

Parcial Algoritmos y Programación II - 7541 Curso Calvo (1º instancia)

31 de mayo de 2018

Padrón:

Apellido y Nombre:

Punteros: APROBADO – DESAPROBADO **TDA:** APROBADO – DESAPROBADO

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

```
int main(){
    int *A, *C, *F;
    int **B;
    int H = 20 + ULTIMO_DIGITO_PADRON;

    A = new int[3];
    for (int i = 0; i < 3; i++) {
        A[i] = H + i;
    }
    C = new int;
    (*C) = A[1];
    F = A + 2;
    cout << (*F) << (*C) << A[0] << endl;

    B = new int*[3];
    B[0] = C;
    B[1] = F;
    B[2] = &H;

    cout << *B[1] << **B << *B[2] << endl;

    (*B[0]) = (*F) + 3;
    H++;
    A[2] = (*C) + 1;

    cout << *C << *B[1] << *B[2] << endl;

    F = C;
    C = A + 2;
    (**B) = A[0];
    A[2] = ** (B + 2);

    cout << H << (*C) << (*F) << endl;

    // liberar la memoria
    return 0;
}
```

2. Implementar para la clase Cola<T> las primitivas, indicando pre y post condiciones:

```
void acolar(T elemento)
```

```
T desacolar()
```

3. Implementar el método buscarMejorItinerario de la clase AgenteDeViajes a partir de las siguientes especificaciones:

<pre>class AgenteDeViajes { public: /* post: busca en 'itinerariosDisponibles' aquel Itinerario de menor costo, que al menos tenga Escalas en los lugares * indicados en 'lugaresPorVisitar' por 'duracionVisita' como mínimo. */ Itinerario* buscarMejorItinerario(Lista<Itinerario*>* itinerariosDisponibles, Lista<string*>* lugaresPorVisitar, unsigned int duracionVisita); }</pre>	<pre>class Escala { public: /* post: crea una Escala en el lugar indicado, * por los días dados en 'duracion' */ Escala(string lugar, unsigned int duracion) /* post: devuelve el nombre del lugar. */ string obtenerLugar(); /* post: devuelve la duración en días */ unsigned int obtenerDuracion(); }</pre>
<pre>class Itinerario { public: /* post: Itinerario creado sin Escalas por el costo indicado. */ Itinerario(unsigned int costo); /* post: devuelve el costo del itinerario en \$. */ unsigned int obtenerCosto(); /* post: devuelve todas las Escalas que lo conforman. */ Lista<Escala*>* obtenerEscalas(); /* post: libera todas las Escalas del Itinerario. */ ~Itinerario(); };</pre>	

4. Diseñar la especificación e implementar el TDA MesaDeRestaurante. Debe proveer operaciones para:

Los alumnos que tienen aprobado el parcialito de punteros y/o TDA no deben realizar el ejercicio 1 y/o 4 respectivamente.

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

Para cada método escribir pre y post condición, si recibe argumentos y cuáles, y si retorna un dato y cuál. De faltar ésto, se considerará el código incompleto.

Duración del examen : 3 horas

- crear la mesa a partir del número de lugares disponibles y el costo [\$] del cubierto por comensal.
- abrir la mesa indicando la cantidad de comensales.
- agregar el monto [\$] dado a la cuenta de la mesa.
- cerrar la mesa, totalizando la cuenta y devolviendo su monto [\$].
- calcular el monto promedio [\$] de las cuentas de la mesa.

Los alumnos que tienen aprobado el parcialito de punteros y/o TDA no deben realizar el ejercicio 1 y/o 4 respectivamente.
Para aprobar es necesario tener al menos el 60% de cada uno de los ejercicios correctos y completos.
Para cada método escribir pre y post condición, si recibe argumentos y cuáles, y si retorna un dato y cuál. De faltar esto, se considerará el código incompleto.
Duración del examen : 3 horas

0

222120

222120

252621

212120

1

232221

232221

262722

222221

2

242322

242322

272823

232322

3

252423

252423

282924

242423

4

262524

262524

293025

252524

5

272625

272625

303126

262625

6

282726

282726

313227

272726

7

292827

292827

323328

282827

8

302928

302928

333429

292928

9

313029

313029

343530

303029

Los alumnos que tienen aprobado el parcialito de punteros y/o TDA no deben realizar el ejercicio 1 y/o 4 respectivamente.

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

Para cada método escribir pre y post condición, si recibe argumentos y cuáles, y si retorna un dato y cuál. De faltar ésto, se considerará el código incompleto.

Duración del examen : 3 horas

TODOS

delete[] A

delete[] B

delete F

Los alumnos que tienen aprobado el parcialito de punteros y/o TDA no deben realizar el ejercicio 1 y/o 4 respectivamente.
Para aprobar es necesario tener al menos el 60% de cada uno de los ejercicios correctos y completos.
Para cada método escribir pre y post condición, si recibe argumentos y cuáles, y si retorna un dato y cuál. De faltar esto, se considerará el código incompleto.
Duración del examen : 3 horas