

Fecha 15 / 6 / 2020

Evaluación de Aprendizaje número 02

1ra. Entrega / Recuperatorio

Apellido y Nombre:

DNI: 40345608

Comisión nro.: 63

Turno: mañana

Días Cursada: jueves y sábados

Materia: Elementos de Programación

Código:1024

La empresa de transporte "FexDex." realiza viajes en todo nuestro país, llevando distintos tipos de mercadería. Posee una flota de 50 camiones. Cinco de sus camiones, con los números 46 a 50, están equipados especialmente para traslado de mercadería que necesita frío. Esta empresa, nos solicita la confección de un programa que realice distintos informes para su Departamento Ejecutivo.

A) Se nos suministra unas planillas en hojas de oficio, teniendo que hacer la carga de la misma (fin de dato un camión con número = 0, donde se registró la siguiente información: ➤ Número de Camión. (1 a 50) ➤ Mes de recaudación (1 a 12) ➤ Importe acumulado en el mes. ➤ Cantidad de Kilómetros recorridos en el mes.

B) Debemos obtener de los datos suministrados los siguientes informes: 01. Total, de Kilómetros recorridos por camión durante todo el año. 02. De los Camiones con Equipamiento Especial (46 a 50), informar cuales no trabajaron, Nro. de Camión y Leyenda " Sin Recaudación".

03. Mostrar en forma ordenada de mayor a menor, los Kilómetros realizados por el Camión 13.

04. Hay que informar que mes, tiene menos Kilómetros recorridos durante este periodo. 05.

Informar Mes y Número de Camión, que Ingreso Mayor Importe Acumulado.

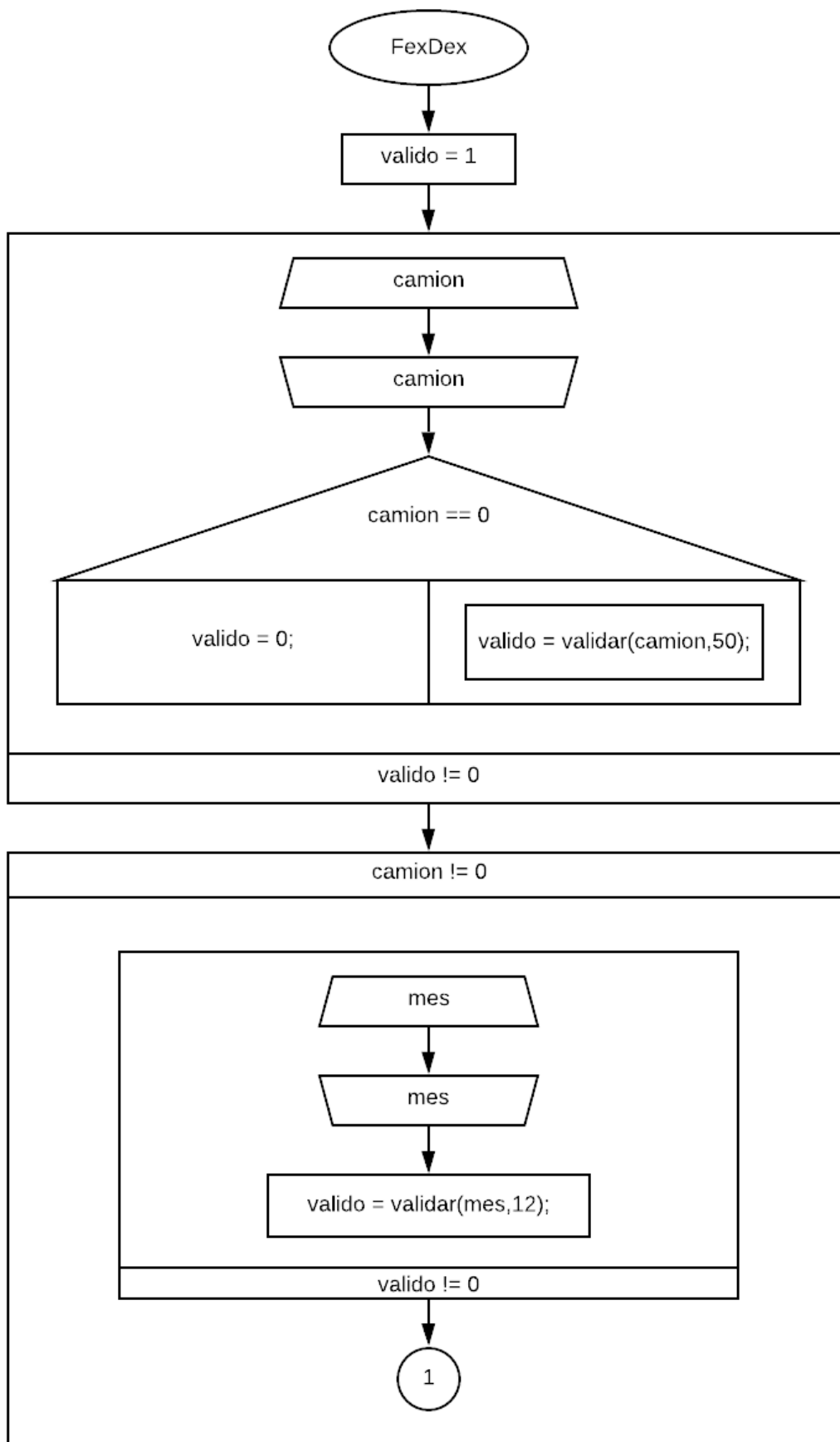
Realizar las siguientes funciones:

