MC102 - Algoritmos e Programação de Computadores

Lista de Exercícios 1

1. Responda qual tipo de objeto deve ser usado para armazenar cada uma das seguintes informações:

   a. A idade de uma pessoa.

   b. A área do seu quintal em metros quadrados.

   c. A média da quantidade de chuva no mês de fevereiro.

   d. O número de estrelas na galáxia.

2. Considere o trecho de código abaixo:

```
a = 10
b = a
c = 9
d = c
c = c + 1
```

   Após a execução desse trecho de código, qual será o valor armazenado em cada variável?

3. Faça um programa que leia um número ponto flutuante x e calcule o valor de $f(x) = \sqrt{x} + (x/2) + x^x$. (Dica: $\sqrt{x} = x^{1/2}$).

4. Faça um programa que leia dois valores inteiros nas variáveis x e y e troque o conteúdo das variáveis. Por exemplo, supondo que $x = 2$ e $y = 10$ foram os valores lidos, o seu programa deve fazer com que $x = 10$ e $y = 2$. Refaça este problema usando apenas x e y como variáveis.

5. Faça um programa que leia os valores correspondentes aos três lados a, b e c de um triângulo. O programa então deve determinar se o triângulo é isósceles, escaleno ou equilátero, informando isto para o usuário, e em seguida o programa deve imprimir a área A do triângulo utilizando a fórmula de Heron:

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

   onde

$$s = \frac{a+b+c}{2}.$$

6. Considere um programa que deve classificar um número como par ou ímpar e, além disso, classificar ele como menor do que 100 ou maior ou igual a 100. A solução abaixo faz essa classificação de maneira correta?

```
print("Digite um número:")
a = int(input())
if a % 2 == 0 and a < 100:
  print("O número é par e menor do que 100")
else:
```

```
  if a >= 100:
    print("O número é par e maior ou igual que 100")
if a % 2 != 0 and a < 100:
  print("O número é ímpar e menor do que 100")
else:
  if a >= 100:
    print("O número é ímpar e maior ou igual que 100")
```

7. Faça um programa que leia um caractere 'F' ou 'C', que indica se o próximo valor corresponde à temperatura em Fahrenheit ou Celsius. Em seguida, o programa deve ler o valor da temperatura e então imprimir o valor correspondente à temperatura na outra unidade de medida. Observação: $(C = 5/9 \times (F - 32))$.

8. Faça um programa que leia um ano (valor inteiro) e imprima se ele é bissexto ou não. Observação: um ano é bissexto se ele é múltiplo de 400, ou se ele é múltiplo de 4 mas não é múltiplo de 100.

9. Suponha que uma pessoa possa se aposentar pelo INSS caso atenda alguma das situações abaixo:

   - É do sexo masculino, possui pelo menos 65 anos e pelo menos 10 anos de contribuição.
   - É do sexo masculino, possui pelo menos 63 anos e pelo menos 15 anos de contribuição.
   - É do sexo feminino, possui pelo menos 63 anos e pelo menos 10 anos de contribuição.
   - É do sexo feminino, possui pelo menos 61 anos e pelo menos 15 anos de contribuição.

   Crie um programa que leia um caractere ('M' ou 'F'), que representa o sexo de um indivíduo, e dois inteiros, que representam a idade e o tempo de contribuição. O programa deverá então imprimir "Aposentável" se o indivíduo atenda uma das situações acima. Caso contrário, o programa deverá imprimir "Não Aposentável".

10. Faça um programa que leia dois números e em seguida um caracter que representa um operador aritmético ('+', '−', '∗' ou '/'). Seu programa então deve imprimir o resultado do operador aplicado aos dois números dados.

MC102 - Algoritmos e Programação de Computadores

Lista de Exercícios 2

1. Considere o seguinte menu:

```
1 - Pizza Marguerita
2 - Pizza de Calabresa
3 - Pizza de Pepperoni
4 - Pizza de Mussarela
5 - Sair
```

   O seu programa deve: imprimir o menu; ler um número de 1 até 5; e imprimir a opção do menu correspondente ao número lido. Isso deve ser repetido até que o usuário selecione a opção 5.

