

Programação I

Ainda mais funções (ficha 9)

1. Elabore um programa que, dado um inteiro n imprime as potências da base 2, de 2^0 até 2^n . Utilize a função `potencia2(int n)` que devolve a n -ésima potência de 2.

```
Qual potencia de 2? 2
2**0=1
2**1=2
2**2=4
```

2. Os primeiros dois valores da sequência de Fibonacci são 1 (`fib(1) = fib(2) = 1`). A partir do 3º, os valores correspondem à soma dos dois anteriores. Implemente um programa que imprime os primeiros n números da sequência de Fibonacci. Como auxiliar, implemente a função `fibonacci(num)` que devolve o número fibonacci de `num`.

```
Qual o numero? 8
1 1 2 3 5 8 13 21
```

3. Implemente a função `fibonacci_menor(v)` que devolve o maior número de fibonacci menor que v . Por exemplo, `fibonacci_menor(50)` deve devolver 34.
4. Implemente a função `e_primo(n)` que indica se o número n é primo ou não.
5. Utilizando a função implementada no exercício anterior, implemente uma nova função `mostra_primos(n)` que imprime os números primos entre 2 e n .
6. Utilizando ainda a função `e_primo(n)`, implemente uma nova função `conta_primos(n)` que conta o número de números primos entre 2 e n .
7. Implemente a função recursiva `potencia(x,y)` que calcula x^y (com x e y inteiros).
8. Implemente a função recursiva `log2(x)` que calcula o logaritmo base 2 do seu argumento (inteiro). (com x inteiro). Por exemplo `log2(1024)=10`, `log2(11)=3` e `log2(1)=0`.
9. Implemente a função recursiva `adicao(x,y)` que devolve o resultado da soma dos dois argumentos (com $x \geq 0$ e $y \geq 0$ inteiros) sem usar o operador `+`. Em sua substituição utilize a função `maisUm(n)` que, dado n devolve $n + 1$.