

Práctica 1

Calculadora

Fecha de entrega: 1 de diciembre

1. Descripción de la práctica

La práctica consiste en la implementación de una calculadora en línea de órdenes cuyo comportamiento emule al de las típicas calculadoras disponibles en los teléfonos móviles o al de la calculadora de Windows (al usarla sólo con teclado). El funcionamiento básico de la calculadora es el siguiente: Nada más iniciarse, el programa queda a la espera de que el usuario introduzca una expresión numérica compuesta por una sucesión de operandos y operadores de la forma:

operando1 operador1 operando2 operador2 operando3 operador3 operando3 ...[⌘]

Los *operandos* y *operadores* se introducirán seguidos sin dejar espacios. La secuencia debe empezar y terminar por un operando, siempre debe tener un operador entre cada dos operandos, y termina con un salto de línea ([⌘]).

Los operandos serán números reales (escritos en formato decimal). El programa calculará el valor de la expresión, asociando por la izquierda y sin contar la precedencia (como ocurre en las típicas calculadoras) y mostrará el resultado.

Si ha habido algún error, ya sea por haber introducido un operador no soportado, por haber escrito mal un operando o por poner operandos no válidos (como negativos para las raíces) se mostrará un mensaje de error en lugar del resultado.

Tras mostrar el resultado (o el mensaje de error) el programa quedará a la espera de que el usuario introduzca otra expresión. Este proceso se repetirá mientras el usuario no pulse la tecla **x** (seguida de un salto de línea), lo cual producirá la terminación del programa.

El programa se implementará de forma incremental en sucesivas versiones (un total de 4 versiones más las extensiones opcionales).

1.1. Versión 1.0

En la primera versión sólo se considerarán los operadores binarios de suma, resta, multiplicación y división, representados respectivamente por los caracteres +, -, * y /. En la Figura 1 se muestran algunos ejemplos de ejecución de la versión 1.0. Se muestran tres cálculos (líneas 1, 3 y 5). Se entiende que se ha pulsado *Intro* en al final de esas líneas.



```
2.5*4
10
2.5*4+5/2-10
-2.5
2*2*2*2-0.45
15.55
0_
```

Figura 1

Nótese que el resultado de evaluar una expresión, tras ser mostrado se pierde, y por tanto no se puede utilizar como primer operando de la siguiente expresión (por la restricción de que una expresión debe siempre empezar por un operando). Este caso se considerará en la versión 3.0.

En esta versión se espera que todas las expresiones sean sintácticamente correctas y el programa terminará cuando se introduzca un 0 como primer operando de la expresión.

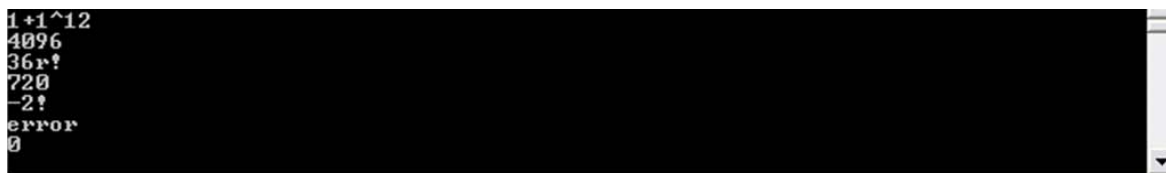
1.2. Versión 2.0

Para la segunda versión se deberá extender la calculadora con los siguientes operadores:

- Raíz cuadrada: Se ejecutará con el carácter *r* y se comprobará que el operando es un número positivo. En caso de no serlo se mostrará un mensaje de error y se cancelará la operación en curso.
- Exponencial ("x elevado a y"): Se ejecutará con el carácter *^* y se comprobará que el exponente es un número entero. En caso de no serlo se mostrará un mensaje de error y se cancelará la operación en curso. *Nota: No se permite usar la función pow() de la biblioteca cmath.*
- Factorial: Se ejecutará con el carácter *!* y se comprobará que el operando es un número entero no negativo. En caso de no serlo se mostrará un mensaje de error y se cancelará la operación en curso.

Los operadores de raíz cuadrada y factorial son unarios y se escribirán en notación posfija. El programa interrumpirá el cálculo actual cuando detecte un error, de operando, como en los tres casos anteriores, o sintáctico (operador no soportado, aritmética incorrecta).