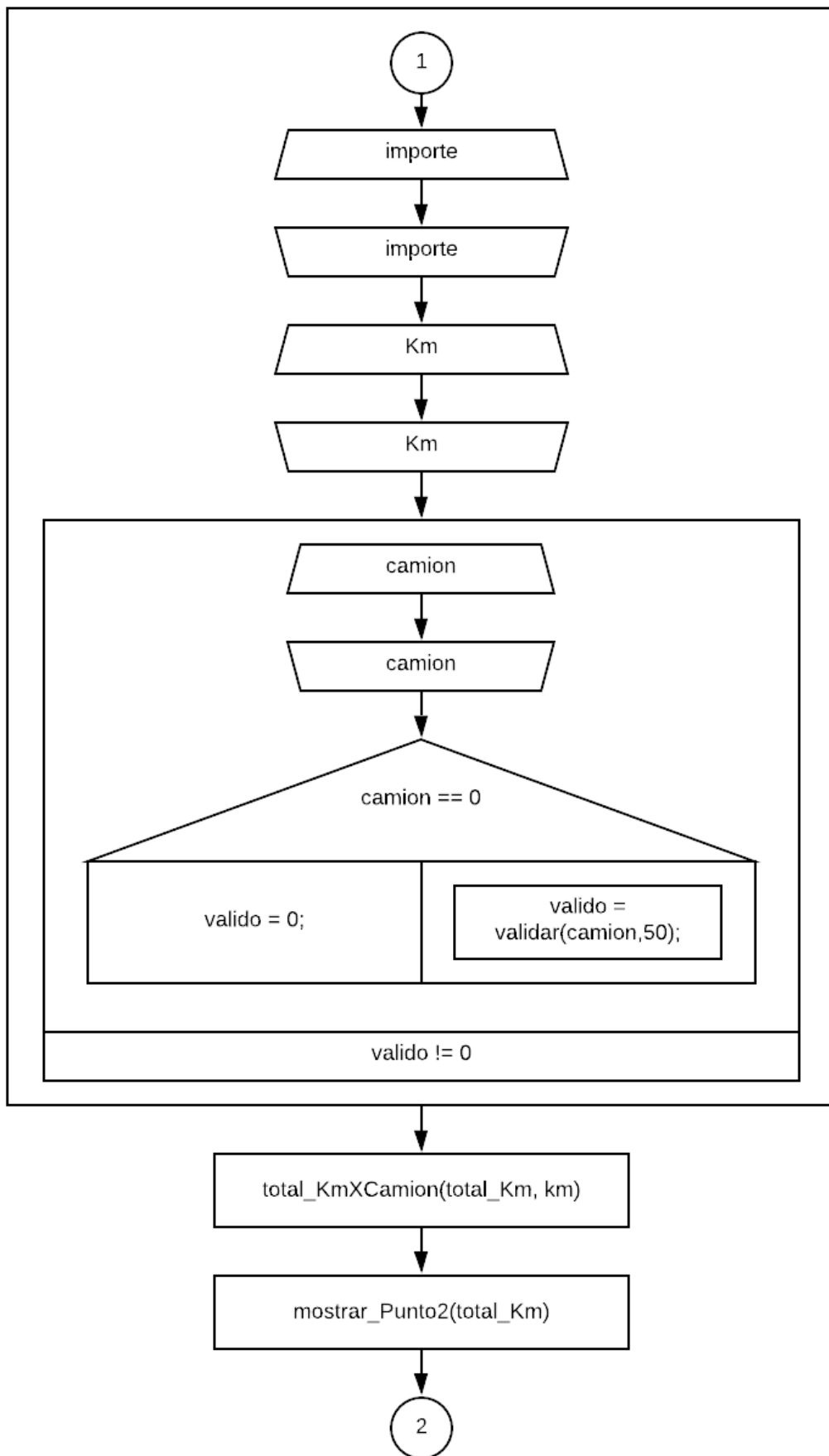
1) Ordenar () para el punto 03.

