

UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ  
FACULTAD DE INGENIERÍA  
ÁREA DE CIENCIAS DE LA COMPUTACIÓN  
1er. EXAMEN PARCIAL DE ESTRUCTURAS DE DATOS II

Nombre: Ricardo Daniel Campos Rodríguez Grupo: 223201 Calif: 6.05

1. El dueño de un vivero desea llevar un inventario de las plantas que comercializa y para ello requiere registrar para cada planta: su nombre común, la frecuencia de riego en días y el número de individuos en existencia, pero además por especie quiere registrar, en el caso de las de ornato, si es de sol o sombra y el color de su flor; en el caso de los árboles, si se trata de árbol frutal y la altura máxima de crecimiento. Considerando las siguientes estructuras realice lo que se pide

```
typedef struct {
    char sol; // s/n
    char *color;
} TOrnato;
```

```
typedef struct {
    char frutal; // s/n
    float altura;
} TArbol;
```

```
typedef struct {
    char *nombre;
    int diasriego;
    int existencia;
    char tipo; // o - Ornato a - Árbol
    void *generico;
} TPlanta;
```

- a) Realice una función que reciba el apuntador al arreglo con la información del inventario de plantas del vivero y regrese la cantidad de árboles frutales que tiene el vivero en existencia (no regresar solamente el número de especies). **[1.5 pts]**
- b) Realice una función que reciba el apuntador al arreglo con la información del inventario y regrese en un arreglo dinámico
- c) la información de los nombres de las plantas que dan flores de color rojo; los nombres deberán ir separados por una coma (',').

2. Un banco almacena la información de sus clientes en un arreglo dinámico. Cada cliente puede tener múltiples cuentas de débito, y cada cuenta está definida por un número de cuenta y un saldo. Este mes, el banco ha decidido cerrar todas las cuentas de débito que tengan un saldo de cero. Además, si un cliente no tiene ninguna cuenta con saldo positivo (es decir, todas sus cuentas tienen un saldo de cero), ese cliente también deberá ser eliminado del registro del banco.

```
typedef struct {
    int num_cuenta;
    float saldo;
} TCuenta;
```

```
typedef struct {
    TCuenta *cuentas;
    int cant_cuentas;
} TCliente;
```

```
typedef struct {
    TCliente *clientes;
    int cant_clientes;
} TBanco;
```

Utilizando las estructuras previamente definidas, escribe una función en C que elimine todas las cuentas con saldo cero de cada cliente. Si un cliente no tiene ninguna cuenta con saldo positivo después de eliminar las cuentas con saldo cero, eliminar también a ese cliente del arreglo dinámico. Asegúrate de liberar toda la memoria que ya no sea necesaria para evitar fugas de memoria. **[3.0 pts]**

3. Se tiene un arreglo dinámico de nCad cadenas de tamaño: tamCad, se requiere generar/crear a partir de este, una matriz dinámica que contenga solo las vocales de cada cadena como se ilustra en las siguientes figuras.

	J	U	A	N	\0	
	L	U	Z	\0		
	M	A	R	I	A	\0
	J	O	S	E	\0	
	D	I	E	G	O	\0

tamCad


- a) Proponer las estructuras de datos necesarias solo para la matriz resultante. Como se puede notar, la estructura debe incluir el número de vocales que tiene cada renglón. **[1.0 pts]**

- b) Escribir una función que reciba la matriz dinámica mencionada y genere la matriz dinámica requerida. **[2.0 pts]**











③ Ricardo Daniel Campos Rodríguez

223201

a) typedef struct {  
int nCad;  
char ~~int~~ \*voc;  
~~char \*\*Matriz;~~  
} TRes;

// Perdon :)

b) TRes \*resF(char \*\*M, int n) {

TRes \*res;

res->nCad = n;

res->~~voc~~ = (~~TRes~~ \*) malloc(sizeof(~~int~~) \* n);

res->~~Matriz~~ = (~~char~~ \*\*) malloc(sizeof(~~char~~ \*) \* n);

Este viene

for (int i = 0; i < n; i++) {

for (int j = 0; j < strlen(M[i]); j++)

if (vocal(\*(&(M[i] + j)))

\*(&(res->voc + i)) += 1;

for (int j = 0; j < strlen(M[i]); j++)

if (vocal(\*(&(M[i] + j)))

strcpy(res->Matriz + i, \*(&(M[i] + j)));

return res;

// En esta línea fuera del loop de j

res->Matriz + i = (char \*) malloc(sizeof(res->voc + i));

strcpy(res->Matriz + i, '0' \* (res->voc + i));

int vocal(char L) {

if (L == 'A' || L == 'E' || L == 'I' || L == 'O' || L == 'U')

return 1;

else

return 0;

0.75