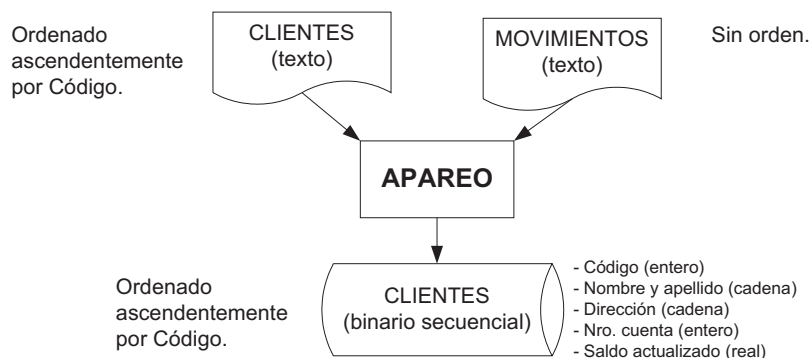


75.40 Algoritmos y Programación I – Segundo Parcial Integrador
Cátedra: Ing. Pablo Guarna
21 de diciembre de 2006

1. Programación Pascal (6 pts)

Desarrollar un programa Pascal que realice el siguiente apareo:



Las líneas de los archivos de entrada tienen las siguientes estructuras:

CLIENTES

Campo	Inicio	Ancho
Código	1	4
Nom. y apellido	5	30
Dirección	35	50
Nro. cuenta	85	4
Saldo	89	10

MOVIMIENTOS

Campo	Inicio	Ancho
Código	1	4
Monto	5	10

La invocación del programa debe realizarse utilizando argumentos por línea de órdenes, de la siguiente manera:

```
apareo.exe clientes.txt movimientos.txt clientes_out.dat
```

Para la lectura y parseo de una línea del archivo de texto de *clientes* deberá implementarse una función que retorne un puntero a un registro (en memoria dinámica) con los datos del cliente, según el siguiente prototipo:

```
function siguiente_cliente(var clientes_in: text): ^t_cliente;
```

Para la lectura y parseo de una línea del archivo de texto de *movimientos* deberá implementarse una función que retorne un puntero a un registro (en memoria dinámica) con los datos del movimiento, según el siguiente prototipo:

```
function siguiente_movimiento(var movimientos: text): ^t_movimiento;
```

El programa desarrollado debe estar completo, incluyendo tipos de datos, cuerpo principal, etc. Utilizar nombres representativos, y comentar aquellos fragmentos que, por su complejidad, lo requieran. Tener en cuenta las precauciones que deben tomarse por el uso de memoria dinámica.

2. Programación ANSI C (4 pts)

Desarrollar una función en lenguaje C que imprima en salida estándar la cadena recibida por parámetro, pero en forma rebatida y usando recursividad. La función debe ajustarse al siguiente prototipo:

```
void rebatir(char *str);
```

RUTINAS ÚTILES

Para Manejo de Argumentos por Línea de Órdenes

`function ParamCount: longint;`

Descripción: Retorna la cantidad de argumentos recibidos por línea de órdenes.

`function ParamStr(n: longint): string;`

Descripción: Retorna el n -ésimo argumento de línea de órdenes (n debe ser mayor o igual 0 y menor o igual a ParamCount). El 0-ésimo argumento (primero en la lista) es el nombre del comando ejecutado.

Para Procesamiento de Cadenas de Caracteres

`procedure Val(s: string; var v: ; var code: word);`

Descripción: Convierte el valor representado en la cadena s a su valor numérico, y lo almacena en el parámetro v , el cual puede ser de alguno de los siguientes tipos: longint, real o byte. A través del parámetro $code$, retorna 0 si la conversión se realizó con éxito.

`function Copy(s: string; index: integer; count: integer): string;`

Descripción: Retorna una cadena que es una copia de los $count$ caracteres comenzando desde la posición $index$.