2. Faça um programa que leia um número n. Após isso, o seu programa deve ler uma sequência de n números e imprimir uma mensagem indicando se a sequência lida está ordenada de forma crescente ou não.

3. Faça um programa que leia dois números m e n. O seu programa deve imprimir o Máximo Divisor Comum (MDC) dos números m e n. Você deve utilizar a seguinte regra (chamada de Algoritmo de Euclides) para calcular o MDC dos dois números dados:

$$mdc(m, n) = m, \text{ se } n = 0;$$
$$mdc(m, n) = mdc(n, m \,\%\, n), \text{ se } n > 0.$$

4. Escreva um programa que leia um número n. O seu programa deve imprimir o menor número primo que é maior ou igual n e o maior número primo que é menor ou igual a n.

5. O que será impresso pelo programa abaixo? Assuma que o valor de `D` na atribuição inicial de `x` é o valor do último dígito do seu RA.

```
x = 5 + D
y = 0
while True:
  y = (x % 2) + 10 * y
  x = x // 2
  print('x =', x, 'y =', y)
  if x == 0:
    break

while y != 0:
  x = y % 100
  y = y // 10
  print('x =', x, 'y =', y)
```

6. Escreva um programa que leia n números inteiros e imprima quantos deles estão nos seguintes intervalos: $[0, 25]$, $[26, 50]$, $[51, 75]$ e $[76, 100]$. Por exemplo, para $n = 10$ e os seguintes números $\{2, 61, -1, 0, 88, 55, 3, 121, 25, 75\}$, seu programa deve imprimir:

```
[0,25]: 4
[26,50]: 0
[51,75]: 3
[76,100]: 1
```

7. Escreva um programa para computar a raiz quadrada de um número positivo. Use a ideia abaixo, baseada no método de aproximações sucessivas de Newton. O programa deve imprimir o valor da vigésima aproximação.

   Seja Y um número positivo. Sua raiz quadrada é a raiz da função

   $$f(x) = x^2 - Y.$$

   A primeira aproximação é $x_1 = Y/2$. A $(n+1)$-ésima aproximação é

   $$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

8. Aponte os erros de implementação existentes no código abaixo, desenvolvido com o intuito de calcular e imprimir o fatorial de um número inteiro não negativo.

```
valor = int(input("Digite um número: "))
fatorial = n = valor

if n >= 0:
  while n > 0:
    n = n - 1
    fatorial = fatorial * n

  print("O fatorial de", valor, "é igual a:", fatorial)

print("Não existe fatorial de", valor)
```

9. Faça um programa que leia um número inteiro positivo C. O seu programa deve imprimir todas as soluções da equação

   $$x_1 + x_2 + x_3 = C,$$

   onde as variáveis x_1, x_2, e x_3 são inteiras não negativas.

10. Faça uma programa que leia um número na base decimal e imprima esse número na base binária.

11. Faça um programa que leia um número inteiro positivo n e imprima n linhas com o seguinte formato (exemplo para $n = 6$):

```
1
 2
  3
   4
    5
     6
```

12. Faça um programa que leia um número n e imprima n linhas com o seguinte formato (exemplo para $n = 6$):

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

13. Faça um programa que leia um número n e imprima n linhas com o seguinte formato (exemplo para $n = 6$):

```
+ * * * * *
* + * * * *
* * + * * *
* * * + * *
* * * * + *
* * * * * +
```

14. Um jogador da Mega-Sena é supersticioso e só faz jogos em que o primeiro número do jogo é par, o segundo é ímpar, o terceiro é par, o quarto é ímpar, o quinto é par e o sexto é ímpar. Faça um programa que imprima todas as possibilidades de jogos que este jogador supersticioso pode jogar.

MC102 - Algoritmos e Programação de Computadores

Lista de Exercícios 3

1. Escreva um programa que leia n números reais e calcule o desvio padrão destes valores. O desvio padrão é dado pela seguinte equação: $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$, onde n é a quantidade de números, x_i é o i-ésimo valor e $\overline{x}$ é a média dos valores.

