ESTUDIOS
GENERALES
CIENCIAS

PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

TÉCNICAS DE PROGRAMACIÓN

PRIMERA PRÁCTICA CALIFICADA
SEMESTRE ACADÉMICO 2020-2

Horario: Todos Duración: 110 minutos

Elaborado por los profesores del curso.

ADVERTENCIAS:

- **SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO. ESTO, Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .c O .h CON FECHA U HORA DE CREACIÓN ANTERIOR AL LABORATORIO SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.**

INDICACIONES:

- 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 30 líneas de código aproximadamente**.
- NO SE PUEDEN EMPLEAR ARCHIVOS DE DATOS AUXILIARES NI VARIABLES GLOBALES.
- En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.
- DEBE COLOCAR SU NOMBRE Y CÓDIGO EN CADA UNO DE LOS ARCHIVOS .h Y .c QUE EMPLEE EN SUS PROYECTOS.

Cree un proyecto en NetBeans con el nombre: PrimeraPractica2020-2 y en él desarrolle el programa que resuelva el problema que se describe a continuación.

Una entidad bancaria necesita una aplicación en lenguaje C que le permita manejar las cuentas de sus clientes, la aplicación deberá permitir que el usuario ingrese inicialmente el número de la cuenta (p. e.: 23541890) y el saldo que tiene la cuenta en ese momento (p. e.: 87410.89), Este último valor puede ser positivo, negativo o también cero (p. e.: 87410.89, -56387.56 o 0.0).

El programa deberá definir tres variables de punto flotante para manejar el dinero de la cuenta, los nombres de estas variables serán los siguientes: "**saldoInicial**", **haber**" y "**debe**", además debe definir una variable de tipo char denominada: "**estado**".

En la variable "**saldoInicial**" debe colocar el saldo leído, sin modificar. Luego si en el "**saldoInicial**" se colocó un valor positivo o 0 entonces debe copiarse ese valor en la variable "**haber**", en la variable "**debe**" deberá colocar **cero** y en la variable "**estado**" debe colocar el caracter '**H**' que indicará que la cuenta está habilitada.

Si por el contrario en la variable "**saldoInicial**" se colocó un valor negativo, deberá copiarlo en la variable "**debe**" como un valor positivo, en la variable "**haber**" debe colocar cero y en y en la variable "**estado**" debe colocar el caracter '**I**' que indicará que la cuenta está inhabilitada.

Una vez que se haya efectuado ese proceso, el programa permitirá ingresar una serie de transacciones a esa cuenta, se podrán realizar depósitos (D) o retiros (R). Para esto se ingresarán dos datos por iteración: un carácter indicando el tipo de transacción y el monto de la transacción, este último siempre será positivo (p. e.: D 345.67 o R 1009.15). Si en vez de ingresar D o R se ingresa una C indicará que todas las transacciones para esa cuenta habrán terminado, se imprimirá un resumen como se muestra más adelante y se leerán datos para una nueva cuenta; si por el contrario se ingresa una F todo el proceso termina y se imprimirá un resumen final. Junto con la C o la F se ingresa un número que no requiere validar.

Por cada transacción que se realice en una cuenta deberá realizar lo siguiente:

- Si la operación es un depósito deberá verificar qué variable tiene el saldo, si está en la variable "**haber**" deberá sumarle el monto leído y si está en la variable "**debe**" deberá restarlo. En este último caso si el valor se torna negativo deberá moverlo a la variable "**haber**" como positivo y la variable "**debe**" deberá quedar en cero, luego, debe cambiar la variable "**estado**" de 'I' por '**H**'.
- Si la operación es un retiro deberá verificar qué variable tiene el saldo, si está en la variable "**haber**" deberá restárselo y si está en la variable "**debe**" deberá sumarlo. En el primer caso si el valor se torna negativo deberá moverlo a la variable "**debe**" como positivo y la variable "**haber**" deberá quedar en cero, luego debe cambiar la variable "**estado**" de '**H**' por '**I**'.

A continuación, se presentan algunos ejemplos de ejecución:

```
Ingrese el numero de cuenta y el saldo inicial: 23541890   -56387.56
Estado de la cuenta: 23541890
SALDO INICIAL        HABER          DEBE      ESTADO
-56387.56             0.00       56387.56        I

Ingrese una transaccion: D 11111.11
Estado de la cuenta: 23541890
SALDO INICIAL        HABER          DEBE      ESTADO
-56387.56             0.00      45276.45        I

Ingrese una transaccion: D 50000.0
Estado de la cuenta: 23541890
SALDO INICIAL        HABER          DEBE      ESTADO
-56387.56          4723.55           0.00        H

Ingrese una transaccion: R 5000.0
Estado de la cuenta: 23541890
SALDO INICIAL        HABER          DEBE      ESTADO
-56387.56             0.00        276.45        I

Ingrese una transaccion: C 0.0
SALDO FINAL: -276.45 – Cuenta inhabilitada(*)
Cantidad de depósitos: 2     Cantidad de retiros: 1

Ingrese el numero de cuenta y el saldo inicial: 10537728   0.0
Ingres una transaccion: R 300.0
…
Ingrese una transaccion: F 0.0
**FIN DE PROCESO***
Numero de clientes atendidos: 25
```

(*) Si es la variable "haber" la que se queda con el saldo, deberá aparecer positivo y el mensaje será "Cuenta habilitada"

Si usted lo cree conveniente, la pregunta podrá ser elaborada empleando funciones, sin embargo, de hacerlo, éstas deberán ser desarrolladas obligatoriamente en archivos independientes (.h y .c) al archivo main.c, no podrá definir funciones en el archivo main.c.

CRITERIOS DE CALIFICACIÓN EN ESTA PREGUNTA:

1. Si el programa presentado que presenta más de tres errores de sintaxis serán calificados sobre la mitad del puntaje.
2. Si el programa no muestra los resultados o los muestren y no sean correctos, no podrán tener más del 75% de la nota.
3. Se descontará 15% de la nota si el programa define variables con nombres que no tengan sentido. Las variables deben empezar con una minúscula, se emplearán mayúsculas para separar las palabras compuestas (p. e.: baseInf).
4. Se descontará 15% de la nota si no se colocan comentarios relevantes, incluyendo un encabezado al inicio del programa en el que se indique el nombre del autor, la fecha, y una descripción de lo que hace programa).
5. No se calificará el código puesto como comentario.
6. No se calificarán aquellas funciones implementadas en el archivo main.c

San Miguel, 14 de septiembre del 2020