

# Project Euler Task 8

Козиний Сергей

3 ноября 2016 г.

## Problem 8

“

The four adjacent digits in the 1000-digit number that have the greatest product are  $9 \times 9 \times 8 \times 9 = 5832$ .

73167176531330624919225119674426574742355349194934  
96983520312774506326239578318016984801869478851843  
85861560789112949495459501737958331952853208805511  
12540698747158523863050715693290963295227443043557  
66896648950445244523161731856403098711121722383113  
62229893423380308135336276614282806444486645238749  
30358907296290491560440772390713810515859307960866  
70172427121883998797908792274921901699720888093776  
65727333001053367881220235421809751254540594752243  
52584907711670556013604839586446706324415722155397  
53697817977846174064955149290862569321978468622482  
83972241375657056057490261407972968652414535100474  
82166370484403199890008895243450658541227588666881  
16427171479924442928230863465674813919123162824586  
17866458359124566529476545682848912883142607690042  
2421902267105562632111109370544217506941658960408  
07198403850962455444362981230987879927244284909188  
84580156166097919133875499200524063689912560717606  
05886116467109405077541002256983155200055935729725  
71636269561882670428252483600823257530420752963450

Find the thirteen adjacent digits in the 1000-digit number that have the greatest product. What is the value of this product?

”

Для того, чтобы найти максимальное произведение среди всех произведений рядом стоящих цифр в данной последовательности, для начала сгенерируем все подобные произведения:

— *Haskell lang*

```
import qualified Data.List as List
```

```
import qualified Data.Char as Char
```

```
numString = "7316717653133062491922511967442657474235534919493496983520312  
.....7745063262395783180169848018694788518438586156078911294949545  
.....9501737958331952853208805511125406987471585238630507156932909  
.....6329522744304355766896648950445244523161731856403098711121722  
.....3831136222989342338030813533627661428280644448664523874930358  
.....9072962904915604407723907138105158593079608667017242712188399  
.....8797908792274921901699720888093776657273330010533678812202354  
.....2180975125454059475224352584907711670556013604839586446706324  
.....4157221553975369781797784617406495514929086256932197846862248
```

```

.....2839722413756570560574902614079729686524145351004748216637048
.....4403199890008895243450658541227588666881164271714799244429282
.....3086346567481391912316282458617866458359124566529476545682848
.....9128831426076900422421902267105562632111110937054421750694165
.....8960408071984038509624554443629812309878799272442849091888458
.....0156166097919133875499200524063689912560717606058861164671094
.....0507754100225698315520005593572972571636269561882670428252483
.....600823257530420752963450 "
numDigit = map Char.digitToInt numString
windowMuls = List.unfoldr (\lst-> case lst of
    (x:xs)-> Just (foldl (*) 1 (take 13 lst), drop 1 lst)
    []-> Nothing) numDigit

```

Здесь: *numString* — строковое представление данного числа; *numDigit* — представление данного числа в виде последовательности цифр; *windowMuls* — подсчитанные произведения по скользящему окну шириной в 13 позиций. Вычисление *windowMuls* производится при помощи функции *unfoldr*. Процедура

*unfoldr* :: (*b* -> *Maybe*(*a*,*b*))-> *b* -> [*a*] (здесь и далее " :: " будет означать принадлежность к типу) по заданному зерну *seed* :: *b* воссоздает последовательность элементов *seq* :: *a* с помощью функции *func* :: *b* -> *Maybe*(*a*,*b*), где на каждом шаге, правый элемент вычисляемой пары, обернутой в монаду *Maybe* включается в результирующий список а левый - служит аргументом для *func* на следующем шаге итерации. Если на некотором шаге *Maybe*(*a*,*b*) принимает значение *Nothing*, список заканчивается и вычисление обрывается.

В итоге:

```

task8 :: Int
task8 = maximum windowMuls

```

Результат:

```

#Console
austrotaxus@small-box:~/EulerProj$ stack ghci
Using main module: Package 'EulerProj' component exe:EulerProj-exe
with main-is file: /home/austrotaxus/EulerProj/app/Main.hs
The following GHC options are incompatible with GHCi and have not
been passed to it: -threaded
Configuring GHCi with the following packages: EulerProj
GHCi, version 7.10.3: http://www.haskell.org/ghc/  :? for help
[1 of 2] Compiling Lib
( /home/austrotaxus/EulerProj/src/Lib.hs, interpreted )
[2 of 2] Compiling Main
( /home/austrotaxus/EulerProj/app/Main.hs, interpreted )
Ok, modules loaded: Lib, Main.
*Main Lib> task8
23514624000

```

Проект, содержащий это и другие решения можно найти по адресу:

<https://github.com/Austrotaxus/EulerProj/>