1. **CapitalizeFirst**: Realice una función que reciba un std::string y "capitalice" cada inicio de palabra del mismo. Por ejemplo si el texto original es "este es el primer parcial de introducción al cómputo" el resultado deberá ser: "Este Es El Primer Parcial De Introducción Al Cómputo". El prototipo de la función deberá ser:

```
void CapitalizeFirst(std::string &phrase);
```

2. **Validación de CBU**: La Clave Bancaria Uniforme (CBU) es un código utilizado en Argentina por los bancos para la identificación de las cuentas de sus clientes. Mientras que cada banco identifica las cuentas con números internos, la CBU identifica una cuenta de manera única en todo el sistema financiero argentino. Está compuesta por 22 dígitos, separados en dos bloques (de 8 y 14 dígitos respectivamente). Por ejemplo:

   28505909 40090418135201

   En el primer bloque, los primeros 3 dígitos corresponden al Banco. Luego le sigue el primer "dígito verificador". Los siguientes 3 dígitos identifican la sucursal y por último el segundo dígito verificador.

   El segundo bloque se compone de 13 dígitos para el número de cuenta y el tercer y último dígito verificador.

   En el ejemplo dado sería:

   Banco: **285** (B_0=2, B_1=8, B_2=5) DV_0 = 0

   Sucursal: **590** (S_0=5, S_1=9, S_2=0) DV_1 = 9

   Cuenta: **4009041813520** (C_0=4, C_1=0, C_2=0, …) DV_2 = 1

   El procedimiento para validar una clave CBU es el siguiente:

   - Validación del primer bloque: Se deben sumar los dígitos correspondientes al Banco, DV_0, y Sucursal, con los siguientes pesos: (7, 1, 3, 9, 7, 1, 3). En el ejemplo sería B_0*7 + B_1*1 + B_2*3 + DV_0*9 + S_0*7 + S_1*1 + S_2*3=81. El último dígito de esta suma se resta de 10 (En el ejemplo: DIF_1=10-1=9) y este valor debe coincidir con el DV_1 (En el ej. DV_1=9=DIF_1).
   - Validación del segundo bloque: Sumar los dígitos del número de cuenta, con los siguientes pesos: (3, 9, 7, 1, 3, 9, 7, 1, 3, 9, 7, 1, 3). En el ejemplo sería (C_0*3 + C_1*9 + C_2*7 + C_3*1 + C_4*3 + C_5*9 + C_6*7 + C_7*1 + C_8*3 + C_9*9 + C_{10}*7 + C_{11}*1 + C_{12}*3=139). El último dígito de esta suma se resta de 10 (En el ejemplo: DIF_2=10-9=1) y este valor debe coincidir con el DV_2 (En el ej. DV_2=1=DIF_2).

   Implemente una función que reciba una CBU en forma de string nativo, y lo verifique. Debe retornar verdadero o falso. Respete el siguiente prototipo:

```
int cbu_check(const char cbu[]);
```

   Implemente un programa que solicite al usuario un número de CBU, lo verifique e imprima por pantalla si el número es válido o no. Implemente la función de forma que resulte sencillo cambiar los pesos que se le da a cada dígito.

3. **Excepciones**: Haga un bloque **`try`**/**`catch`** y en su cuerpo llame a una función **`funA`**, a su vez **`funA`** llama a **`funB`**. Declare tanto en **`funA`** como en **`funB`** instancias de un UDT que anuncien su construcción y destrucción. Haga finalmente que **`funB`** haga el **`throw`** de una excepción. Que puede observar con respecto a las instancias creadas? Coloque múltiples catch para tipos

distintos y verifique el correcto funcionamiento cuando dispara excepciones de distinto tipo desde **`funA`**. Que pasa si dispara una exception de un tipo no "**`catch`**eado"? Verifique el comportamiento del **`catch(…)`**.

4. Compruebe que una clase con constructores privados no puede (por ahora) instanciarse.