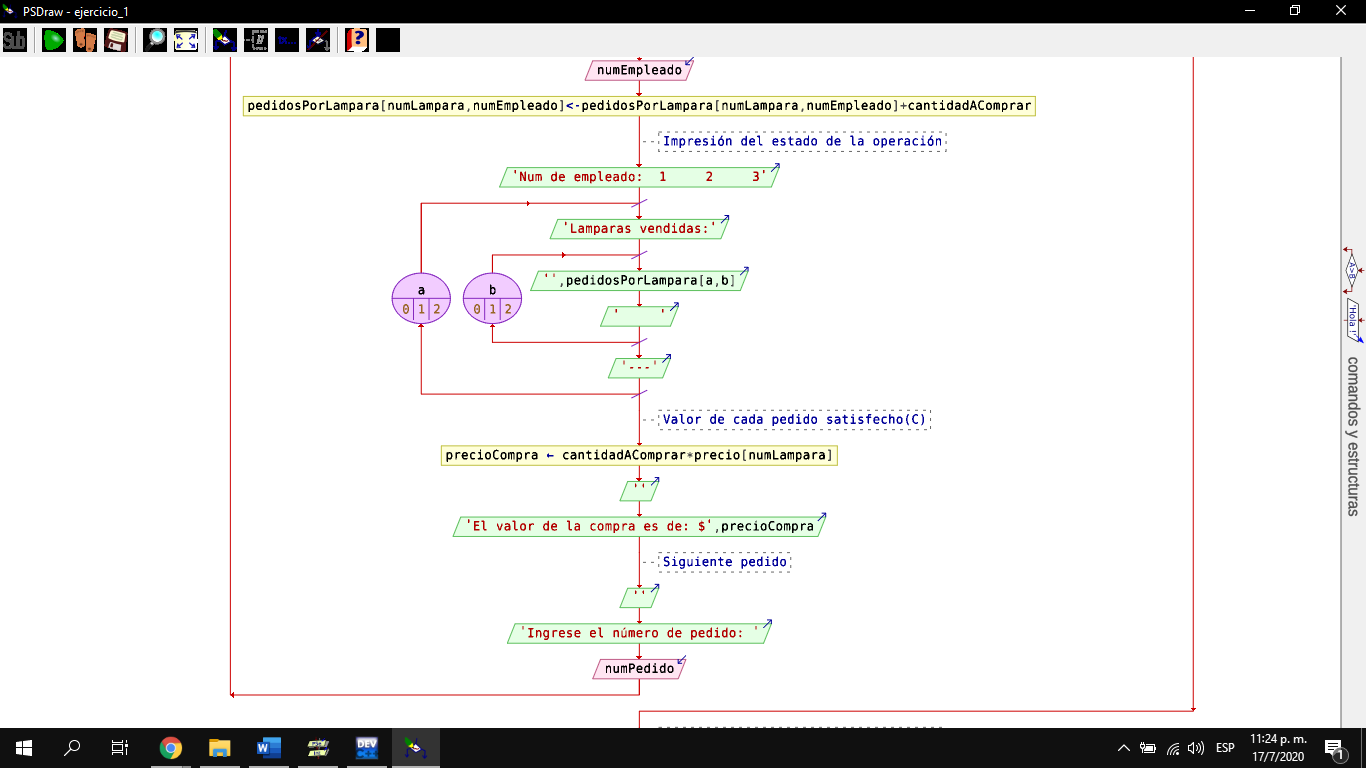
Diagrama de flujo (sin funciones):

