Matemática Discreta

lista extra 02

Nota: o objetivo dessa lista de exercícios é praticar o uso do comando `while`.

1. **Números de Fibonacci**

   Os dois primeiros números de Fibonacci são

   $$f_0 = 0 \qquad \text{e} \qquad f_1 = 1$$

   E daí, os próximos são calculados pela fórmula

   $$f_n = f_{n-1} + f_{n-2}$$

   Por exemplo,

   - $f_2 = f_1 + f_0 = 1$
   - $f_3 = f_2 + f_1 = 3$
   - $f_4 = f_3 + f_2 = 5$

   e por aí vai ...

   Faça um programa que calcula todos os números de Fibonacci menores do que 1000.

2. **O algoritmo de Euclides**

   Em aproximadamente 300AC, Euclides descobriu o seguinte programa de computador para calcular o MDC de dois números a e b

```
Entrada:  a > b

  1. Calcule

       r  <--  resto da divisão de a por b

  2. Se ( r = 0 )

       Imprima (b),  e vá embora ...

  3. Senão, atualize

       a  <--  b
       b  <--  r

     e volte para o passo 1
```

   Implemente esse programa na linguagem C.

3. **Fatoração**

   Faça um programa que encontra a decomposição em fatores primos de um número n fornecido como entrada.

   Por exemplo,

$$24 \quad = \quad 2 \cdot 2 \cdot 2 \cdot 3$$

4. **Conversão de base**

   Faça um programa que converta um número n para a base 2.

   Por exemplo,

   - $3 \quad = \quad (11)_2$
   - $10 \quad = \quad (1010)_2$
   - $27 \quad = \quad (11011)_2$

4. **Decomposição decimal**

   Faça um programa que imprima os dígitos de um número n separadamente.

   Por exemplo,

$$3491 \qquad \Rightarrow \qquad \texttt{3 4 9 1}$$

5. **A regra de divisibilidade do 3**

   Lembre da regra de divisibilidade do 3:

   - Somar todos os dígitos do número,
     Se o resultado for divisível por 3, então o número é divisível por 3
     Se não for, então não é.

   Faça um programa que aplica essa regra para decidir se um número n é divisível por 3 ou não.

6. **A regra de divisibilidade do 13**

   Agora, considere a seguinte regra de divisibilidade do 13

   - Multiplique o último dígito por 4
     E some o resultado
     ao número formado pelos outros dígitos.
     Repita essa operação até obter um número menor do que 100.
     Se esse número for divisível por 3, então o número original é divisível por 13
     Se não for, então não é.

   Faça um programa que aplica essa regra para decidir se um número n é divisível por 13 ou não.

7. **Lançamento de moedas**

Se você lançar uma moeda 5 vezes, dificilmente você vai obter 5 `Caras`.

Mas, se você continuar lançando a moeda de novo e de novo, mais cedo ou mais tarde vai conseguir obter 5 `Caras` consecutivas.

Quanto tempo leva isso?

Bom, nós podemos fazer um programa para descobrir isso.

Utilize a instrução

```
resultado = rand() % 2
```

para simular o lançamento de uma moeda.

E escreva um programa que lança a moeda repetidamente até que ocorram 5 `Caras` consecutivas.

No final, o seu programa deve informar quantas vezes a moeda foi lançada.

Nota: acrescente as linhas

```
time_t s;
srand((unsigned) time(&s));
```

no início do seu programa, e declare a biblioteca `time.h`.