**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.- (4 ptos)**Un balneario cuenta con un archivo VERANO2021 con los consumos de los clientes que alquilaron carpa ó sombrilla en la temporada 2021 y otro archivo con los clientes que dejaron hecha una reserva durante el 2020.

Las Tarifas fueron: Carpa $ 80.000, Sombrilla $ 50.000 y cada Visita $200

|  |  |
| --- | --- |
| VERANO2021 | RESERVAS |
| * Cliente (ANU5 campo de secuencia, clave primaria) * Nombre * CodCarSom(ANU 3)   (Carpas C01..C50 - Sombrillas S01..S99)   * Cantidad de visitantes * Gastos de consumos | * Cliente (ANU5 campo de secuencia, clave primaria) * Importe reserva |

|  |
| --- |
| HISTÓRICO |
| CodCarSom  Cant |

Se tiene además un archivo histórico con la cantidad de veces que se alquiló cada carpa y sombrilla con anterioridad al 2021,

Se pide, actualizar el Histórico a partir de VERANO2021 y emitir el siguiente listado:

*Liquidación Temporada 2021*

*Cliente Consumos Reserva(\*) Importe Total****(\*\*)***

*Xxxxx $ 9999.99 $ 9999 $ 99999.99*

*Xxxxx*  *$ 9999.99*  *$ 9999*  *$ 99999.99*

*. . . . . . . .*

*Promedio de Importe Total por cliente: $ 99999.99*

***(\*) Si no reservó corresponde 0***

***(\*\*)Importe Total****= Valor carpa ó sombrilla + Consumos + importe\* Visitantes - Reserva*

Informar además:

1. Monto total en reservas hechas y NO usadas en el 2021.
2. Cantidad de clientes que no hicieron reserva previa.
3. Código de carpa con mayor gasto de consumos.

**2.-(3 ptos)** Sea Mat una matriz entera de NxN, desarrollar una función entera **recursiva** que devuelva el máximo de los elementos simétricos iguales (incluir los de la diagonal)

Ejemplo N = 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **8** | -2 | 16 | -2 | 10 |  |
| -2 | **14** | -13 | 15 | **5** |  |
| 7 | 10 | -**15** | **21** | 1 | =>**21** |
| 12 | -20 | **21** | **-6** | 36 |  |
| -5 | **5** | -12 | -23 | **20** |  |

**3.- (3 ptos)** El siguiente fragmento de código verifica si en un vector V de enteros hay algún elemento positivo en las posiciones pares.

program Final\_20200312\_ejer3a;

var

V: array [1..50] of . . .;

i, N, contador:byte:

aux:. . .;

begin

contador:= . . . ;

For i:=1 to N do

Begin

aux:= . . . mod . . . ;

if aux . . . . . . then

if V[i] . . . . . . then

contador:=contador+ 1;

end;

if Contador . . . . . . then

Writeln(. . . . . . .)

else

Writeln(. . . . . . .);

end.

Se pide:

1. **Completar** el código en las líneas punteadas
2. **Calcular el tiempo de ejecución** del FOR encerrado en el recuadro.
3. **Rescribir** el programa para mejorar la eficiencia de la solución, evitando consultas innecesarias.