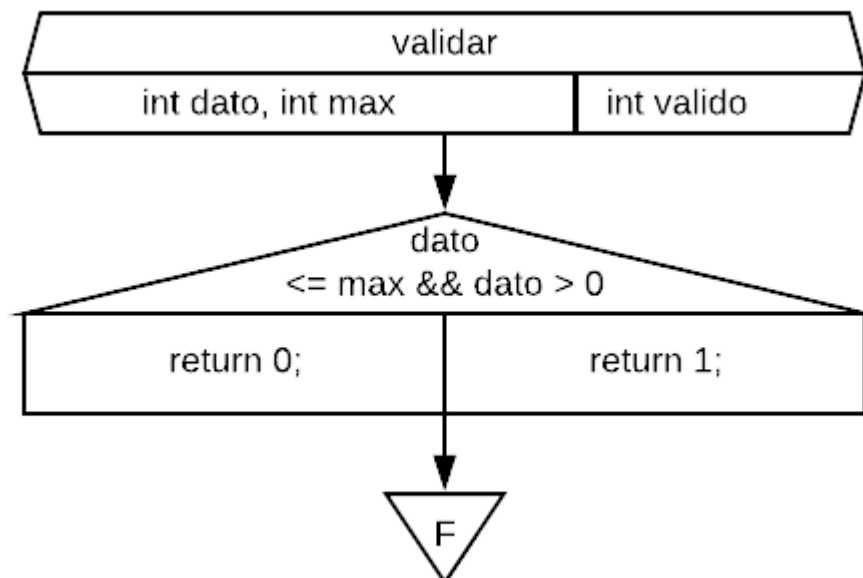
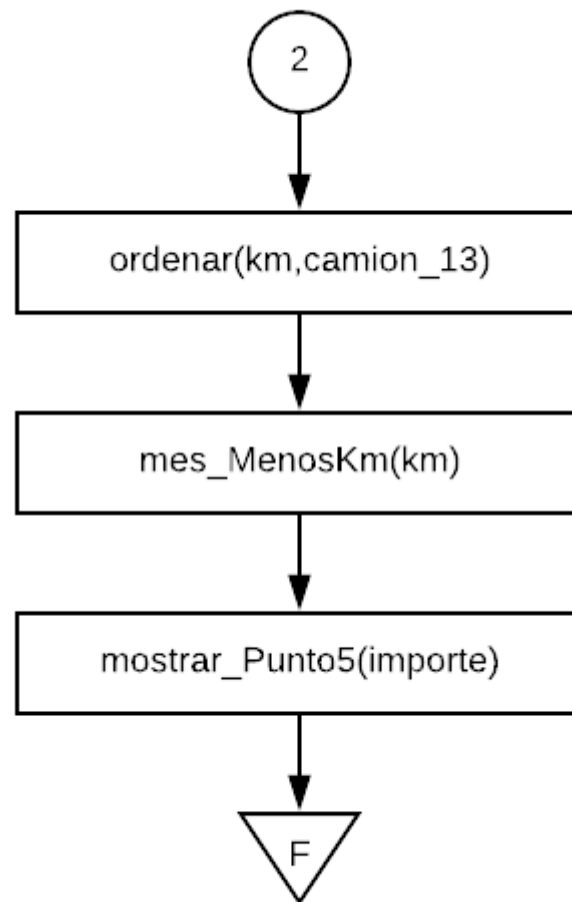
2) Mostrar_Punto5()

