

## Exercise 2.2.11:

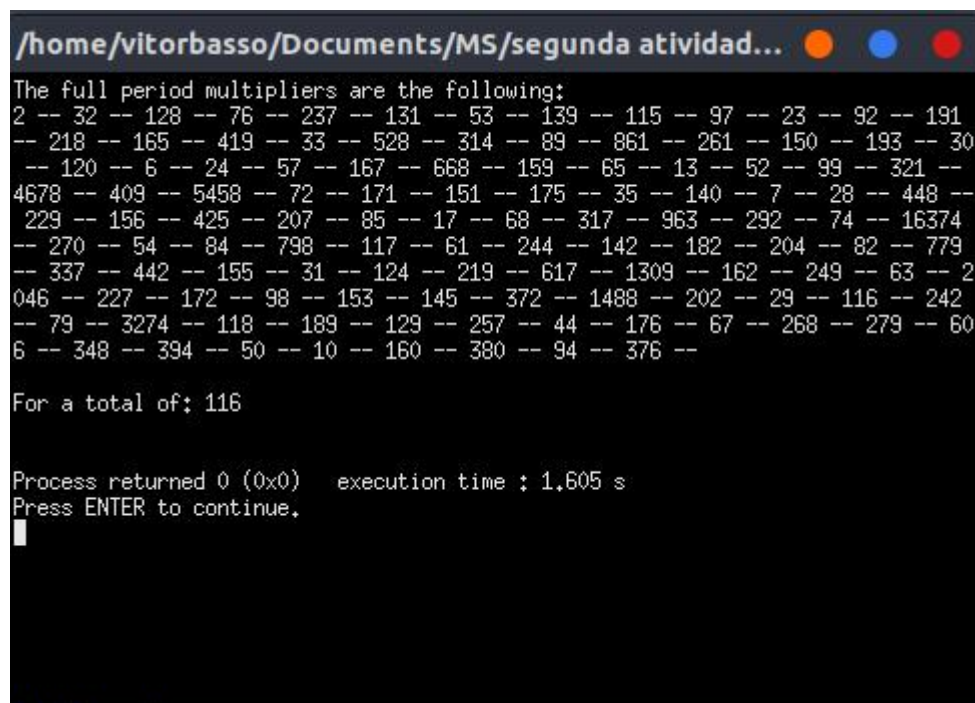
Let  $m$  be the largest prime modulus less than or equal to  $2^{15} - 1$  (see Exercise 2.1.6).

(a) Compute all the corresponding modulus-compatible full-period multipliers.

(b) Comment on how this result relates to random number generation on systems that support 16-bit integer arithmetic only.

A)

Os full period multipliers que também são module compatible para esse caso são os seguintes:



```
/home/vitorbasso/Documents/MS/segunda atividade...
The full period multipliers are the following:
2 -- 32 -- 128 -- 76 -- 237 -- 131 -- 53 -- 139 -- 115 -- 97 -- 23 -- 92 -- 191
-- 218 -- 165 -- 419 -- 33 -- 528 -- 314 -- 89 -- 861 -- 261 -- 150 -- 193 -- 30
-- 120 -- 6 -- 24 -- 57 -- 167 -- 668 -- 159 -- 65 -- 13 -- 52 -- 99 -- 321 --
4678 -- 409 -- 5458 -- 72 -- 171 -- 151 -- 175 -- 35 -- 140 -- 7 -- 28 -- 448 --
229 -- 156 -- 425 -- 207 -- 85 -- 17 -- 68 -- 317 -- 963 -- 292 -- 74 -- 16374
-- 270 -- 54 -- 84 -- 798 -- 117 -- 61 -- 244 -- 142 -- 182 -- 204 -- 82 -- 779
-- 337 -- 442 -- 155 -- 31 -- 124 -- 219 -- 617 -- 1309 -- 162 -- 249 -- 63 -- 2
046 -- 227 -- 172 -- 98 -- 153 -- 145 -- 372 -- 1488 -- 202 -- 29 -- 116 -- 242
-- 79 -- 3274 -- 118 -- 189 -- 129 -- 257 -- 44 -- 176 -- 67 -- 268 -- 279 -- 60
6 -- 348 -- 394 -- 50 -- 10 -- 160 -- 380 -- 94 -- 376 --

For a total of: 116

Process returned 0 (0x0)   execution time : 1.605 s
Press ENTER to continue.
```

B)

Esse resultado possibilita o uso da fórmula  $x = a * (x \% q)$ , o que garante um melhor desempenho a sistemas de 16-bit integer, pois evita o overflow realizando a multiplicação do multiplicador com um número menor ( $x \bmod q$  em vez de  $a * x$ ). Portanto, é possível calcular uma faixa maior de números aleatórios utilizando dessa fórmula no lugar de  $x_{i+1} = (x_i * a) \bmod m$ , especialmente em sistemas que suportam apenas aritméticas de 16-bit integer.