ML Primes

rickmk.com/rmk/Com/primes.html

ML PRIMES

By Rick Kephart

This program shows how to have the computer print out all the prime numbers from 1 to 10,000. It was written using only the C-128's built-in ML monitor.

First, we prepare an area of memory with some non-zero value(s). This routine will fill the area from $1400 to $3AFF (5120 to 15103), which is 9983 bytes (that's enough because the highest prime will be 9973), with non-zero values:

```
1300  A9 00    LDA #$00
1302  85 FD    STA $FD;     Use zero-page addressing at $FD-$FE
1304  A8       TAY
1305  A9 14    LDA #$14
1307  85 FE    STA $FE;    Start at $1400

1309  91 FD    STA ($FD),Y
130B  C8       INY
130C  D0 FB    BNE $1309

130E  E6 FE    INC $FE
1310  A5 FE    LDA $FE
1312  C9 3B    CMP #$3B;    Stop when $3B00 is reached
1314  D0 F3    BNE $1309
```

Now here is where the sieve is applied. Multiples of all numbers from 2 to 255 are eliminated. Since the square root of 10,000 is 100, it really isn't necessary to go any higher than that, but to simplify programming (at the cost of about one second of running time) this program goes all the way to there, and checks all numbers and not just primes.

```
1316  A9 02    LDA #$02;    Start eliminations with "2"
1318  85 FC    STA $FC;     $FC holds the current number being eliminated

131A  A5 FC    LDA $FC;     We're not going to eliminate that number, but
131C  18       CLC;         rather every multiple of it, so we start by
131D  65 FC    ADC $FC;     doubling it
131F  85 FD    STA $FD;     And then store it in the zero-page pointer
1321  A9 14    LDA #$14;    The high byte will be $14
1323  69 00    ADC #$00;    Unless the doubling of $FC caused a carry
1325  85 FE    STA $FE

1327  A9 00    LDA #$00
1329  91 FD    STA ($FD),Y; Put a "0" in that space
```

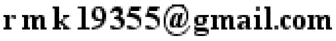
```
132B  18       CLC;         Add the value in $FC to the pointer
132C  A5 FD    LDA $FD
132E  65 FC    ADC $FC
1330  85 FD    STA $FD
1332  A5 FE    LDA $FE
1334  69 00    ADC #$00
1336  85 FE    STA $FE

1338  C9 3B    CMP #$3B;    Go as high as $3AFF
133A  D0 EB    BNE $1327

133C  E6 FC    INC $FC;     All multiples from 2 to 255
133E  D0 DA    BNE $131A
```

Now, all non-prime numbers have been eliminated, and they are ready to be printed out.

```
1340  A9 0D    LDA #$0D;    First a carriage-return
1342  20 D2 FF JSR $FFD2

1345  A9 00    LDA #$00;    $FD and $FE will be used to count in Dcimal
mode
1347  85 FD    STA $FD
1349  85 FE    STA $FE

134B  A9 14    LDA #$14;    A dynamic pointer will be used at $1367
134D  8D 68 13 STA $1368
1350  A9 01    LDA #$01;    No need to start at 0
1352  8D 67 13 STA $1367

1355  78       SEI;         Use DEC mode to count so the number need not
be
1356  F8       SED;         converted from binary to decimal before
printing
1357  18       CLC
1358  A5 FD    LDA $FD
135A  69 01    ADC #$01
135C  85 FD    STA $FD
135E  A5 FE    LDA $FE
1360  69 00    ADC #$00
1362  85 FE    STA $FE
1364  D8       CLD
1365  58       CLI

1366  AD 00 3B LDA $1401;   Get the byte from memory and see if it's
prime
1369  F0 1E    BEQ $1389;   Not prime if it's zero

136B  A2 01    LDX #$01;    Get two bytes of decimal number
```

```
136D  A9 20    LDA #$20;    Print a space before it to separate it from
the
136F  20 D2 FF JSR $FFD2;   Previous number printed

1372  B5 FD    LDA $FD,X;   Print the first part of the number
1374  48       PHA
1375  4A       LSR
1376  4A       LSR
1377  4A       LSR
1378  4A       LSR
1379  09 30    ORA #$30
137B  20 D2 FF JSR $FFD2

137E  68       PLA;         Print the second part of the number
137F  29 0F    AND #$0F
1381  09 30    ORA #$30
1383  20 D2 FF JSR $FFD2

1386  CA       DEX;         Get the other byte
1387  F0 E9    BEQ $1372

1389  18       CLC;         Move up the dynamic pointer
138A  AD 67 13 LDA $1367
138D  69 01    ADC #$01
138F  8D 67 13 STA $1367
1392  AD 68 13 LDA $1368
1395  69 00    ADC #$00
1397  8D 68 13 STA $1368

139A  C9 3B    CMP #$3B;    Only go as high as $3AFF
139C  D0 B7    BNE $1355

139E  60       RTS;         End
```

You can write to me at .


rmk19355@gmail.com

Predicting Prime Numbers: Finding Prime Numbers