En la Figura 2 se muestran algunos ejemplos de ejecución de la versión 2.0. Se muestran tres operaciones: 2 elevado a 12, factorial de la raíz cuadrada de 36, y factorial de -2 (que da error).



```
1+1^12
4096
36r!
720
-2!
error
0
```

Figura 2

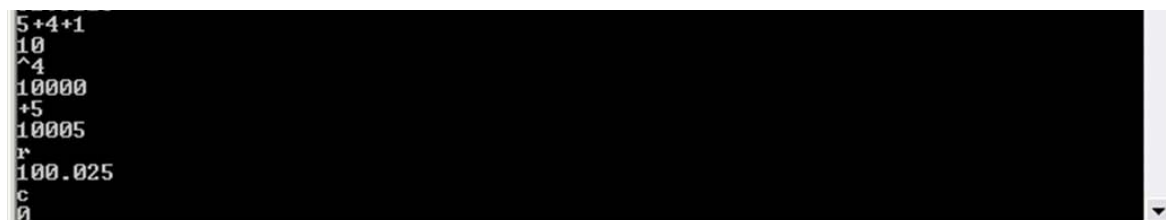
1.3. Versión 3.0

Como se ha explicado anteriormente, la calculadora de momento no permite utilizar el resultado obtenido en una expresión como primer operando de la siguiente expresión. En esta tercera versión se deberá considerar este caso. Por tanto, aparte del tipo de expresiones ya consideradas, se deberá implementar el soporte para que la calculadora entienda expresiones que empiecen por un operador, entendiéndose en tal caso que el primer operando es el resultado del cálculo anterior (siendo cero si nos encontramos en el primer cálculo).

Para ello se recomienda utilizar la función `peek()` sobre `cin` (`c = cin.peek()`), la cual permite observar el siguiente carácter del flujo de entrada sin ser realmente leído.

La versión 3.0 además permitirá cancelar el cálculo actual al pulsar la tecla `c` (la `C` de las típicas calculadoras), que simplemente pondrá el resultado a cero y terminará el cálculo actual. Similarmente, para salir, el usuario introducirá la letra `x` (en lugar de un cero) para terminar la ejecución. Esto permite usar ceros como operandos.

En la Figura 3 se muestran ejemplos de ejecución en los que se reutilizan resultados anteriores. El usuario ha escrito las líneas 1, 3, 5, 7 y 9. El resto de líneas muestran los resultados.



```
5+4+1
10
^4
10000
+5
10005
r
100.025
c
0
```

Figura 3

1.4. Versión 4.0

Finalmente, la versión 4.0 permitirá configurar las teclas que la calculadora utiliza para cada operación (la operación de cancelación **c** y la de salida **x**) para cambiarlas por otros caracteres. La calculadora ahora comenzará mostrando un menú principal con tres opciones:

- 1.- Calculadora
- 2.- Configuración
- 3.- Terminar

Al pulsar 1 se accede a la calculadora propiamente dicha. Pulsando 2 se procede al establecimiento de las teclas para cada operación. Para ello se irá solicitando la tecla para cada operación, dando siempre la oportunidad al usuario de pulsar directamente *Intro* si se quiere dejar la tecla actual. La opción 3 sale del programa. Nótese que ahora, si estando en el modo calculadora se pulsa la tecla de salir (por defecto **x**) se saldrá al menú principal de la calculadora.

2. Parte opcional

Opcionalmente se deberá implementar en la calculadora el soporte necesario para que un resultado intermedio pueda ser guardado y ser recuperado después en otros cálculos.

Para ello, el usuario pulsaría la tecla **m** durante el cálculo de una expresión o bien al mostrar el resultado, y el valor actual se guardaría en una variable. Después, al introducir una expresión, el usuario tendría la posibilidad de pulsar la tecla **n** en lugar de un operando y se reutilizaría el último valor guardado. Obsérvese que sólo se permite almacenar un resultado en memoria.

También se propone implementar el soporte necesario para que la calculadora escriba en un archivo (log) todas las operaciones realizadas en la última sesión (formateado adecuadamente para ser fácilmente leído). El nombre del archivo será **log.txt**.

3. Entrega de la práctica

La práctica se entregará a través del Campus Virtual. Se habilitará una nueva actividad **Entrega de la Práctica 1** que permitirá subir un archivo (en este caso, `main.cpp`) con el contenido de la versión 4.0 de la práctica. Es suficiente con que uno de los dos integrantes del grupo suba el archivo.

Antes de ser entregada, la práctica deberá ser probada exhaustivamente. Es recomendable haber revisado con el profesor el funcionamiento de las versiones intermedias.

Fecha límite de entrega: **1 de diciembre a las 23:55.**