



Apellido y Nombre:

DNI:

PARCIAL:

Calificación :

## TEMA A

### Ejercicio 1

El banco "TuPlataMiPlata", mantiene el estado de las cuentas de sus clientes en el archivo "cuentas.dat", que actualiza todos los días con los movimientos provenientes del archivo de texto "movimientos.txt" de la siguiente manera:

- Si el movimiento es un débito (D): se le debe restar al saldo de la cuenta cliente el importe del movimiento.
- Si el movimiento es un crédito (C): se debe sumar el importe del movimiento al saldo de la cuenta cliente.

Ambos archivos están ordenados por el número de cuenta ascendente.

En el archivo de movimientos pueden venir números de cuenta inválidos que no deben ser procesados, el archivo de cuentas no tiene cuentas inválidas.

Se pide:

- Realizar la actualización del archivo de cuentas con la información de los movimientos.

Desarrollar una función para validar el número de cuenta, ver "Validación del número de cuenta bancaria"

- Todas las cuentas incorrectas se deben grabar en el archivo "error\_nro\_cta.txt", el mismo debe estar ordenado por número de cuenta descendente y debe contener el número de cuenta que figura en el archivo de movimientos, un tabulado, el número verificador correcto.

Validación del número de cuenta bancaria:

- El número de cuenta es un número secuencial de 6 dígitos.
- Contiene un dígito verificador de cuenta que se agrega al final del número de cuenta precedido por una barra.
- El dígito verificador se calcula como: la suma de las cifras hasta que quede un solo dígito de la resta entre dígitos pares y los dígitos impares del número de cuenta.

Ejemplo:

Número secuencial: 637043

$674 - 303 = 371$

$3 + 7 + 1 = 11$

$1 + 1 = 2$

Número de cuenta con el dígito verificador = 637043/2

## Ejercicio 2

Implemente la clase VectorEnteros en C++ tal que el main que se encuentra al pie sea válido. Analice la salida por consola para determinar el funcionamiento de los operadores.

A tener cuenta:

1. Sea cuidadoso con el manejo de la memoria.
2. Aunque su compilador lo permita, no utilice VLE (Variable Length Array).
  - a. Ej: "int vec[x]" donde x es determinado en tiempo de ejecución.
3. Sea cuidadoso de su tiempo y no desarrolle ningún método que no sea necesario para que el main propuesto funcione correctamente.
4. No suponga ningún tamaño máximo para los vectores.
5. Informe cualquier situación anormal con una excepción.

```
int main()
{
    int ve1[]      = {1,2,3,4,5,6,7,8};
    int ve2[]      = {9,10,11,12,13};
    VectorEnteros v1;
    VectorEnteros v2(ve2, sizeof(ve2)/sizeof(ve2[0]));
    VectorEnteros v3;
    cout<<v1.agregar(ve1, sizeof(ve1)/sizeof(ve1[0]))<<endl;
    v3 = v1+v2;
    v3 = v3+14;
    cout<<v3<<endl;
    return 0;
}
```

Salida esperada:

```
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

```
Process returned 0 (0x0)   execution time : 0.106 s
Press any key to continue.
```

## EVALUACIÓN

### NOTA GENERAL

- El parcial se desarrolla con la cámara encendida, durante toda la duración del mismo.
- En caso de ser requerido, debe abrir el micrófono.
- La hora límite de entrega es 17:45 hs.
- Las notas serán enviadas por MIEL
  
- Desarrolle cada ejercicio en un proyecto separado.
- **Incluya en el encabezado de cada archivo, // apellido\_nombre\_DNI**
- Recuerde antes de comprimir, eliminar las carpetas bin y obj de cada proyecto.
- **Entregue ambos proyectos compactados en un zip, “apellido\_nombre\_DNI.zip”.**
- Entregue el parcial usando prácticas de MIEL.
- Enviar a todos los tutores.
- ¡La evaluación es individual!

¡El mayor de los éxitos!