

Prueba Práctica Competitiva: Tribonacci

Se trata de elaborar un programa ejecutable, “tribo”, que calcule términos arbitrariamente grandes de la sucesión de Tribonacci, definida como:

$$tribo_0 = 0$$

$$tribo_1 = 0$$

$$tribo_2 = 1$$

$$tribo_n = tribo_{n-1} + tribo_{n-2} + tribo_{n-3}, n \geq 3$$

Varios ejemplos de ejecución:

```
$ ./tribo 0
0
$ ./tribo 10
81
$ ./tribo 100
53324762928098149064722658
$ ./tribo 1000
815507705949063215012634973737520390101047421640059634182536354
954214345249517992411833219017896606412691713984091121798255747
368520490502996057925004321483423983646944214344896256767238653
269823324951340326550513652712428004750634815007544492510783789
625725711384
```

El programa debe admitir como argumento al ejecutarse desde la línea de comando un entero indicando el término de la sucesión que se desea calcular y debe escribir en la salida estándar la representación decimal del término correspondiente de la sucesión de Tribonacci. El código fuente del programa debe estar escrito en OCaml, manteniéndose en lo posible dentro del paradigma funcional. No está permitido el uso de bucles (for o while) ni variables. Para la aritmética de precisión arbitraria se recomienda usar el módulo `Big_int_Z` de la librería `Zarith`. Esta librería no suele venir incluida con la instalación por defecto de OCaml. Parte del trabajo a realizar consiste en conseguir instalar esta librería en su equipo¹

¹La librería `Zarith` requiere tener instalada en el equipo la librería `GMP` de aritmética con

Esta práctica es competitiva, lo que significa que sólo unos pocos alumnos pueden puntuar con ella. Es importante mantener la discreción durante todo el proceso. Las implementaciones que se aprecien duplicadas serán descalificadas. Para ser admitido, el programa debe ser capaz de calcular con exactitud todos los términos de la sucesión de Tribonacci que se le pidan. Se valorará especialmente la velocidad de ejecución². Se recomienda utilizar un algoritmo especialmente eficiente para el cálculo de los términos de la sucesión. Tenga en cuenta que la precisión debe ser absoluta.

Para acelerar el cálculo se puede adaptar el cálculo matricial de fibonacci³ a esta sucesión. Partiendo de la igualdad:

$$\begin{pmatrix} fib_n \\ fib_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

y calculando la potencia de la matriz basandose en la igualdad:

$$x^n = \begin{cases} (x^2)^{n/2} & \text{si } n \text{ es par} \\ x * (x^2)^{n/2} & \text{si } n \text{ es impar} \end{cases}$$

Esta prueba práctica es de carácter individual. Sólo se admitirán las primeras 20 implementaciones correctas recibidas antes de la fecha límite de entrega. De estas, serán seleccionadas para su evaluación las 10 más eficientes. El código debe venir debidamente comentado para su fácil comprensión. Para la calificación se tendrá en cuenta, además de la eficiencia, la sencillez, claridad y originalidad del código. La puntuación máxima para esta prueba es de 0,8 puntos de los 2 que se pueden obtener en el apartado de “Pruebas Prácticas”. Si se detecta algún plagio, los implicados en el mismo perderán la oportunidad de recibir ninguna puntuación en la sección de Pruebas Prácticas.

Normas y fecha límite de entrega

La entrega se realizará a través de la tarea en Moodle creada específicamente para este fin. Se entregará sólo un archivo de nombre “tribo.ml” con el código fuente del programa. La tarea permanecerá abierta hasta las 20:00 del 21 de noviembre, o hasta que se supere el número de 20 propuestas correctamente recibidas (lo que antes ocurra). Diariamente se actualizará el número de propuestas recibidas.

precisión arbitraria. Para un uso más cómodo, es conveniente haber instalado previamente el gestor de librerías de OCaml Findlib (ocamlfind). Si instaló el compilador de OCaml usando opam, también debería usar opam para instalar los paquetes ocamlfind y zarith.

²Se espera, por ejemplo, que sea capaz de calcular, en un ordenador personal estándar actual, el término 2.000.000 de la sucesión en menos de 1 segundo.

³https://es.wikipedia.org/wiki/Sucesión_de_Fibonacci

La entrega requiere que se pulse el botón de “Enviar tarea”, y solo se podrá realizar un envío. Es decir, una vez realizado el envío, la tarea ya no podrá editarse (no se podrá eliminar, ni se podrán realizar cambios en la misma).