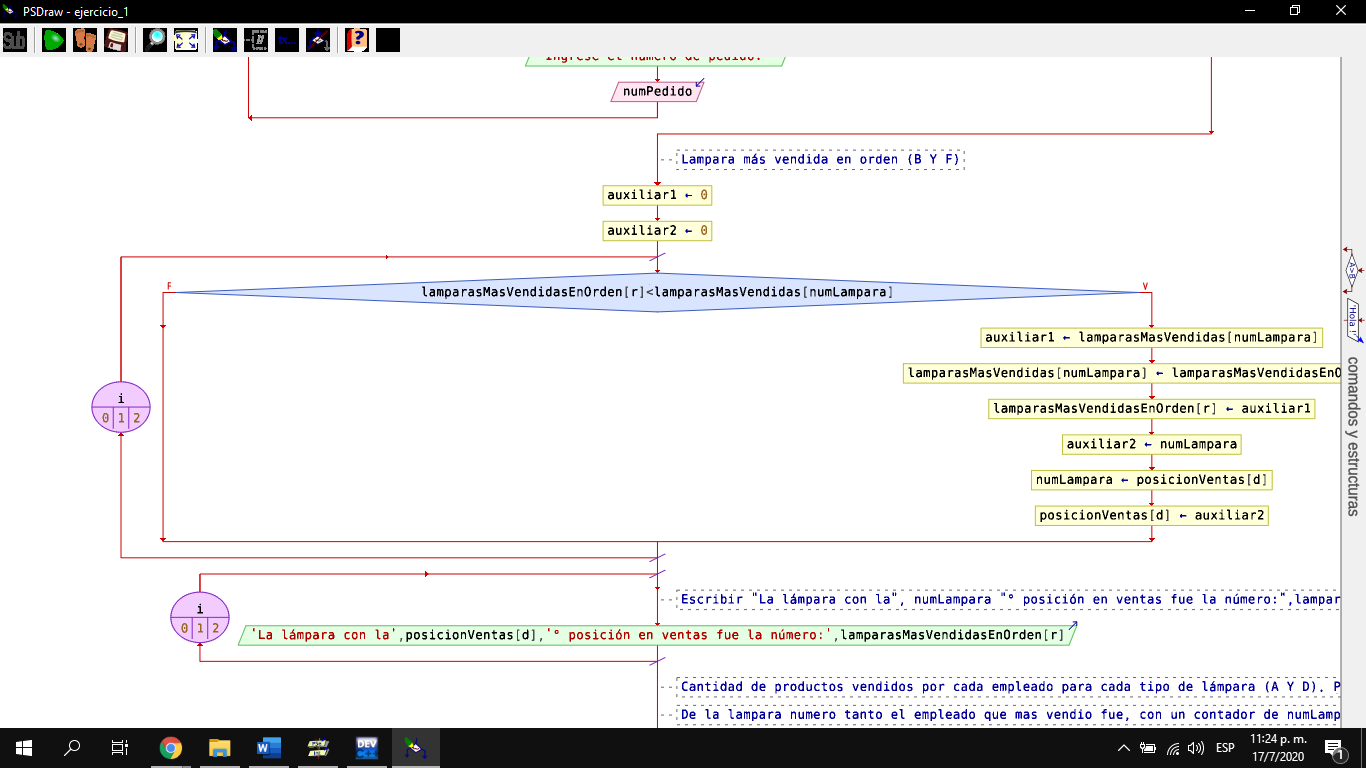
*El archivo se encuentra adjunto como imagen en la carpeta del parcial*