2. Mostre o que o programa abaixo irá imprimir caso seja executado (execute o programa a mão, sem usar o interpretador Python).

```
v1 = []
v2 = []
n = 123456789
while n != 0:
  v1.append(n % 10)
  n = n // 10
  v2.append(1)

for a in v1[:-1]:
  print(a, end = ", ")
print(v1[-1])

for a in v2[:-1]:
  print(a, end = ", ")
print(v2[-1])

for i in range(len(v1)):
  for j in range(v1[i]):
    v2[i] = 2 * v2[i]

for a in v2:
  print(a, end = ", ")
print(1)
```

3. Escreva um programa que leia duas sequências de n e m valores inteiros, onde $n \leq m$, e imprima quantas vezes a primeira sequência ocorre na segunda.

   Exemplo:

   ```
   Primeira sequência: 1 0 1
   Segunda sequência: 1 1 0 1 0 1 0 0 1 1 0 1 0
   Resultado: 3
   ```

4. Faça um programa que leia dois conjuntos de números inteiros distintos e imprima a interseção destes dois conjuntos (os números presentes em ambos os conjuntos).

   Exemplo:

   ```
   Primeiro conjunto: 1 2 3 4 5
   Segundo conjunto: 2 5 7 1 9 18
   Resultado: 1 2 5
   ```

5. Faça um programa que leia dois conjuntos de números inteiros distintos e imprima a união destes conjuntos (os números presentes em pelo menos um dos conjuntos).

   Exemplo:

   ```
   Primeira conjunto: 1 2 3 4 5
   Segundo conjunto: 2 5 7 1 9 18
   Resultado: 1 2 3 4 5 7 9 18
   ```

6. Faça um programa que leia duas sequências de números inteiros ordenados e imprima uma sequência com os números ordenados das duas sequências anteriores. Você não deve usar o método `sort`.

   Exemplo:

   ```
   Primeira sequência: 1 3 5 5 7 9 10
   Segunda sequência: 2 2 4 6 8 8 10
   Resultado: 1 2 2 3 4 5 5 6 7 8 8 9 10 10
   ```

7. Faça um programa que calcule o produto interno de dois vetores u e v de mesmo tamanho n. O programa deve ler primeiramente o valor de n e em seguida deve ler duas sequências de mesmo tamanho de números reais e salvar cada sequência em uma lista. O programa deve então calcular e imprimir o produto interno dos vetores lidos.

8. Escreva um programa que leia uma sequência de números inteiros e os salve em uma lista. Em seguida o programa deve ler um outro número inteiro C. O programa deve então encontrar dois números de posições distintas da lista cuja multiplicação seja C e imprimí-los. Caso não existam tais números, o programa deve imprimir uma mensagem correspondente.

   Exemplo:

   ```
   Lista =  [2, 4, 5, 10, 7]
   C = 35
   Resultado: 5 e 7

   Lista =  [2, 4, 5, 10, 7]
   C = 25
   Resultado: não existem tais números
   ```

9. Escreva um programa que leia uma sequência de n números inteiros positivos maiores que 1 e os salve em uma lista `v`.

   O programa deve então imprimir um quadrado de n linhas por n colunas onde em cada posição (i, j), com $i, j \in \{0, \ldots, n-1\}$, deste quadrado deverá ser impresso 1 caso os números `v[i]` e `v[j]` sejam coprimos, e 0 caso contrário.

   Os pares de números `v[i]` e `v[j]` são coprimos se não há nenhum divisor $d > 1$ que seja comum a ambos. Por exemplo 15 e 8 são coprimos, pois os divisores de 8, que são 2, 4 e 8, não são divisores de 15. Abaixo temos um exemplo de execução do programa para $n = 6$ e `v = [2, 3, 4, 5, 6, 7]`.

```
      v[0] v[1] v[2] v[3] v[4]  v[5]
v[0]  0    1    0    1    0     1
v[1]  1    0    1    1    0     1
v[2]  0    1    0    1    0     1
v[3]  1    1    1    0    1     1
v[4]  0    0    0    1    0     1
v[5]  1    1    1    1    1     0
```

   Note no exemplo que `v[0] = 2` é coprimo de `v[1] = 3`, `v[3] = 5` e `v[5] = 7`.