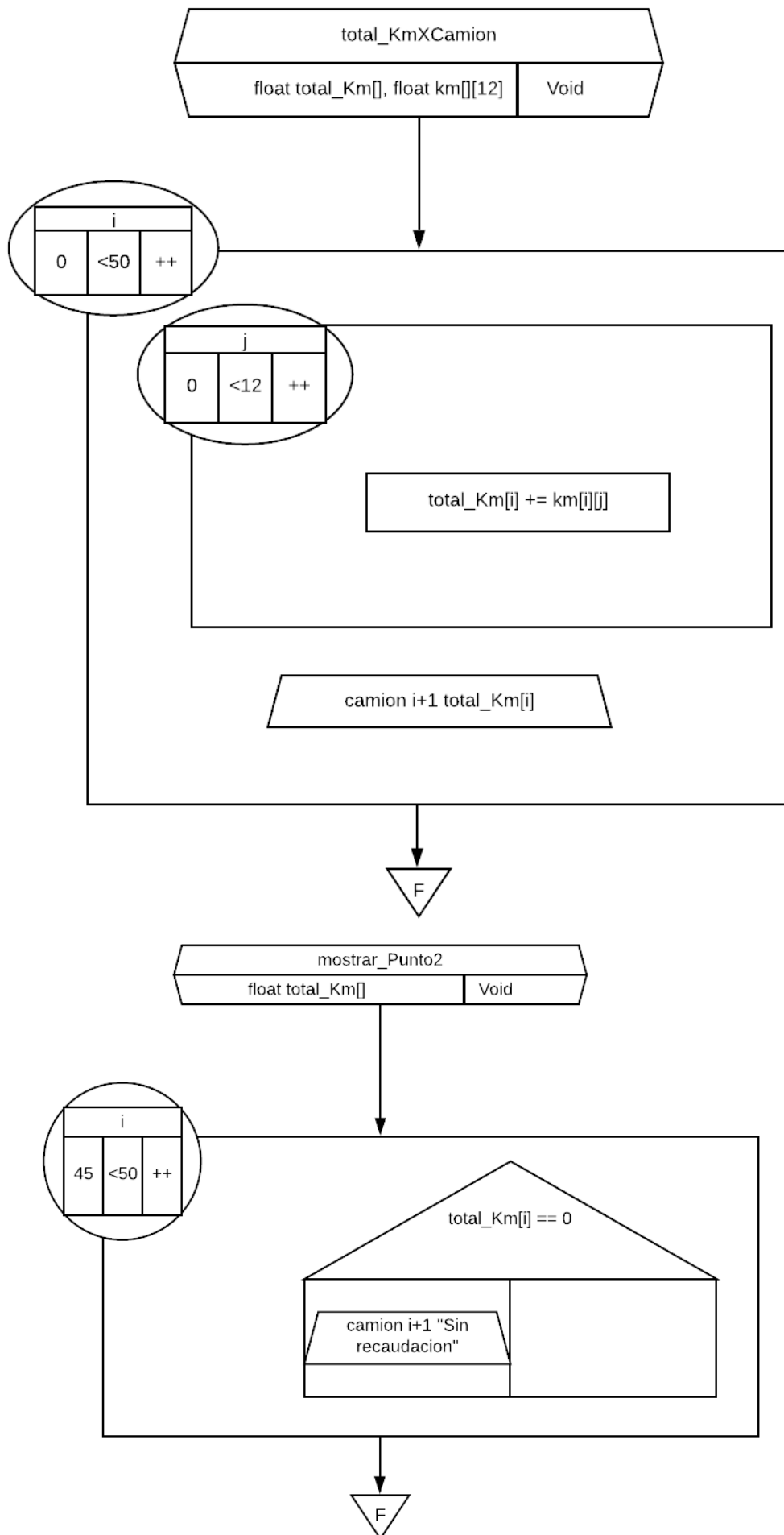
3) Validar () para Número de Camión y Mes de recaudación.