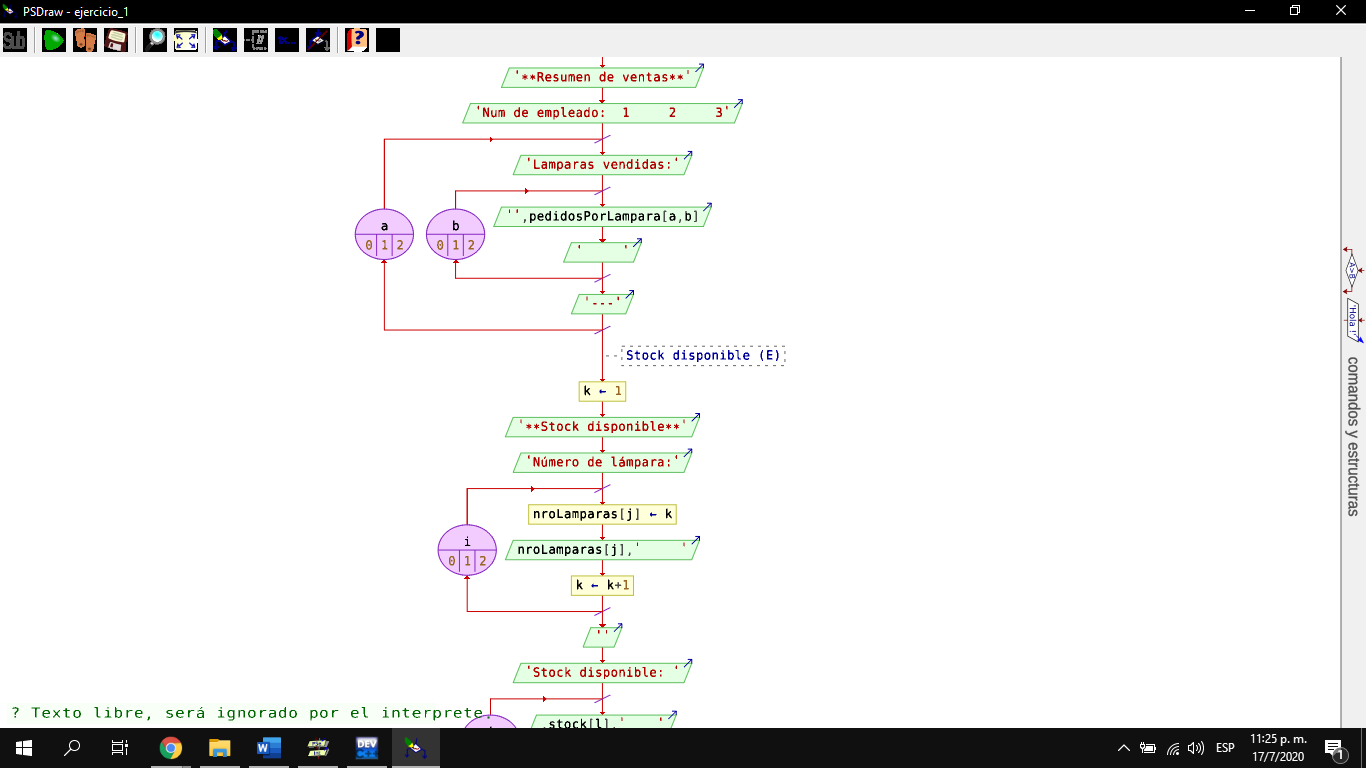


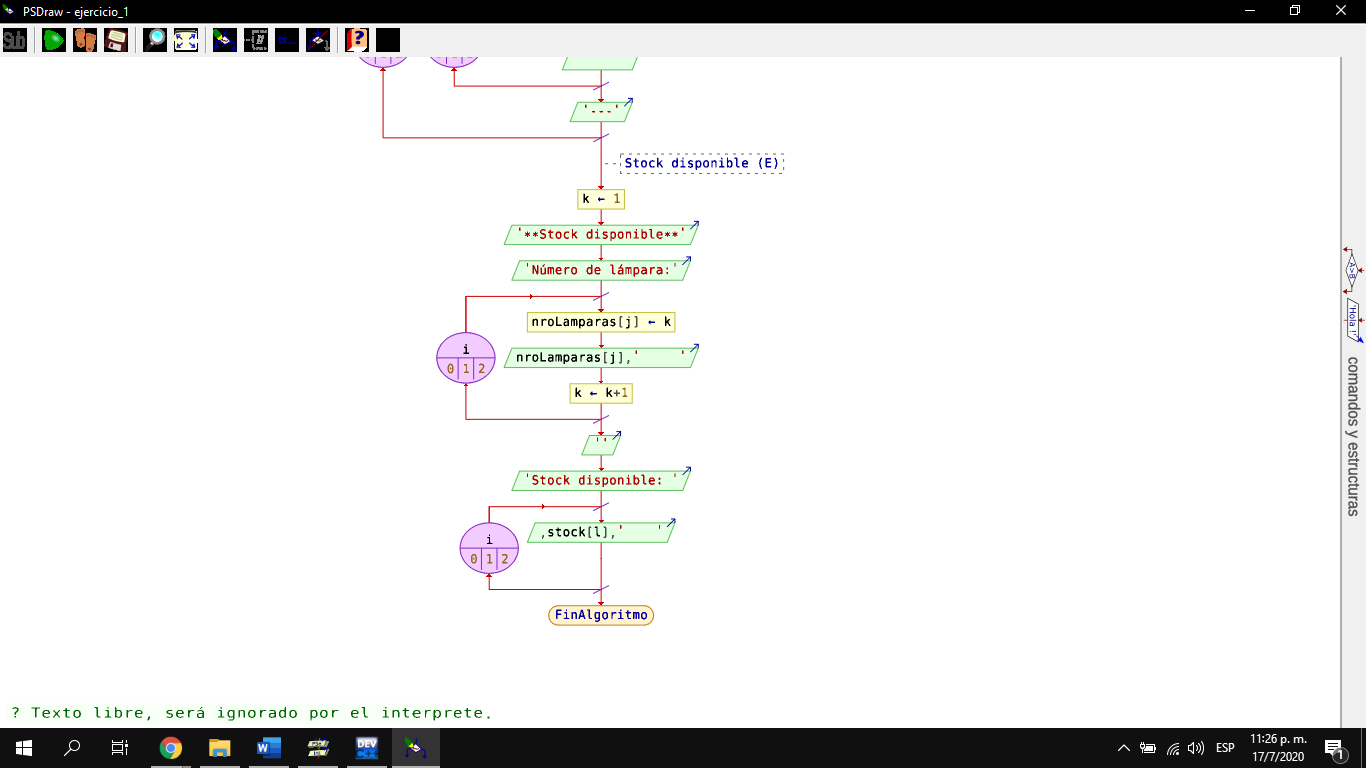












2)

