

Parcial Algoritmos y Programación II 7541 Curso Calvo (2º instancia)

10 de Junio 2022

Padrón:

Apellido y Nombre:

Mail:

Punteros: APROBADO – DESAPROBADO

TDA: APROBADO - DESAPROBADO

//Indicar la salida por pantalla y eliminar la memoria correctamente

```
int main(){
    int *A, *C, *F;
    int **B;
    char *D, *E;
    char G;
    int H;
    H = 68 + ULTIMO_DIGITO_PADRON;
    G = 'A';
    A = new int;
    (*A) = H;
    F = &H;
    cout << (*F) << (*A) << endl;;
    B = &F;
    (**B) = 65;
    cout << (*F) << (*A) << (**B) << endl;

    F = new int[3];
    (**B) = 70;

    for (int i = 0; i < 3; i++) {
        F[i] = (*A) + i;
    }
```

```
cout << (*F) << (**B) << endl;

D = (char*)A;
E = (char*)(F+1);
C = (*B);
C[1] = H;

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

for(int i = 0; i < 3; i++) {
    F[i] = 65;
}
C = C + 2;
cout << (*D) << (*E) << (*C) << endl;

return 0;
}
```

2) Implementar la clase BuscadorDeNoticias a partir de las siguientes especificaciones:

```
class BuscadorDeNoticias {
public:
    /* post: busca en noticias aquellas Noticias que tengan asociada todas las categorías dadas en conCategorias y que NO contengan ninguna de las categorías de sinCategorias,
    * teniendo un mínimo de categorías
    * Devuelve una nueva Lista con todas las Noticias encontradas. */
    Lista<Noticia*>* buscarNoticias(Lista<Noticias*>* noticias,
        Lista<Categoria*>* conCategorias, Lista<Categoria*>* sinCategorias, int cantidadMinimaDeCategorias);
};
```

```
class Noticia {
public:
    /* post: inicializa la noticia con el título y cuerpo
    * indicados, sin categorías asociadas.
    */
    Noticia(string titulo, string cuerpo);
    /* post: devuelve el título de la noticia.
    */
    string getTitulo();
    /* post: devuelve el cuerpo de la noticia.
    */
    string getCuerpo();
    /* post: devuelve las categorías asociadas a la noticia.
    */
    Lista<Categoria*>* getCategorias();
};
```

```
class Categoria {
public:
    /* post: inicializa la categoría con el nombre y la
    * descripción indicadas.
    */
    Categoria(string nombre, string descripcion);
    /* post: devuelve el nombre que la identifica.
    */
    string getNombre();
    /* post: devuelve la descripción de la categoría.
    */
    string getDescripcion();
};
```

3) Diseñar la especificación e implementar los TDAs que modelen un **Tren**, cumpliendo las siguientes características:

El **Tren** se compone de una serie de **Vagones**. Cada **Vagón** posee una capacidad de carga (medida en toneladas), una capacidad de pasajeros y una longitud (medida en metros). Se deben poder agregar y retirar **Vagones** del **Tren**. Agregar o retirar carga o pasajeros. Proveer métodos para conocer la longitud total, capacidad total y restante de pasajeros y capacidad total y restante de carga del **Tren**.

Solo rinde el tema que no tiene aprobado.

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

Para cada método debe 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.

Mantener apagados celulares, i-pod, reproductores de mp3, etc. durante el examen

Duración del examen : 2 horas y 45 minutos