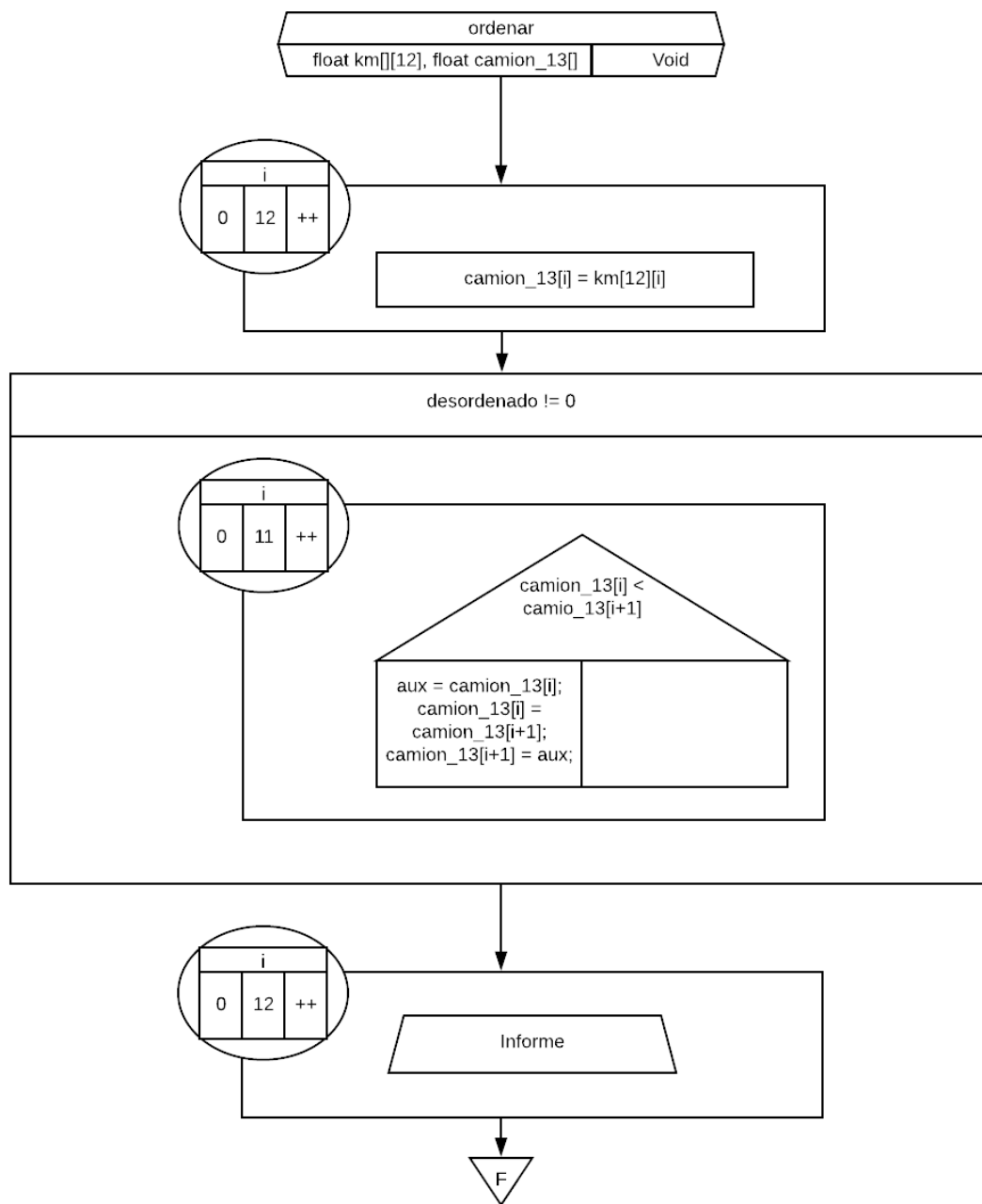
4) Mostrar_Punto2 () .

