

```

JOB
PRIMES SIMPLE TEST
STORE 10/50 BLOCKS
COMPUTING 1000 INSTRUCTIONS
OUTPUT
0 ANY 100 LINES
COMPILER ABL

```

```

| This is a demonstration program for the Ferranti Atlas 1 emulator
| called The Sieve of Eratosthenes, it generates and lists all the
| prime numbers up to 1000.

```

```

2)   H 1000
1)   0101   1   0   A2
     1362   0   0   a0/9   | set mask of bits
     0121   3   0   2       | b3 = 2
3)   1362   0   0   a0/4   | unset every b3th bit
     0124   3   0   1       | b3++
     0121  127   0   a3     | repeat until b3 == b1
     [end of main calculation]

```

R2

```

[print header]
0101   2   0   A90/0
1067   2   0   A90/0+0.4

```

```

[print prime numbers]

```

```

0121  88   0   G102
0121   2   0   0       | b2 is next potential prime
0121  11   0   0
0121   4   0   G101-1
4)   1362   0   0   A0/5   | inspect every half-word in the map
     0124  11   0   0.4
     0172  11  16   0
     0227  127   0   A4
     [Terminate last line]
     1065   0   0   2.1
     1117   0   0   0

```

R4

T

Knock out every b3 bits

```

0113  90   0   A99+0.4
0121  10   3   0
1302  10  10   0       | b10 = b3 ** 2
0170  10   1   0
0227  127   0   A0/2   | if b10 > b1 then print results & stop run
0121  20   3   0
1314  20   0   24       | b20 = b3/24 , b97 = remainder
0124  20  20   0
0124  20  20   0       | shift up for 1/2 word addressing
0101  11  20  G100     | b11 contains b3th bit
0163  97   0   0
0107  11  97  G201
0214  127  11  A99     | if (the b97th bit of b11 == 0) return

1)   0121  11  10   0
     1314  11   0  24   | b11 = b10/24 , b97 = remainder
     0124  11  11   0
     0124  11  11   0   | shift up for 1/2 word addressing
     0163  97   0   0
     0101  14  97  G200   | b14 = mask
     0117  14  11  G100   | apply mask to the b11th half-word

     0172  10   1   0

```

```

0124 10 3 0 | b10 += b3
0227 127 0 A1 | if (b10 < b1) repeat else return

99) 0121 127 0 0 | return

```

R5

```

T
Inspect every half-word in the map
0113 90 0 a99+0.4
0101 12 11 G100 | b12 = b11th half-word in the map
0215 127 12 a3
0124 2 0 24 | if (b12 == 0) ( B2 += 24 ; return)
0121 127 0 A99
3) 0121 3 0 23
1) 0216 127 12 A2 | test topmost bit
0121 81 2 0
1362 0 0 A2/L1
[end of line control]
0203 127 4 A2
1065 0 0 2.1
0121 4 0 G101-1
2) 0124 2 0 1 | b2++
0124 12 12 0 | b12 << 1
0203 127 3 A1 | do 24 times
99) 0121 127 0 0 | return

```

R9

```

T
Setup mask
[ Set up a mask of B1 bits starting at the 1st bit
(counting from 0) of A100/0]
0113 90 0 A99+0.4
0121 10 1 1
0121 11 0 J77777777
0121 16 0 0 | half-word mask index
1) 0113 11 16 G100 | next mask half-word
0122 10 0 24 | B10 -= 24
0124 16 0 0.4 | B16 += 0.4
1216 127 10 a1 | round again?
| now remove the surplus trailing bits
0121 11 0 11.4 | b11 = 11.4
98) 0170 10 0 0
0224 127 0 a97 | if (b10 == 0) -> A97
0101 13 11 G200 | mask 1 bit missing
0117 13 16 G100-0.4 | last half-word &= mask
0122 11 0 0.4 | b11 -= 0.4
0124 10 0 1 | b10++
0121 127 0 a98
97) 0121 11 0 J17777777
0117 11 0 G100 | unset first 2 bits
99) 0121 127 0 0 | return
Z

```

90) CT01

The Sieve of Eratosthenes.

```

G200=2053:154.4 | array of half-words with 23 bits set
G201=2053:167.0 | array of half-words with 1 bit set
G100=8: | data area
G101 = 14 | max. number of primes to be printed on a line
G102 = 6:0.0 | format of number to be printed

```

EA1

***Z