Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática
Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos.
1er Parcial. [2010-09-16]

ATENCIÓN: Para aprobar deben obtener un **puntaje mínimo** del 50 % en clases (Ej 1), 40 % en programación (Ej 2), y un 60 % sobre las preguntas de teoría (Ej 3).

[Ej. 1] **[clases (30pt)]**

a) **[lista (10pt)]** Escribir la implementación en `C++` del TAD lista (clase `list`) implementado por punteros **ó** cursores. Los métodos a implementar son `insert(p,x)`, `erase(p)`, `next()`/`iterator::operator++(int)`.

b) **[pila-cola (10pt)]** Escribir la implementación en `C++` de los métodos `push`, `pop`, `front`, y `top` de los TAD pila y cola (clases `stack` y `queue`), según corresponda.

c) **[map (10pt)]** Escribir la implementación en `C++` del TAD correspondencia (clase `map`) implementado con listas ordenadas. Métodos a implementar: `find(key)`.

[Ej. 2] **[Programación (total = 50pt)]** Recordar que en los ejercicios de programación **deben usar la interfaz STL**.

a) **[cutoff-map (15pt)]** *Consigna:* Implemente una función `void cutoff_map(map<int, list<int> > &M,int p,int q);` que elimina todas las claves que NO están en el rango `[p,q)`. En las asignaciones que quedan también debe eliminar los elementos de la lista que no están en el rango. Si la lista queda vacía entonces la asignación debe ser eliminada. *Por ejemplo:* Si `M={1->(2,3,4), 5->(6,7,8), 8->(4,5), 3->(1,3,7)}`, entonces `cutoff_map(M,1,6)` debe dejar `M={1->(2,3,4), 3->(1,3)}`. Notar que la clave 5 ha sido eliminada si bien está dentro del rango porque su lista quedaría vacía.
Restricciones: El programa no debe usar contenedores auxiliares.

b) **[compacta (20pt)]** *Consigna:* Escribir una función `void compacta(list<int> &L,stack<int> &S);` que va tomando un elemento entero `n` de `S` y, si es positivo, saca `n` elementos de `L` y los reemplaza por su suma. Esto ocurre con todos los elementos de `S` hasta que se acaben, o bien se acaben los elementos de `L`.
Por ejemplo: Si `L=(1,3,2,1,4,5,3,2,4,1)` y `S=(3,2,-1,0,2,5,2,-3)` entonces `L` debe quedar así `L=(6,5,8,7)`, y `S=(2,-3)` (es decir, sobran elementos de `S`).
Otro ejemplo: Si `L=(1,3,2,1,4,5,3,2,4,1,3,2,1,4,7)` y `S=(3,2,-1,0,2,5)` entonces `L` debe quedar así `L=(6,5,8,12,1,4,7)`, y `S=()` (es decir, sobran elementos de `L`).
Restricciones: El programa no debe usar contenedores auxiliares.

c) **[concat-map (15pt)]** *Consigna:* Escribir una función `void concat_map(map<int,list<int> >& M, list<int>& L);` tal que reemplaza los elementos de `L` por su imagen en `M`. Si un elemento de `L` no es clave de `M` entonces se asume que su imagen es la lista vacía.
Por ejemplo: Si `M={5->(3,2,5),2->(4,1),7->(2,1,1)}` y `L=(1,5,7,2,3)`, entonces debe quedar `L=(3,2,5,2,1,1,4,1)`.
Restricciones: El programa no debe usar contenedores auxiliares.

[Ej. 3] **[Preguntas (total = 20pt, 4pt por pregunta)]**

a) Ordenar las siguientes funciones por tiempo de ejecución. Además, para cada una de las funciones $T_1, \ldots, T_5$ determinar su velocidad de crecimiento (expresarlo con la notación $O(\cdot)$).

$$T_1 = 5 \cdot 2^n + 2n^3 + 3\,n!+$$
$$T_2 = 2 \cdot 10^n + \sqrt{3} \cdot n + \log_8 n+$$
$$T_3 = n^2 + 5 \cdot 3^n + 4^{10}$$
$$T_4 = 2.3 \log_8 n + \sqrt{n} + 5\,n^2 + 2\,n^5$$
$$T_5 = 1000 + 3 \log_4 n + 8^2 + 5 \log_{10} n$$

b) ¿Cuáles son los tiempos de ejecución para los diferentes métodos de la clase `map<>` implementada con **listas desordenadas** en el caso promedio?
Métodos: `find(key)`, `M[key]`, `erase(key)`, `erase(p)`, `begin()`, `end()`, `clear()`).

c) ¿Porqué decimos que $(n+1)^2 = O(n^2)$ si en realidad es siempre verdad que $(n+1)^2 > n^2$?

d) Discuta las ventajas y desventajas de utilizar listas doblemente enlazadas con respecto a las simplemente enlazadas.

e) ¿Qué ventajas o desventajas tendría implementar la clase *pila* en términos de **lista simplemente enlazada** poniendo el tope de la pila en el fin de la lista?