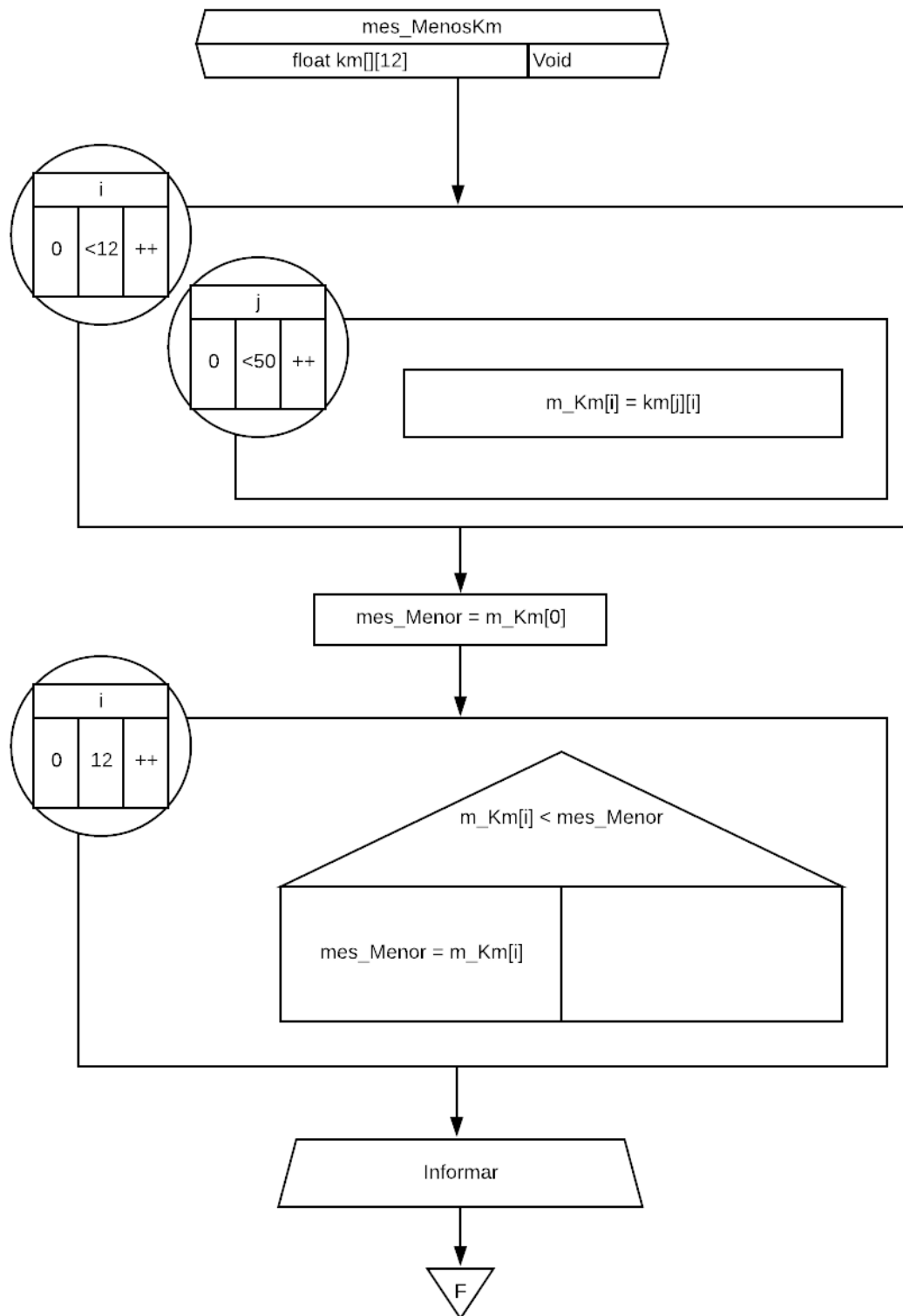


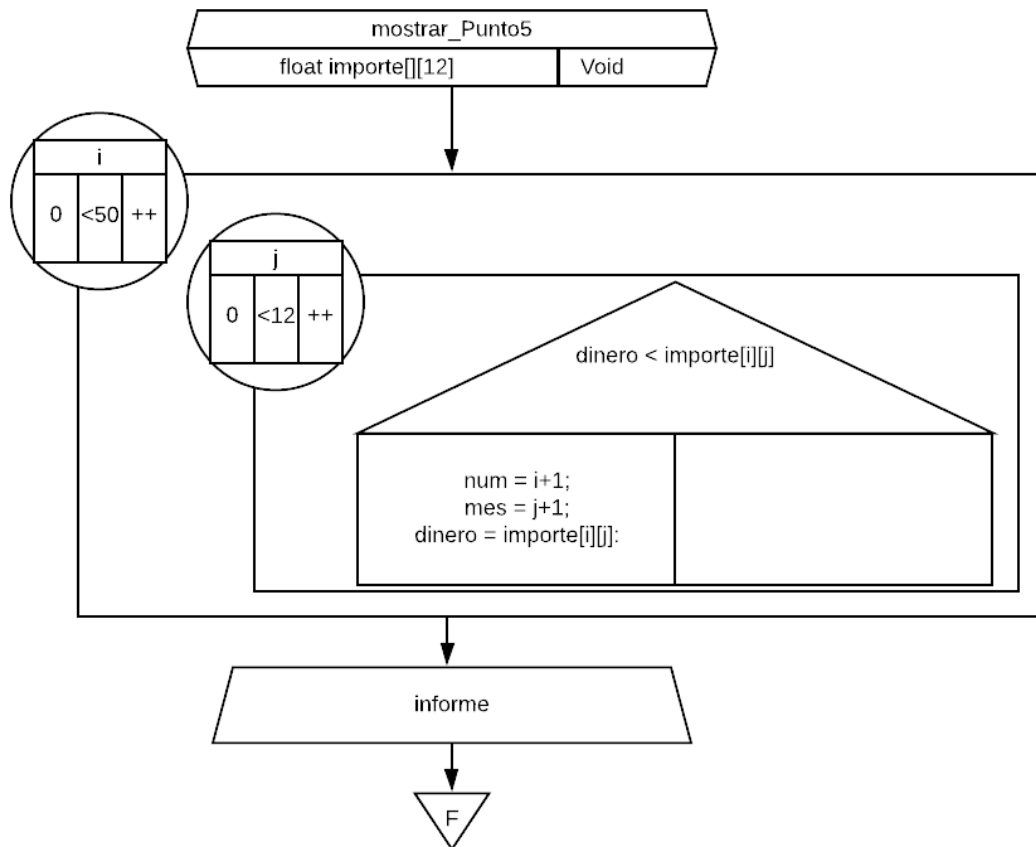












Código

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int validar(int , int );
```

```
void total_KmXCamion(float [], float[][12]);
```

```
void mostrar_Punto2(float []);
```

```
void ordenar(float[][12], float []);
```

```
void mes_MenosKm(float[][12]);
```

```
void mostrar_Punto5(float[][12]);
```



```

int main()
{
    int valido, camion, mes;
    float importe[50][12] = {{0}};
    float km[50][12] = {{0}};
    float total_Km[50] = {0};
    float camion_13[12] = {0};

    printf("\n \t\t Bienvenidos al sistema de carga de -FedEx- . \n\n");

    valido = 1;
    //Validar numero de camion
    do
    {
        printf("\n Ingrese el numero de camion 1 - 50 (o 0 para finalizar) :");
        scanf("%d", &camion);
        if(camion == 0)
        {
            valido = 0;
        }
        else
        {
            valido = validar(camion,50);
        }
    } while (valido != 0);

    //Carga de datos
    while (camion != 0)
    {
        //Validar numero de mes
        valido = 1;
    }
}

```

```

do
{
    printf("\n Ingrese el mes a cargar 1 - 12 :");