Código en C:

*El archivo se encuentra adjunto en la carpeta del parcial*

#include <stdio.h>

#include <stdlib.h>

/\*Ej 2 2do Parcial

Codifique las siguientes funciones en un mismo programa:

a) Que permita cargar una matiz de 4x4 por pantalla.

b) Que calcule el valor maximo de la matriz, sus duplicados y en que posicion se encuentran.

c) Utilizando un solo ciclo FOR, ningun IF, que permita calcular la suma de sus diagonal principal.

d) Calcular el valor promedio de la matriz y cuantos de sus valores son mayores a ese promedio.\*/

int main(int argc, char \*argv[]) {

//Creacion de variables y matrices

int matriz[4][4];

int matrizValor[4][4];

int matrizFilas[4][4];

int matrizColumnas[4][4];

int i=0;

int j=0;

int k=0;

int l=0;

int m=0;

int n=0;

int imayor=0;

int jmayor=0;

int suma=0;

int total=0;

float promedio=0;

int imayorAlProm=0;

int jmayorAlProm=0;

int mayorAlPromTot=0;

int mayor=0;

int filas=4;

int columnas=4;

int diagonal=0;

int salir=1;

//a) Ingreso de datos filas y columnas de la matriz

printf ("Ingrese los numeros de la matriz:\n");

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

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

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

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

}

}

//Se imprime la matriz resultante

printf( "Matriz resultante: \n");

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

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

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

}

printf( "\n");

}

printf("\n\n\*\*Impresion de resultados\*\*\n\n");

//c) Calcular la suma de diagonal principal.

printf( "Los numeros de la diagonal principal son: \n");

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

diagonal=diagonal+matriz[i][i];

printf( " %i ", matriz[i][i] );

}

printf( "\n");

printf ("El valor de la suma diagonal principal es : %i\n", diagonal);

printf( "\n\n");

//d) Calcular el valor promedio de la matriz.

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

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

total= total+matriz[i][j];

}

}

printf ("El valor total de la matriz es: %i \n", total);

promedio = total/16;

printf ("El valor promedio de la matriz es: %.2f \n", promedio); //no pude hacer que salga con los decimales correspondienes

//Cuantos valores son mayores a ese promedio.

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

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

if(matriz[i][j]>promedio){

mayorAlPromTot++;

}

}

}

printf ("La cantidad de valores mayores al promedio es : %i\n", mayorAlPromTot);

printf( "\n\n");

//b) Calcular el valor maximo de la matriz

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

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

if (matriz[i][j]>mayor){

mayor=matriz[i][j];

imayor=i;

jmayor=j;

}

}

}

printf ("El mayor valor de la matriz es : %i\n", mayor);

printf ("La posicion del mayor valor es fila: [%i] columna: [%i]\n", imayor, jmayor);

printf( "\n\n");

printf("Los valores duplicados de la matriz son: \n");

//Se comparan los valores de la matriz consigo misma y cuando se encuentra un valor duplicado se guarda el valor y su ubicacion

//Primer ciclo

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

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

//Segundo ciclo

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

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

if (matriz[k][l]==matriz[i][j] &&(k!=i||l!=j)) //superimportante el condicional --> &&(k!=i||l!=j))

{

matrizValor[i][j]=matriz[i][j];

matrizFilas[i][j]=i;

matrizColumnas[i][j]=j;

}

}

}

}

}

//Impresion de los valores duplicados encontrados con su ubicacion

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

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

if (matriz[i][j]==matrizValor[i][j])

{

printf("Valor : %i \n", matrizValor[i][j]);

printf("La ubicacion del valor es: fila %i columna %i \n", matrizFilas[i][j], matrizColumnas[i][j]);

}

}

}

printf ("Presione 0 para salir");

scanf ("%i", &salir); //Agregado ya que sino el ejecutable termina sin mostrar los resultados

return 0;

}