Parcial Algoritmos y Programación II - 7541 Curso Calvo (3º instancia) 10 de Diciembre de 2013

Padrón: **Apellido y Nombre:**

Punteros: 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;
  char *D, *E;
  char G;
  int H;

  H = 70;
  G = (char) H;
  A = new int;
  (*A) = H;
  F = A;
  C = F;
  (*A) = 65;
  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. Implementar para la clase ListaSimplementeEnlazada el método

```
/* post: agrega elemento a la lista, insertándolo en la posición indicada por posicionElemento.
 */
template<class T>
void insertar(int posicionElemento, T elemento);
```

3. Implementar el método buscarReceteasPosibles de la clase Chef a partir de las siguientes especificaciones:

```
class Chef {
  public:
    /* post: busca en la lista recetasAlternativas una Receta que pueda ser preparada con los mismos Ingredientes que
     *       recetaPorSustituir. Considera que todos los Ingredientes de recetaPorSustituir sean Ingredientes de
     *       la otra Receta comparando su nombre y validando que la cantidad sea inferior.
     *       Devuelve la primera Receta que cumpla la condición, en caso de no encontrar ninguna devuelve NULL.
     */
    Receta* buscarRecetaSustituta(Receta* recetaPorSustituir, Lista<Receta*>* recetasAlternativas);
};
```

```
class Receta {

  public:

    /* post: inicializa la Receta sin Ingredientes asignados.
     */
    Receta(string nombre);

    /* post: elimina todos los Ingrediente de la Receta.
     */
    ~Receta();

    /* post: devuelve todos los Ingredientes de la Receta.
     */
    Lista<Ingrediente*>* getIngredientes();

    /* post: devuelve el nombre que identifica a la Receta.
     */
    string getNombre();
};
```

```
class Ingrediente {

  public:

    /* post: queda inicializado con el nombre y cantidad
     *       indicados.
     */
    Ingrediente(string nombre, float cantidad);

    ~Ingrediente();

    /* post: devuelve el nombre del Ingrediente.
     */
    string getNombre();

    /* post: devuelve la cantidad del Ingrediente.
     */
    float getCantidad();
};
```

4. Implementar una clase que modele un Polinomio: $P_{(x)} = c_0 + c_1x + c_2x^2 + \ldots + c_nx^n$
Operaciones:

- Construir una instancia como el polinomio unidad. $P_{(x)} = 1$
- Evaluar el Polinomio en un punto dado. $P_{(x=30)}$
- Agregar un término, indicando su grado y valor de coeficiente.
- Remover un término, indicando su grado y valor de coeficiente.
- Obtener el valor del coeficiente para el término de grado i.
- Cambiar el valor del coeficiente para el término de grado i.

Los alumnos que tienen aprobado el parcialito de punteros no deben realizar el ejercicio 1.
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