    scanf("%d", &mes);
    valido = validar(mes,12);
} while (valido != 0);

//Corregir datos para concordar con el indice
camion -= 1;
mes -= 1;

//Ingreso importe
printf("\n Ingrese el importe acumulado : ");
scanf("%f", &importe[camion][mes]);

//Ingreso km's
printf("\n Ingrese la cantidad de km realizados : ");
scanf("%f", &km[camion][mes]);

//Vuelta a comenzar del ciclo de carga -----
valido = 1;

//Validar numero de camion
do
{
    printf("\n Ingrese el numero de camion 1 - 50 (o 0 para finalizar) :");
    scanf("%d", &camion);
    if(camion == 0)
    {
        valido = 0;
    }
    else
    {
        valido = validar(camion,50);
    }
} while (valido != 0);

```

```
}
```

```
//Finaliza la carga
```

```
printf("\n\nLa carga de datos a finalizado satisfactoriamente!\n");
```

```
system("pause");
```

```
system("cls");
```

```
//-----
```

```
//Llamada a la funcion para sumar el total de km's de cada camion
```

```
total_KmXCamion(total_Km,km);
```

```
system("pause");
```

```
system("cls");
```

```
//-----
```

```
//Camiones de frio sin trabajar
```

```
printf("\n A continuacion se muestran los camiones con equipamiento especial que no  
trabajaron. \n\n");
```

