

Folha prática 9

Introdução à Programação, DCC/FCUP, 2009/2010

Notas. (i) Todas as funções que escrever devem ser funções puras. (ii) $a|b$ significa “a divide b” (em Python: `b%a==0`). (iii) \rightarrow significa “retorna”. (iv) Propõe-se a seguinte ordem de resolução: 1, 3, 2, 5, 4, 6.

1. *Dos segundos aos dias...*

Escreva uma função `converte(n)`, onde n é um inteiro não negativo, que converte n segundos para um tuplo da forma (d,h,m,s) cujos elementos são os números de dias, horas, minutos e segundos, respectivamente. Por exemplo,

```
converte(3680)  ->  (0,1,1,20)
```

2. *Um modo de encriptar textos...*

Escreva uma função `encripta(s,n)`, onde s é uma “string” constituída apenas por letras minúsculas (não acentuadas) e espaços e n é um inteiro; a função retorna a “string” que resulta de mudar cada letra de s para a letra que está n posições mais à frente. Os espaços ficam inalterados. Por exemplo,

```
encripta("a paz",2)  ->  "c rcb"
```

Notas. A seguir à letra ‘z’ considera-se que está a letra ‘a’. A função que escrever deve funcionar também para valores negativos de n , por forma a facilitar a descriptação,

se $\boxed{\text{encripta}(s,n) \rightarrow t}$, então $\boxed{\text{descripta}(t,-n) \rightarrow s}$

Teste este tipo de utilização de `encripta`. Lembra-se que os códigos das letras são consecutivos. Recorde-se das funções `ord` e `chr`.

3. *Decomposição em factores primos...*

Sabe-se que qualquer inteiro n pode ser decomposto de uma forma única (à parte a ordenação dos factores) num produto de primos

$$n = p_1^{e_1} \times p_2^{e_2} \times \dots \times p_k^{e_k}$$

Escreva uma função `decomp(n)` que retorna a “string” que expressa essa decomposição, na forma que se exemplifica

```
decomp(7)      -> "7"
decomp(24)     -> "2^3 * 3"
decomp(2200)   -> "2^3 * 5^2 * 11"
```

Pode utilizar a função `str`; exemplo `str(25) -> "25"`.

Repare que: (i) os primos ocorrem por ordem crescente, (ii) não se coloca o expoente quando ele é 1, (iii) o sinal `*` só ocorre entre os factores.

Sugestão. Use o seguinte método

```

p=2          # Primeiro inteiro tentado como divisor
sinal=False  # 'True' quando é para colocar '*'
r=""         # string a construir
enquanto n>1:
    e=0
    enquanto p|n:
        n=n/p; e=e+1
    if e>0:    # nao vao ocorrer os inteiros com expoente 0
        r = r + texto(p,e,sinal)
        sinal=True
    p=p+1

```

onde `texto` é uma função apropriada (que deve também escrever).
 Note que, sempre que `e>0`, o correspondente `p` é primo!

4. *Reduzir o contraste de uma imagem...*

Uma “imagem” a preto e branco pode ser representada por uma lista de linhas, sendo cada linha uma lista de inteiros entre 0 (preto) e 255 (branco). Reduzir o contraste de uma imagem é substituir cada inteiro da imagem (excepto os que estão nos lados) pela média dos seguintes 5 inteiros: o próprio inteiro, o inteiro acima, o inteiro abaixo, o inteiro à esquerda e o inteiro à direita.

Escreva uma função `reduz(im)` que calcula a imagem cque resulta de reduzir o contraste a `im`. Admite-se que `im` é quadrada. Por exemplo

[[10, 100, 100],		[[10, 100, 100],
[40, 10, 200],	-->	[40, 90, 200],
[50, 100, 100]]		[50, 100, 100]]

pois $(10+100+100+40+200)/5 = 90$.

Nota. Pode copiar a imagem `im` com

```

from copy import *
imc = deepcopy(im)

```

5. *“Master mind”...*

Escreva uma função

`acertou(s,t)`

onde `s` é o “segredo” e `t` a “tentativa” de um jogo “Master mind”. A função retorna um tuplo `(a,f)` onde `a` e `f` são o número de cores que acertou “no sítio” e “fora”, respectivamente, conforme se exemplifica

```
acertou("1234455","1235544")  ->  (3,4)
      1,2,3 no sítio, 4,4,5,5 fora
acertou("4411111","1133344")  ->  (0,4)
      2 1's e 2 4's fora
acertou("4444444","1144111")  ->  (2,0)
```

Notas. `s` e `t` são “strings” de dígitos que representam cores; têm o mesmo comprimento.

6. *Mais “Master mind”...*

Usando a função que escreveu na questão anterior (`acertou(s,t)`) escreva uma outra função `joga(pinos,cores)` em que

- (a) O computador gera um segredo `s` – string aleatória de comprimento `pinos`, podendo as cores ir de 0 a `cores`.
- (b) O utilizador faz sucessivas tentativas, fornecendo ao computador strings `t`. O computador indica de cada vez o número de cores que acertou no sítio e fora.
- (c) O programa termina quando o utilizador tiver acertado em todas as cores (em número de `pinos`) do segredo.