

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**ESTUDIOS GENERALES CIENCIAS**

**TÉCNICAS DE PROGRAMACIÓN**

**Práctica N° 3**

**Semestre académico 2014-1**

Elaborado por el profesor del curso

**Notas:**

- NO se podrá emplear funciones, procedimientos ni operadores que manejen cadenas de caracteres.
- NO se podrán utilizar, registros ni objetos en esta práctica.
- NO se podrán emplear variables globales ni archivos auxiliares.
- Las soluciones deberán desarrollarse bajo un estricto diseño descendente, por lo que no se calificarán aquellos módulos que son llamados por otros que estén incompletos. Cada módulo no debe sobrepasar las 20 líneas aprox.
- No se pueden usar apuntes de clase ni calculadoras.

---

**PREGUNTA 1: (12 puntos)**

Una tienda de abarrotes contrató a un inexperto programador para que le confeccione una aplicación que le permita manejar las ventas de su local. El programa registra los datos de cada una de las ventas que se realizan a través de los días y luego imprime la boleta de venta. Si es cierto que la aplicación funcionó durante un tiempo, cuando la tienda necesitó que se amplíen las funcionalidades de la aplicación se dieron cuenta que ésta había sido tan mal elaborada por lo que decidieron despedir al programador y contratarlo a usted para que le solucione el inconveniente.

El problema se presenta porque la aplicación crea un solo archivo con la información de las ventas. El archivo es similar al que se muestra a continuación:

```
000001  3  10  2013  23874566
Julio José López Pérez
Av. ABC 123 Las Flores
234561
Vinagre blanco
2.0  5.28  10.56
...
12983476
Pan francés
25.0  0.25  6.25
-1  153.95
00002  3  10  2013  87653401
Ana Cecilia Roncal Neyra
...
```

El archivo contiene la siguiente información: En la primera línea aparece el número de la factura seguido de la fecha en la que se emitió y por último el DNI del cliente al que se le emitió la boleta. En la segunda línea se colocó el nombre del cliente y en la siguiente línea su dirección. A partir de allí viene la información de cada artículo que compró.

Por cada artículo se colocó en la primera línea el código del producto, en la segunda la descripción del mismo, y en la tercera la cantidad de productos que compró, su precio unitario y el subtotal. Una vez terminada la lista de productos que el cliente compró viene una línea que empieza con el valor -1 seguido del total facturado. A partir de allí vienen los datos de la siguiente boleta y así sucesivamente.

Como puede apreciar, además del tamaño del archivo que es extremadamente grande, la información que se encuentra en el archivo sólo permite imprimir las boletas de ventas.

Se le pide a usted que elabore un programa en Pascal que permita dividir este archivo en tres nuevos archivos. El primero se denominará "Clientes.txt", el segundo "Productos.txt" y el tercero "Boletas.txt". En el archivo de clientes debe guardar el DNI, nombre y dirección, en el archivo de productos debe aparecer el código, la descripción y precio unitario del producto, y en el archivo de boletas se debe almacenar sólo el número de factura, la fecha, el DNI del cliente, el código y la cantidad de los productos que adquirió en esa boleta.

Debido a que el archivo original es extremadamente grande no podrá cargar todo ese archivo en la memoria del computador, y por eficiencia, sólo podrá leer este archivo una sola vez en el programa. Sin embargo se sabe que la tienda tiene un estimado de alrededor de 400 clientes y 1000 productos.

Deberá tomar en cuenta que:

- En el archivo original a un mismo cliente pudo emitir innumerables boletas y que de igual manera pasa con los productos que pueden repetirse a través del archivo. Sin embargo, tanto en el archivo de clientes como en el de productos, los datos no pueden aparecer repetidos.
- Como el archivo original ha registrado los datos en un período de tiempo grande, un mismo cliente podría haber cambiado de dirección, por lo que en el archivo de clientes se debe guardar la última dirección del cliente que aparece en el archivo original, lo mismo pasa con los productos, su precio unitario podría cambiar por lo que sólo se guardará su último valor.
- Como un archivo de textos no se puede actualizar, deberá trabajar la información de los clientes y productos en arreglos. Debe tomar en cuenta que en los arreglos no pueden haber, en ningún momento, clientes o productos repetidos por lo que el proceso debe sacar el dato del archivo original, luego lo buscará en el arreglo correspondiente, si no lo encuentra lo debe colocar al final del arreglo, y si lo encuentra deberá verificar si cambió el precio o la dirección y reemplazar sólo ese dato.

## PREGUNTA 2: (8 puntos)

Hoy en día la información que se transmite a través de medios electrónicos es muy susceptible de ser capturada por terceros y que le den mal uso. Por esa razón que existen muchos métodos que permiten cifrar la información que se envíe de modo que si alguien la captura no lo pueda entender.

Se pide que elabore un programa que permita cifrar un archivo de textos, el método que se empleará es una variante del método que utilizó Julio César en la antigua Roma y que recibe precisamente el nombre de "Método César" o "Método de desplazamiento". El método se describe a continuación:

- En un archivo de textos se guarda en la primera línea la lista de caracteres que se van a emplear en los mensajes a cifrar (no necesariamente están todos los caracteres de la tabla ASCII), en la segunda línea se colocan una serie de valores enteros, la cantidad de valores puede variar entre 1 y 10, y los valores pueden ir desde 2 a n-2, siendo n el número de caracteres que se colocó en la primera línea. El archivo es similar al siguiente:

ABCDEF...WXYZabcd...xyz0123...89_.,:~+...()&? 5 12 7 23 19 3 35 ...
--

- El programa toma uno a uno los caracteres del archivo y los coloca en un arreglo del tipo: `type TArrChr = array [1..255] of char;` luego coge los enteros y los coloca en un arreglo de tipo: `TArrInt array [1..10] of Integer;`
- El programa luego empieza a leer uno a uno los caracteres de otro archivo que contiene el mensaje y a cada carácter se le aplica una transformación como se explica a continuación:
  - Se toma el primer número del arreglo (en el ejemplo 5) y en otro arreglo del mismo tipo `TArrChr` se colocan los caracteres a partir de la posición de número, esto es:

	1	2	3	4	5	6	7	8	9	...	73	74	75	76	77...
Arreglo original →	A	B	C	D	E	F	G	H	I	...	K	(	)	&	?
Arreglo nuevo →	E	F	G	H	I	J	K	L	M	...	?	A	B	C	D

Los primeros n-1 caracteres (en este caso 4) se colocan al final del arreglo nuevo.

- Se ubica el carácter en el arreglo original y se toma el carácter que se encuentra en la misma posición en el nuevo arreglo. Ese carácter se coloca en otro archivo que guardará el mensaje cifrado, por ejemplo si el carácter leído fuera 'G', el carácter escrito sería 'K'.

	1	2	3	4	5	6	7	8	9	...	73	74	75	76	77...
Arreglo original →	A	B	C	D	E	F	G	H	I	...	K	(	)	&	?
							↓								
Arreglo nuevo →	E	F	G	H	I	J	K	L	M	...	?	A	B	C	D

- Luego se procede de manera similar con el siguiente carácter del mensaje y el siguiente número. Si los números se acabasen se vuelven a tomar desde el principio.
- Si un carácter del mensaje no se encontrara en el arreglo se guarda en el mensaje sin modificar.

San Miguel, Viernes 30 de mayo del 2014