

Examen de febrero – 1 de febrero de 2012

Tiempo disponible: 2 horas y 30 minutos

Se pide construir un programa (**febrero.cpp**) que muestre repetidas veces el siguiente menú de opciones y ejecute la opción seleccionada hasta que el usuario escoja la 0:

- 1 – Cálculos de e elevado a x
- 2 – Detecta capicúa
- 0 - Salir

Opción 1: 4 puntos Opción 2: 4 puntos
Programa principal: 2 puntos

El programa deberá encargarse de validar la opción introducida por el usuario, de forma que reiteradamente solicite una opción mientras que la introducida no sea una de las tres válidas.

Una vez que el usuario haya elegido una opción válida, si es la 0 terminará la ejecución del programa, y si es alguna de las otras realizará el correspondiente procesamiento, atendiendo a las siguientes indicaciones:

1. Si el usuario elige la **opción 1**, se le pedirá que introduzca por teclado una secuencia de números enteros positivos (cada uno en una línea) y finalizada en uno negativo (centinela). Para cada número entero positivo x que se lea, se calculará y mostrará a continuación en la pantalla el valor de la potencia e^x . Para realizar dicho cálculo se utilizará la siguiente aproximación:

$$e^x = \sum_{i=0}^n \frac{x^i}{i!} \quad [1]$$

La primera vez que se use esta opción, el cálculo se realizará con los cinco primeros términos ($n = 5$), esto es:

$$e^x = \sum_{i=0}^4 \frac{x^i}{i!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$$

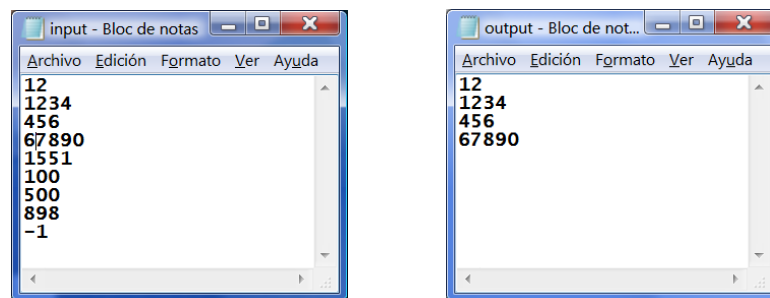
La segunda vez que se use la opción deberán calcularse con dos términos más ($n = 7$) y así sucesivamente. *¡Cada vez que se use esta opción, mayor precisión!*

Requisitos de abstracción procedimental de la opción 1: debes implementar un subprograma que calcule y devuelva el factorial de un número dado (siendo el número ≥ 0), y otro que, a partir de un entero positivo x dado y un número de términos n , devuelva e^x calculado según [1] con el número de términos indicado.

- Si el usuario elige la **opción 2**, se procesará un archivo de enteros positivos `input.txt` (cada número en una línea y acabado en el centinela -1) (¿existe?) y se generará un archivo `output.txt` con todos los números de `input.txt` anteriores al primer capicúa que haya en dicho archivo (ten en cuenta que podría no haber ningún número capicúa en `input.txt`, en cuyo caso `output.txt` debería contener todos los números de `input.txt` a excepción del centinela).

Recuerda que un número es capicúa si es igual a su imagen especular (por ejemplo, 121, 1551 y 67876 son números capicúas porque al calcular su imagen especular, o inverso, resulta el mismo número).

Por ejemplo, para un archivo `input.txt` como el que aparece a continuación en primer lugar, el archivo `output.txt` quedaría como se muestra:



Requisitos de abstracción procedimental de la opción 2: debes implementar un procedimiento que, dado un número entero positivo, devuelva `true` si es capicúa y `false` si no lo es.

Cuida el estilo y depura a medida que vayas construyendo el programa (no esperes a terminarlo todo; prueba cada parte que vayas completando).

No olvides poner un comentario al principio con tu nombre y nº de grupo. Deberás entregar el archivo **febrero.cpp** con el código del programa a través del Campus Virtual.