

Programação I

Mais Exercícios (ficha 7)

v1.0

1. Implemente a função `simplifica(n,d)` que dados o numerador e denominador, imprime a fração simplificada. Esta fração pode ser calculada dividindo o numerador e denominador pelo máximo divisor comum dos dos números (utilize a função `mdc(a,b)` implementada anteriormente. Por exemplo, ao invocar `simplifica(4,8)`, a função imprime `n=1, d=2`; ao invocar `simplifica(1536,78360)`, a função imprime `n=64, d=3265`.
2. Implemente um programa que solicita dois números fracionários (`n1, d1, n2 e d2`) e apresenta a sua soma (fração simplificada). Relembra-se que para somar duas frações, basta somar o numerador quando as frações têm o mesmo denominador ou fazer a multiplicação cruzada dos denominadores (e respetivos numeradores) quando os denominadores são diferentes. Utilize a função `simplifica(n,d)` para simplificar o resultado.
3. Implemente a função `revFatorial(n)` que, dado um número inteiro `n` indica qual o número cujo fatorial é esse número (caso não exista deve devolver zero). Por exemplo, `revFatorial(120)` devolve 5 e `revFactorial(100)` devolve 0. Utilize divisões sucessivas por 2, 3, ... até chegar ao número 1; $120/2=60$, $60/3=20$, $20/4=5$, $5/5=1$.
4. Implemente uma versão recursiva de `revFatorial(n)`.
5. Imagine que tem um comando (de botão único) de uma porta de garagem que funciona da seguinte forma:
 - se a porta está aberta ou fechada, faz com que ela se mova até completar o ciclo de abertura ou fecho
 - se a porta está a abrir ou fechar, faz com que ela pare; clicando novamente a porta irá na direção oposta até completar ou clicar novamente no botão.Desta forma, a porta pode estar num de 5 estados: **fechada**, **aberta**, **afechar**, **aabrir** e **parada** e as acções possíveis são: **clicar**, **completo**, **fim**
 - (a) Implemente a função `novoEstado(estado,comando)` que dado o estado da porta e um comando, devolve o novo estado.
 - (b) Implemente um programa que vai solicitando ao utilizador as acções e indicando qual o estado seguinte até que seja introduzido **fim**. Assuma que o estado inicial é **fechado**.
6. Escreva um programa que pede ao utilizador um número inteiro `h` e imprime um triângulo de números com `n` linhas reproduzindo o formato abaixo. Utilize uma variável para construir a cadeia de caracteres.

```
1
1 2
1 2 3
....
1 2 3 ... n
```

7. Implemente um programa que pede ao utilizador um número inteiro **h** e desenha um pinheiro com altura **h**. por exemplo se **h=3** o programa deve imprimir

```
  *
 ***
*****
  *
  *
```

se **h=4** o resultado é

```
  *
 ***
*****
*****
  *
  *
```