

**Leia com atenção.** Deve resolver os exercícios no papel, e só no fim, testar as respostas no computador. Deve ser rigoroso nas respostas; por exemplo, se a pergunta for qual é o valor de `[x+1 for x in range(2) for y in range(3)]`, todas as seguintes respostas estão erradas (cotação 0):

```
[1,2,1,2,1,2]      [1 1 1 2 2 2]      1,1,1,2,2,2      [1,1,...,2]
[1,1,1]+[2,2,2]    3*[1]+3*[2]      (1,1,1,2,2,2)
```

(em particular. 2 das respostas erradas são expressões e não valores). Para facilitar a compreensão das funções ou programas que escrever, marque com barras verticais os níveis de indentação. (2) Escreva as funções de forma clara e concisa; se necessário, use funções auxiliares. (5) Não se permita explicitamente a utilização de funções adicionais, as únicas funções que pode utilizar são: `len`, `range` e `append`. Por exemplo, não pode usar as funções `float`, `list` ou `sorted`. Na indexação não pode usar "slices". Por exemplo, se `a` é uma lista, `a[0]`, `len(a)` e `a.append(5)` são permitidos, mas `a[0:2]` e `sorted(a)` estão proibidos.

```
A) m=0
   while m>=-10:
       m=m-2
   print m

B) m=0
   if m>0:
       s=5
       s=2
   print s

C) s="ab"
   for i in range(5):
       s=s+s
   print len(s)
```

- Qual o valor impresso pelo programa A)? Resposta: \_\_\_\_\_
- Qual o valor impresso pelo programa B)? Resposta: \_\_\_\_\_
- Qual o valor impresso pelo programa C)? Resposta: \_\_\_\_\_
- Quais os valores das seguintes expressões?

```
1/3       range(1,5)       1%4 
len("12"+"ahh") 
[a+b for a in range(2) for b in range(2)] 
```

- Escreva uma função `contac(s,v)` que calcula e retorna o número de caracteres da string `s` que existem na string `a`. Por exemplo

```
contac("vaquinha","abvv") -> 3
contac("vaquinha","")     -> 0
```

(as 3 letras 'v', 'a' e 'a' de "vaquinha"). Nota. Pode usar a função `in`.

- Escreva uma função `correcta(b,x)` onde  $b \geq 2$  é uma base de numeração e `x` é uma lista de inteiros que são dígitos nessa base. A função deve retornar `True` se os parâmetros estão nessas condições e `False` caso contrário. Por exemplo

```
correcta(2,[1,0,1,2,1]) -> False      (2 não é dígito da base 2)
correcta(10,[1,[],8])  -> False      ([[] não é inteiro)
correcta(10,[2,0,0,9]) -> True
correcta(1,[9,2,0,9])  -> False      (1 não é base de numeração)
correcta(2.5,[9,2,0,9]) -> False     (2.5 não é base de numeração)
correcta(5,[1,2.5,0,9]) -> False     (2.5 não é dígito na base 5)
correcta(3,[0,2,0,1])  -> False     (primeiro dígito é 0)
```

O primeiro dígito não pode ser 0 (último exemplo), a não ser que seja o único dígito; pode usar a função<sup>1</sup> `type`.

<sup>1</sup>Por exemplo, `type(x)==int` sse `x` é inteiro