```
;LFSR random generators
;
;with cc65 compiler/assembler package
;compile: ca65 -t c64 random.s && ld65 -t c64 -o random random.o c64.lib
;
;© 2007 Hanno Behrens (pebbles@schattenlauf.de)
;LGPL Licence
;
;Call random and you get a lowbyte in A and an highbyte in Y
;you should only use the 8 bit lowbyte cause thats best random.
;The full 16 A/Y may be used but suffer of some bad disadvantages
;that should be best avoided by not using Y.
;The lowbyte of the one algo shifts right, the other left,
;so the result is mixed up a bit better than just with plain shifts
;The periode of this two combined generators is nearly 2^31 (exact
2^16-1*2^15-1)
;Make shure you don't feed either seeds just with plain zero cause
;the zero remains zero. All other seeds will do.
;
;DO NOT USE THIS ALGORITHMS FOR CRYPTOGRAPHY, THEY ARE WEAK
;USE FORTUNA, YARROW OR OTHER STRONG (AND MUCH SLOWER THAN THIS) INSTEAD
;
;LFSR do produce on bit feedback periods as followed:
;0+1 = 3
;1+2 = 7
;2+3 = 15
;2+4 = 31
;4+5 = 63
;5+6 = 127
;1+2+3+7 = 255
;4+8 = 511
;6+9 = 1023
;8+10 = 2047
;1+9+10+11 = 4095
;0+10+11+12 = 8191
;1+11+12+13 = 16383
;13+14 = 32767
;10+12+13+15 = 65535

        .setcpu "6502X"
        .macpack generic
        .macpack cbm

        .segment "STARTUP"

        .org $9000
        .word *
        .org *-2

;combination of generator 65535 and 32767 has periode of 2147385345
;result in a (lo) and y (hi)
```

```
.proc rand
        jsr rand64k       ;Factors of 65535: 3 5 17 257
        jsr rand32k       ;Factors of 32767: 7 31 151 are independent and
can be combined
        lda sr1+1         ;can be left out
        eor sr2+1         ;if you dont use
        tay               ;y as suggested
        lda sr1           ;mix up lowbytes of SR1
        eor sr2           ;and SR2 to combine both
        rts
.endproc

;periode with 65535
;10+12+13+15
.proc rand64k
        lda sr1+1
        asl
        asl
        eor sr1+1
        asl
        eor sr1+1
        asl
        asl
        eor sr1+1
        asl
        rol sr1         ;shift this left, "random" bit comes from low
        rol sr1+1
        rts
.endproc

;periode with 32767
;13+14
.proc rand32k
        lda sr2+1
        asl
        eor sr2+1
        asl
        asl
        ror sr2         ;shift this right, random bit comes from high -
nicer when eor with sr1
        rol sr2+1
        rts
.endproc

;feel free to set seeds as wished, if put in zeropage some speed-boost is
;the result. For example sr1=$5c sr2=5e would fit
sr1:    .word $a55a
sr2:    .word $7653
```