

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação I
AD1 1º semestre de 2012.

GABARITO

1. Questão Única

Escreva um programa que dado um número natural qualquer permita testá-lo quanto a sua primalidade e decompô-lo nos seus fatores primos. A sua implementação deve aceitar inteiros longos e não ficar presa caso o número seja muito grande. O seu programa deve imprimir o tempo de execução numa outra janela, usando um componente adequado para exibição. Algumas sugestões e requerimentos da implementação:

- Neste local há uma boa discussão de algoritmos para verificação de primalidade: http://en.wikipedia.org/wiki/Primality_test
- Use o caracter ^ para representar exponenciação: $61782633 = (3^2)(11)(624067)$
- Supondo que o seu computador atinge 1 gigaflop (10^9 operações de ponto flutuante por segundo), faça uma estimativa do tempo gasto para detectar que $2^{61} - 1$ (2305843009213693951) é primo e compare-a com o tempo do seu algoritmo.

1. Solução

Conforme descrito na Wikipedia ^{1 2}, um número primo é um número natural que possui exatamente dois divisores naturais distintos: o número 1 e ele mesmo.

O teste mais simples para verificar a primalidade de um dado número inteiro n , consiste em dividir n por 2, e por cada inteiro ímpar m , onde $2 < m \leq \lfloor \sqrt{n} \rfloor$. Se o resultado de qualquer uma dessas divisões for um número inteiro, então n não é um número primo, caso contrário, é primo.

Admitindo que o computador executa 10^9 divisões por segundo (1 gigaflop), destacam-se duas maneiras de estimar o tempo que o computador necessitará para verificar se o número ($2^{61} - 1 = 2305843009213693951$) é primo:

a) usando o limite superior de $m = n$, tem-se:

$$\frac{2305843009213693951}{2} \cdot \frac{1}{10^9} = 1152921504,6 \text{ segundos}$$

Considerando que 1 ano tem 31536000 segundos, seriam necessários mais de 36 anos para verificar a primalidade do número desejado.

$$\frac{1152921504,6}{31536000} = 36,6 \text{ anos}$$

¹http://en.wikipedia.org/wiki/Prime_number

²http://en.wikipedia.org/wiki/Primality_test

- b) usando o limite superior de $m = \lfloor \sqrt{n} \rfloor$ é possível verificar a primalidade do mesmo número em 0,759 segundos.

$$\frac{\sqrt{2305843009213693951}}{2 \cdot 10^9} = \frac{1518500249}{2 \cdot 10^9} = 0,759 \text{ segundos}$$

Exemplos de Uso:

A Figura 1, mostra a interface do sistema proposto ao verificar a primalidade de um número. O tempo de execução que o sistema utilizou para verificar se o número era primo é apresentado em uma outra janela, fazendo uso do comando:

```
Showmessage ( FormatDateTime ('ss:zzz', hora2-hora1) );
```

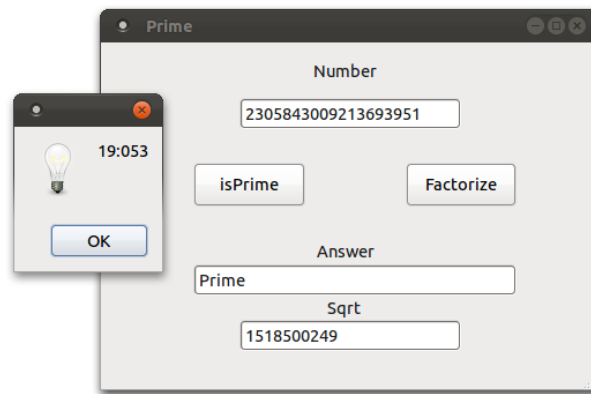


Figura 1: Verificando se o número 2305843009213693951 é primo.

Na Figura 2, a interface mostra a decomposição do número desejado no TEdit *Answer*.

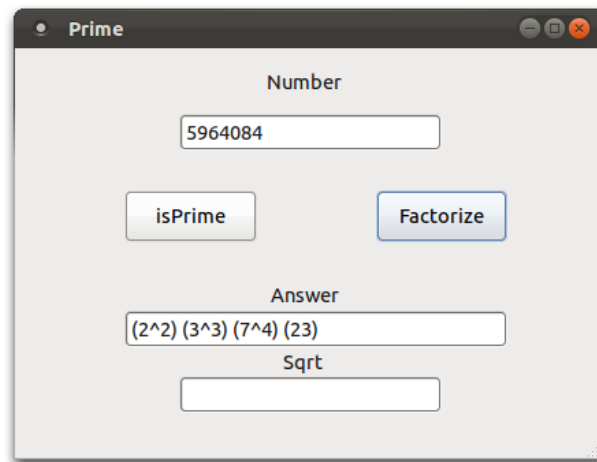


Figura 2: Fatorando o número 5964084.

Para a implementação ser capaz de tratar um número desta magnitude, é necessário utilizar um computador com arquitetura de 64 bits e definir um tipo `BigInteger` para todos os inteiros do programa: `type BigInteger = Int64;`