

**7.1** Escreva um programa que imprima uma tabela de  $\log(x)$  para os  $x = 1, 2, \dots, 10$ . Utilize o operador % para formatar os logaritmos com 6 casas decimais.

**7.2** Escreva um programa que imprima uma tabela das potências de dois desde  $2^0$  até  $2^{32}$  com o seguinte aspeto. Utilize o operador de formatação % para alinhar as colunas.

```
2^ 0 =          1
2^ 1 =          2
...
2^32 = 4294967296
```

▷ **7.3** Escreva uma função `tempo(t)` que, dado um número total de segundos, retorna uma cadeia formatada em horas, minutos e segundos (HH:MM:SS). Exemplo: `tempo(121)` dá '00:02:01' (0 horas, 2 minutos e 1 segundo).

**7.4** Escreva uma função `media_arit(xs)` cujo resultado é a média aritmética de uma lista, isto é,  $\bar{x} = (x_1 + x_2 + \dots + x_n) / n$ .

**7.5** Escreva uma função `media_geom(xs)` cujo resultado é a média geométrica de uma lista, isto é,  $(x_1 \times x_2 \times \dots \times x_n)^{1/n}$ .

▷ **7.6** Escreva uma função `desvio_padrao(xs)` cujo resultado é o desvio padrão amostral de uma lista de  $n$  valores, isto é,

$$\sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

onde  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$  é a média aritmética dos valores. Pode assumir que  $n > 1$ .

**7.7** Escreva uma função `prod_interno(xs,ys)` cujo resultado é o produto interno de dois vetores representados como listas de comprimento  $n$ , isto é  $x \cdot y = \sum_{i=1}^n x_i \times y_i$ .

**7.8** Escreva uma função `intervalo(xs,a,b)` cujo resultado é a contagem dos valores da lista  $xs$  que estão entre  $a$  e  $b$  inclusivé; pode assumir que  $a \leq b$ .

▷ **7.9** Um *quadrado mágico* é uma matriz de  $n \times n$  de números inteiros tal que todas as linhas, colunas e diagonais somam o mesmo valor. No exemplo seguinte cada linha, coluna e diagonal soma 15:

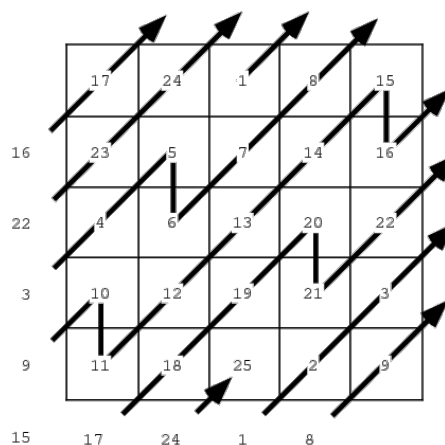
2	7	6
9	5	1
4	3	8

Escreva uma função `magico(A)` que testa se uma matriz (representada como lista de listas) é um quadrado mágico; o resultado deve ser um valor lógico.

## 7.10

O algoritmo seguinte permite gerar quadrados mágicos de tamanho  $n \times n$  para  $n$  ímpar: começemos por colocar um “1” no meio da primeira linha; em seguida, colocamos os números de 2 a  $n^2$ : deslocamos na diagonal para cima e para a direita; quando esse movimento sai fora da matriz, devemos “dar a volta” por baixo e pela esquerda; se encontramos uma entrada já preenchida, movemos para baixo; colocamos o próximo número na posição encontrada e repetimos.

A figura à direita exemplifica os passos da construção dum quadrado mágico  $5 \times 5$ .

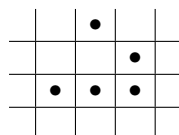


Escreva um procedimento `quadrado(n)` que constroi um quadrado mágico usando este algoritmo; o resultado deve ser o quadrado mágico representado como uma lista de listas.

**7.11** O jogo *Life* foi inventado pelo matemático britânico John H. Conway e é um dos mais conhecidos exemplos de um autómato celular. Considere um tabuleiro bi-dimensional com linhas horizontais e verticais definindo quadrículas; cada quadrícula pode estar vazia ou conter uma célula. Consideram-se *vizinhos* as oito posições directamente adjacentes a uma quadrícula. Dada uma configuração inicial, o jogo desenrola-se em gerações sucessivas; as células morrem ou nascem conforme o número de vizinhos:

1. uma célula com menos de dois vizinhos morre de isolamento;
2. uma célula com mais de três vizinhos morre de sobrepopulação;
3. uma célula com dois ou três vizinhos continua viva;
4. numa quadrícula vazia com exactamente três vizinhos nasce uma nova célula.

Pretende-se escrever um programa para mostrar a evolução das gerações. Vamos representar uma célula pelo carater '0' e uma quadrícula vazia por '.'; o tabuleiro pode então ser representado por uma lista de cadeias de caracteres. Por exemplo, a seguinte configuração de tabuleiro



será representada por `[". . 0 . .", ". . . 0 .", ". 0 0 0 .", ". . . . ."]`.

Para obter mais informação sobre o jogo *Life* pode consultar a página da Wikipédia: [http://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life](http://en.wikipedia.org/wiki/Conway's_Game_of_Life)