

Paradigmas de Programación

“Números combinatorios”

Prueba Práctica Competitiva

Cuando utilizamos para representar números enteros (de \mathbf{Z}) un tipo de dato como *int*, que solo permite representar enteros de un tamaño determinado (y usa aritmética modular) hemos de tener especial cuidado.

El siguiente fragmento de código OCaml¹ implementa una función *comb*: $\text{int} * \text{int} \rightarrow \text{int}$ para representar números combinatorios²: se pretende que $\text{comb}(m, n) = \binom{m}{n}$.

```
let rec fact n =
  if n > 0 then n * fact (n-1)
  else 1;;
let comb m n =
  fact m / fact (m-n) / fact n;;
```

Sin embargo, para valores relativamente pequeños de m y n , *comb m n* da valores que no se corresponden con el valor en \mathbf{Z} de $\binom{m}{n}$, a pesar de que ese valor es mucho menor que mayor entero (*max_int*) que se puede representar en *int*. Así por ejemplo, en mi máquina, en la que los valores int son de 63 bits, $\text{comb}(22, 11) = -784$ (cuando $\binom{22}{11} = 705432$), y la evaluación de *comb* (66,2) provoca un error de ejecución.³

Con este enunciado se acompaña el código fuente del módulo *Comb* y, como ejemplo de uso, un pequeño programa ejecutable que envía a la salida estándar las primeras filas del *Triángulo de Pascal*⁴. El archivo Makefile está configurado para la generación de este ejecutable, con nombre “pascal”, al que se le debe pasar como argumento el número de la última fila a mostrar.

Modifique la implementación del módulo Comb para evitar este problema, de modo que siempre que, $\binom{m}{n} \leq \text{max_int}$, *comb* (m, n) dé “el valor correcto”.

Esta implementación debe ser independiente del tamaño del tipo *int* (debe ser válida, por ejemplo, tanto para plataformas de 32 como de 64 bits).

Intente optimizar esa implementación para que sea lo más eficiente posible en cuanto a uso de memoria y tiempo de ejecución.

¹ El código mostrado corresponde a la implementación del módulo Comb que se adjunta a este enunciado.

² [Números combinatorios en Wikipedia](#)

³ Tenga en cuenta que relaciones que se cumplirían siempre en \mathbf{Z} , como por ejemplo $x * y / y = x$, para todo $y \neq 0$, no tienen por qué cumplirse en int (por ejemplo, en mi máquina $2305843009213693 * 3000 / 3000 = -768614336404565$)

⁴ [El Triángulo de Pascal en Wikipedia](#)

Para esta prueba no debe abandonarse el paradigma funcional; es decir, no se deben emplear bucles ni variables. El código debe venir acompañado de las explicaciones necesarias, incorporadas como comentarios en el mismo archivo *comb.ml*.

De entre todas las propuestas realizadas correctamente y entregadas en tiempo y forma, se seleccionarán las 15 más eficientes, de entre las que consigan dar “el valor correcto” siempre (o casi siempre⁵) que sea posible. Después de inspeccionar el código de estas propuestas, serán calificadas con un máximo de 0,6 puntos (para contar dentro de la sección “Pruebas Prácticas” de la evaluación de la asignatura). Para esta calificación se tendrá en cuenta la sencillez, claridad y originalidad del código. Este ejercicio puede realizarse individualmente o en grupos de hasta 3 personas. Cuando uno de los ejercicios valorados haya sido realizado por un grupo de 2 personas, a cada miembro del grupo se le asignará un 60% de la nota del ejercicio. Si el ejercicio ha sido realizado por un grupo de 3 personas, cada miembro del grupo recibirá un 40% de la nota del ejercicio. El único archivo que puede modificarse es el archivo *comb.ml*. Con la versión modificada de este archivo deben seguir compilando sin problema el resto de los archivos suministrados. La primera línea del archivo *comb.ml* debe contener, como comentario, el nombre completo de todos sus autores.

Normas y fecha límite de entrega

La entrega se realizará en el gitlab de la facultad (<https://git.fic.udc.es>). Dentro de él, se creará un nuevo proyecto con nombre *Comb*. Todos los profesores de la asignatura deben ser añadidos con rol Master, al igual que todos los miembros del grupo, en el caso de que el ejercicio no sea resuelto de manera individual. El proyecto debe contener únicamente el archivo *comb.ml*. Una vez terminado, debe enviarse aviso de esto mediante correo electrónico a las direcciones jorge.grana@udc.es y jose.molinelli@udc.es. La entrega debe realizarse antes de las 24:00 horas del día 21 de diciembre.

⁵ Si no hubiese suficientes propuestas que den el valor correcto siempre que sea posible, se seleccionarán también aquellas que lo den en más casos.