```
mostrar_Punto2(total_Km);
```

```
system("pause");
```

```
system("cls");
```

```
//-----
```

```
//Muestra de mayor menor los km realizados por el camion 13
```

```
printf("\n A continuacion se lista de mayor a menor los Km realizados en cada mes por el  
camion numero 13.\n\n");
```

```
ordenar(km,camion_13);
```

```
system("pause");
```

```
system("cls");
```

```
//-----
```

```
//Busca el mes con menos km's realizados y lo informa
```

```
printf("\n A continuacion se muestra el mes con menos km realizados y la cantidad. \n\n");
```

```
mes_MenosKm(km);
```

```
system("pause");
```

```
system("cls");
```

```
//-----
```

```

//Se busca el mes y el camion con mayor importe del ano

printf("\n A continuacion se muestra el numero de camion y el mes en el que obtuvo el
mayor importe acumulado. \n\n");

mostrar_Punto5(importe);

system("pause");

return 0;
}

```

```

int validar(int dato, int max)
{
    if(dato <= max && dato > 0)
    {
        return 0;
    }
    else
    {
        printf("\n Dato ingresado erroneo, intentelo nuevamente!");
        return 1;
    }
}

```

```

void total_KmXCamion(float total_Km[], float km[][12])
{
    int i, j;

    printf("\n Numero de camion \t\t Total de Km\n\n");

    for(i = 0; i < 50; i++)
    {
        for(j = 0; j < 12; j++)
        {

```

```

        total_Km[i] += km[i][j];
    }

    //Muestras en pantalla los resultados de las operaciones
    printf("\n %15d\t\t %10.2f", i+1, total_Km[i]);
}
printf("\n");
}

```

```

void mostrar_Punto2(float total_Km[])
{
    int i;

    for(i = 45; i < 50; i++)
    {
        if(total_Km[i] == 0)
        {
            printf("\n Camion %d \t Sin recaudacion.\n",i+1);
        }
    }
}

```

```

void ordenar(float km[][12], float camion_13[])
{
    int i, desordenado, mes_Aux;
    int meses[12] = {1,2,3,4,5,6,7,8,9,10,11,12};
    float aux;

    //Copia los datos del camion a un vector individual
    for(i = 0; i < 12; i++)
    {
        camion_13[i] = km[12][i];
    }
}

```

```

}

//Ordena los datos
desordenado = 1;
while (desordenado != 0)
{
    desordenado = 0;
    for(i = 0; i < 11; i++)
    {
        if(camion_13[i] < camion_13[i+1])
        {
            aux = camion_13[i];
            camion_13[i] = camion_13[i+1];
            camion_13[i+1] = aux;

            mes_Aux = meses[i];
            meses[i] = meses[i+1];
            meses[i+1] = mes_Aux;

            desordenado = i;
        }
    }
}

//Muestra los datos de mayor a menor del camion 13
printf("\n Mes \t\t Km \n\n");
for(i = 0; i < 12; i++)
{
    printf("\n %3d \t\t %6.2f \n\n", meses[i], camion_13[i]);
}
}

```

```

void mes_MenosKm(float km[][12])
{
    int i, j, numero_Mes = 1;

```

```

float mes_Menor;

float m_Km[12] = {0};

//Suma el total de Km por mes
for(i = 0; i < 12; i++)
{
    for(j = 0; j < 50; j++)
    {
        m_Km[i] += km[j][i];
    }
}

//Establece el primero como el mes menor y lo compara con el resto
mes_Menor = m_Km[0];

//Recorre el vector buscando el menor
for(i = 0; i < 12; i++)
{
    if(m_Km[i] < mes_Menor)
    {
        mes_Menor = m_Km[i];
        numero_Mes = i;
    }
}

//Informa el mes con menos km's al finalizar
printf("\n En el mes %d se realizo la menor cantidad de km : %.2f \n", numero_Mes + 1,
mes_Menor);
}

void mostrar_Punto5(float importe[][12])
{
    int mes, num, i, j;

    float dinero = 0;

```

```
for(i = 0; i < 50; i++)
{
    for(j = 0; j < 12; j++)
    {
        if(dinero < importe[i][j])
        {
            num = i + 1;
            mes = j + 1;
            dinero = importe[i][j];
        }
    }
}

printf("\n El camion numero %d en el mes %d fue el que obtuvo el mayor importe
acumulado, siendo un total de $%.2f \n", num, mes, dinero);
}
```