

課題 4

- 1、データ領域に 32 ビットのデータ 2 つを使って 64 ビットの 2 つの変数 x と y を以下のように宣言する。下位のアドレスのデータが 32 ビットであるとする。

```
.data
.align 4
x: .long 0x00124011,0x3130FFFF
y: .long 0x07001245,0x12f01348
.text
    ここからメインプログラム....
```

この 64 ビットの変数を 符号なしの 64 ビットの値として乗算し、その結果の 128 ビットの値 (32 ビット \times 4) を下位から eax, ebx, ecx, edx にセットして終了するプログラムを作りなさい。

ヒント： 4 回の符号なし乗算で行うことができる。
32 ビット符号なし乗算命令 `mul` は、`edx` と `eax` を使わなくてはならない。
64 ビット同士を足し合わせるときにはキャリーをわすれないこと。
計算に必要なデータ領域は `.data` のところで、`.long` で確保すること。

- 2、再帰呼び出しを使って、フィボナッチ数 $fib(10)$ を計算し、 eax に格納して終了するプログラムを作りなさい。

フィボナッチ数 $fib(n)$ とは、負でない数 n に対して、
 n が 0 もしくは 1 のときは、1
それ以外の場合は、 $fib(n) = fib(n-1) + fib(n-2)$
で定義される数である。(課題 2 と同じ)

ヒント： レジスタで、引数を渡してもよい。
必要な値は `push` 命令で待避する。