

Ej 1.- Indicar V o F, justificando o ejemplificando adecuadamente (de lo contrario tendrá puntaje cero)

a) Las rotaciones en un árbol AVL desbalanceado, se determinan siguiendo dos niveles desde el nodo desbalanceado considerando el signo del factor de equilibrio.

b) Si se aplica el recorrido en amplitud sobre un digrafo conexo, es posible asegurar que la pila nunca va a vaciarse.

Ej 2) Un hotel con 20 habitaciones cuenta con una lista L que mantiene las reservas ya confirmadas del mes de julio conteniendo en cada nodo: *Habitacion (1..20) (ordenada, no se repite), Capacidad, Precio por persona, Sublista de reservas confirmadas: en cada nodo contiene IdClienteResp (Cadena de 8), Ocupantes (>0), Dia entrada (1 a 30) (ordenada), Dia salida (2 a 31)*

A) Desarrollar en C subprogramas para:

i) A partir del recorrido de la lista L , generar una matriz M de 20×31 que contenga en cada posición, la cantidad de huéspedes confirmados. A partir del recorrido recursivo de M , informar la cantidad de habitaciones que estarán ocupadas todos los días de las vacaciones de invierno (17 al 28 de julio)

ii) En una Pila P se tienen los números de habitación y los días en que ciertas habitaciones deberán clausurarse por una filtración de humedad. Utilizando el TDA Pila, y considerando que en cada elemento se tiene H (no se repite en P), D , C , siendo H la habitación, D día de inicio, C cant de días consecutivos (puede exceder el mes de julio) se pide mediante un único recorrido de L :

- eliminar de L todas las reservas de las habitaciones que se vean afectadas en al menos un día según los datos de P ;
- hallar y mostrar cuánto dinero le cuesta al hotel la baja de estas reservas de Julio;

NOTAS: P puede perderse. No eliminar habitaciones de L

B) Definir el tipo de P asumiendo que es estática. Desarrollar $SacaP()$. Indicar en qué archivo(s) se encuentran estas definiciones.

Ej 3.- (Utilizar TDA N-Ario) Dados un árbol AN N-Ario de enteros y un árbol binario AB que proviene de la transformación de un árbol general de enteros, desarrollar una función *int* que obtenga y devuelva la cantidad de elementos de AN que verifican que, si su grado es G , G estaba en el árbol general en un nivel menor a K . (K dato positivo) [Si la solución se resuelve mediante una función void, el puntaje obtenido no superará la mitad del asignado]

Ej 4.- Dado un grafo acíclico de N vértices (numerados de 1 a N) con aristas ponderadas almacenado en la mitad inferior de una matriz de adyacencia, determinar si todos los vértices pares (2, 4 etc) tienen el mismo grado. [No se permite el uso de estructuras auxiliares]

Julio 2023

2 - Hotel con 20 habitaciones

Lista de reservas { Habitación (1..20)
Capacidad
Precio por persona
Sublista de confirmadas { IdClienteResp
Ocupantes
Dia de entrada/salida

A: Desarrollar en C subprogramas para

i) A partir del recorrido de la lista L, generar una matriz M de 20×31 que contenga en cada posición la cantidad de huéspedes confirmados.

• A partir del recorrido recursivo de M, informar la cantidad de habitaciones que estarán ocupadas todos los días de las vac. de inv. 17-28

main.c

```
#include <stdio.h> #include <stdlib.h> #include <string.h> #include "TDAPila.h"
#define D 31 #define H 20 #define MAX 100 #define ID 9
typedef struct nodito {
    char IdClienteResp[ID];
    unsigned int Ocupantes, DE, DS;
    struct nodito *sig; } nodito;
typedef nodito *Tsub;
typedef struct nodo {
    unsigned int Habitación, Capacidad;
    float PrecioPersona;
    Tsub SubReservas;
    struct nodo *sig; } nodo;
typedef nodo *Tlista;
```



```
void GenerarM(Tlista LH, int Mat[][0]) {
```

```
    int i, j;
```

```
    Tlista actH;
```

```
    TSub acts;
```

```
    actH = LH;
```

```
    while (actH != NULL) {
```

```
        i = actH -> Habitación - 1;
```

```
        acts = actH -> SubReservas;
```

```
        while (acts != NULL) {
```

```
            j = acts -> DE - 1;
```

```
            for (; j < acts -> DS - 1; j++)
```

```
                Mat[i][j] = acts -> Ocupantes;
```

```
            acts = acts -> sig;
```

```
        } actH = actH -> sig;
```

```
    }
```

```
int CantOculnr(int Mat[][0], int i, int j, int Hab, int Dinicio, int Dfin) {
```

```
    if (i == Hab)
```

```
        return 0;
```

```
    else
```

```
        if (j == Dfin)
```

```
            return 1 + CantOculnr(Mat, i + 1, Dinicio, Hab, Dinicio, Dfin);
```

```
        else
```

```
            if (Mat[i][j] != 0)
```

```
                return CantOculnr(Mat, i, j + 1, Hab, Dinicio, Dfin);
```

```
            else
```

```
                return CantOculnr(Mat, i + 1, Dinicio, Hab, Dinicio, Dfin);
```

```
}
```

4- Dado un digrafo aciclico con aristas ponderadas almacenado en la mitad inferior de una matriz de adyacencia, determinar si todos los vertices pares tienen el mismo grado

```
int TienenTodos(int Mat[100][100], int N) {
```

```
    int i, j, cumple, grado, gradoAnt;
```

```
    cumple = 1;
```

```
    i = 1;
```

```
    while (i < N && cumple) {
```

```
        j = 0;
```

```
        grado = 0;
```

```
        while (j < N && cumple) {
```

```
            if (j < i)
```

```
                grado += Mat[i][j] != 0;
```

```
            else
```

```
                grado += Mat[j][i] != 0;
```

```
            if (grado > gradoAnt && i != 1)
```

```
                cumple = 0;
```

```
            j++;
```

```
        }
```

```
        if (i == 1)
```

```
            gradoAnt = grado;
```

```
        else
```

```
            if (grado < gradoAnt)
```

```
                cumple = 0;
```

```
        i += 2;
```

```
    }
```

```
    return cumple;
```

```
}
```