

Parcial Algoritmos y Programación II - 7541 Curso Calvo (3º instancia)

05 de Diciembre de 2022

Padrón:

Punteros: Aprobado – no

Apellido y Nombre:

TDA: Aprobado - no

Listas: Aprobado - no

1) Indicar la salida por pantalla y escribir las sentencias necesarias para liberar correctamente la memoria.

```
int main(){

    int *A, *C, *F;
    int **B;
    char *D, *E;
    char G;
    int H;

    H = 71 + Ultimo Dígito Padrón;
    G = (char) H;
    A = new int;
    (*A) = H;
    F = A;
    C = F;
    (*A) = 65 + Ultimo Dígito Padrón;
    cout << (*A) << (*C) << (*F) << endl;

    B = &C;
    C = new int;
    (**B) = H - 2;
    cout << (*A) << (*C) << (**B) << endl;

    E = (char*) C;
    D = (char*) (*B);
    cout << (*E) << (*D) << endl;

    (*E) = G;
    cout << (*E) << (*C) << endl;

    F = (int*) D;
    while ( (*C) > 0) {
        (*C)--;
        cout << (*C) << (*D) << (**B) << endl;
        (*F) = (*F) - (**B);
    }

    // liberar la memoria
    // ...

    return 0;
}
```

2. Diseñar la especificación e implementar el TDA **Restaurante**. Debe proveer operaciones para:

- Construir el Restaurante a partir de la cantidad de mesas que tiene. Cada mesa se identifica con un número.
- solicitarMesa: un cliente solicita una mesa libre.
- cerrarMesa: deja libre la mesa y permite dejar propina.
- mayorPropina: devuelve el valor de mayor propina dada de todo el restaurante.
- mesaConMasPropina: devuelve el número de mesa que más propina recaudo.
- totalPropinas: devuelve el total recaudado por propinas del restaurante.

3. Implementar el método `buscarMensajeMasVotadoDelUsuario` de la clase `Moderador` a partir de las siguientes especificaciones:

```
class Moderador {
public:
    /* post: busca en la lista 'foros' el Mensaje más votado del autor 'usuarioBuscado' dentro de un Foro que no incluya la
     * temática 'tematicaBuscada'.
     */
    Mensaje* buscarMensajeMasVotadoDelUsuarioSegunTematica(Lista<Foro*>* foros,
        string usuarioBuscado, string tematicaBuscada);
};
```

```
class Foro {
public:

    /* post: inicializa el Foro sin Mensajes asignados.
     */
    Foro(string nombre);

    /* post: elimina todos los Mensajes del Foro
     */
    ~Foro();

    /* post: devuelve el nombre del Foro.
     */
    string obtenerNombre();

    /* post: devuelve todos los Mensajes del Foro.
     */
    Lista<Mensaje*>* obtenerMensajes();

    /* post: devuelve las temáticas del Foro
     */
    Lista<string*>* obtenerTematicas();
};
```

```
class Mensaje {
public:
    /* post: 'usuario' es el autor del mensaje con 'texto'
     * como contenido, sin votos asociados.
     */
    Mensaje(string usuario, string texto);

    /* post: devuelve el nombre del usuario autor del Mensaje.
     */
    string obtenerAutor();

    /* post: devuelve el contenido del Mensaje.
     */
    string obtenerContenido();

    /* post: suma un voto al Mensaje.
     */
    void votar();

    /* post: devuelve la cantidad del Ingrediente.
     */
    unsigned contarVotos();
};
```

Para aprobar es necesario tener al menos el 60% de cada uno de los ejercicios correctos y completos.

Duración del examen: 3 horas

Padron finalizado 0

656565

657373

II

K75

74J74

Padron finalizado 1

666666

667474

JJ

L76

75K75

Padron finalizado 2

676767

677575

KK

M77

76L76

Padron finalizado 3

686868

687676

LL

N78

77M77

Padron finalizado 4

696969

697777

MM

O79

78N78

Padron finalizado 5

707070

707878

NN

P80

79O79

Padron finalizado 6

717171

717979

OO

Q81

80P80

Padron finalizado 7

727272

728080

PP

R82

81Q81

Padron finalizado 8

737373

738181

QQ

S83

82R82

Padron finalizado 9

747474

748282

RR

T84

83S83

Fin