



Apellido y Nombre :

Legajo:

Nota:

NOTA:

- Para aprobar el Examen debe tener como mínimo más del 50% del puntaje de cada ejercicio.
- Para los puntos 1 y 2, SOLO debe desarrollar:
 - a) el Procedimiento/Función que resuelva lo pedido
 - b) La definición de Tipos/Variables usadas y
 - c) Del Programa Principal solo la invocación a lo desarrollado en a)
 - d) Asumir que las estructuras y Archivos ya fueron cargados.

1.-(3 Ptos.)Desarrollar una única función recursiva entera eficiente que, a partir de una matriz A de enteros positivos de NxM, devuelva la cantidad de columnas cuya suma es mayor a un K dado (K es igual para todas las columnas).

Ejemplo; N=5 M=4 K=25

| | | | |
|---|----|----|----|
| 2 | 3 | 2 | 8 |
| 1 | 5 | 15 | 9 |
| 5 | 1 | 1 | 7 |
| 3 | 12 | 1 | 6 |
| 4 | 10 | 1 | 11 |

Respuesta = 2 (la 2° y la 4°)

2.- (4 Ptos.)Se tiene un archivo que contiene la información de las Carpas de un Balneario, y otro con los Consumos de los ocupantes de las mismas realizados en el último día, según el siguiente diseño:

| Archivo CARPAS | Archivo Consumos (*) |
|--|--|
| <ul style="list-style-type: none"> • NroCarpa (entero, campo de secuencia, clave primaria) • DNI_Cliente • ConsumoAcumulado | <ul style="list-style-type: none"> • NroCarpa (entero, campo de secuencia, clave secundaria) • Concepto • Importe |

Se pide, recorriendo una sola vez los archivos:

- a) Actualizar el archivo Carpas, poniendo al día el importe acumulado de consumos de cada carpa y eliminar aquellas que poseen un consumo cuyo concepto sea “Cierre” listando los datos actualizados del registro eliminado.
 (*) En caso de “Cierre”: Este Concepto aparecerá como último consumo del día y el Importe asociado es \$0.
- b) Por fin de proceso informar por pantalla:
 - Nro de Carpa que más consumió en el día. (No incluir los consumos erróneos).
 - Cantidad de Carpas sin consumo en el día
 - Cantidad de consumos erróneos (carpas inexistentes)

3.- (3 Ptos.)El siguiente programa ingresa enteros desde un archivo “Números.txt” y los va insertando de forma ordenada descendente en un vector:

```

Program xxxx;
Const
  MAXELEM = 20;
Type
  TV= array[1..MAXELEM] of .....;
```



Apellido y Nombre :

Legajo:

Nota:

```
Var
  A:TV;Arch:text; I,N,j:byte; Num:integer;
Begin
Assign(..... ,.....);
Reset(.....);
N:= ...;
While not eof(Arch) do
  Begin
    Read(Arch,Num);
    I := 1;
    While (I <= N) ..... (Num ... A[I]) do
      ....;
    If I = N+1 then
      .....
    Else
      Begin
        For j := N ..... I do
          A[j+1] := A[...];
        A[I-1] := Num;
      End;
    .....
  End.
```

Se pide:

- completar el código en las líneas punteadas y
- rescribir el programa para mejorar la eficiencia del código