MC102 - Algoritmos e Programação de Computadores

Lista de Exercícios 4

1. Escreva um programa que leia uma string e, em seguida, imprima a inversa da string lida. Exemplo de entrada:

```
Tangamandapio
```

   Impressão esperada:

```
oipadnamagnaT
```

2. Escreva um programa que leia uma string e, em seguida, imprima a string lida removendo **todos** os espaços.

   Exemplo de entrada:

```
Out    of the night that    covers me
```

   Impressão esperada:

```
Outofthenightthatcoversme
```

3. Escreva um programa que leia uma string e imprima a string lida removendo os espaços **extras** entre as palavras, ou seja, entre as palavras deve haver apenas um único espaço.

   Exemplo de entrada:

```
Out    of the night that    covers me
```

   Impressão esperada:

```
Out of the night that covers me
```

4. Faça um programa que leia duas strings e elimine, da segunda string, todas as ocorrências dos caracteres da primeira string. Por fim, seu programa deve imprimir a segunda string.

   Exemplo de entrada:

```
AMOR
MARESIA
```

   Impressão esperada:

```
ESI
```

5. Faça um programa que leia duas palavras e verifique se uma delas é subsequência da outra, ou seja, a primeira pode ser obtida por meio da remoção de letras da segunda. A ordem das letras não pode ser alterada.

Exemplo de entrada:

```
moda
moradia
```

Impressão esperada:

```
moda é uma subsequência de moradia
```

Exemplo de entrada:

```
cereja
cerveja
```

Impressão esperada:

```
cereja é uma subsequência de cerveja
```

Exemplo de entrada:

```
teste
triste
```

Impressão esperada:

```
teste não é uma subsequência de triste
```

6. Escreva um programa que leia duas palavras e determine se a segunda é um *anagrama* da primeira. Uma palavra é um anagrama de outra se todas as letras de uma ocorrem na outra, *em mesmo número, independente da posição.*

   Exemplo de entrada:

   ```
   ROMA
   AMOR
   ```

   Impressão esperada:

   ```
   Anagramas!
   ```

   Exemplo de entrada:

   ```
   regalia
   alegria
   ```

   Impressão esperada:

   ```
   Anagramas!
   ```

   Exemplo de entrada:

   ```
   xzxyxz
   yzxyzx
   ```

   Impressão esperada:

   ```
   Não são anagramas!
   ```

MC102 - Algoritmos e Programação de Computadores

Lista de Exercícios 5

1. Considere o código em Python abaixo:

```
j = 1

def main():
  a = 9

  if a % 2 == 0:
    a = 2
  else:
    a = 3

  print(fun1(2,4))

  for i in range(3):
    for j in range(3):
      print(fun1(a, i+j))

def fun1(a, b):
  p = 1
  for i in range(b):
    p = p * a
  return p+j

main()
```

   (a) Determine quais são as variáveis locais e globais deste programa. Para cada variável local identifique a que função ela pertence.

   (b) Mostre o que será impresso na tela do computador quando for executado este programa.

2. Escreva uma função que receba dois números inteiros positivos a e b como parâmetro e determine se eles são amigos ou não, devolvendo `True` caso sejam amigos e `False` caso contrário.

   Dois números são amigos se cada número é igual à soma dos divisores próprios do outro (os divisores próprios de um número m são os divisores estritamente menores que m). Por exemplo, os divisores próprios de 220 são 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110, cuja soma é 284; e os divisores próprios de 284 são 1, 2, 4, 71 e 142, cuja soma é 220. Logo, 220 e 284 são números amigos.

   A seguinte função deve ser implementada:

```
def amigos(a, b):
```

3. Escreva uma função que receba um valor inteiro positivo n como parâmetro e devolva o menor valor inteiro b, tal que $b^k = n$ para algum inteiro k. Por exemplo, se $n = 27$ então o valor devolvido deve ser 3. Já se $n = 12$, o valor devolvido deve ser 12.

   A seguinte função deve ser implementada:

```
def menor_base_log(n):
```

4. Um inteiro positivo n é **pitagórico** se existem inteiros positivos a e b tais que $a^2 + b^2 = n$. Por exemplo, 13 é pitagórico, pois $2^2 + 3^2 = 13$.

   (a) Escreva uma função que receba como parâmetro três inteiros a, b e n, e devolva `True` caso $a^2 + b^2 = n$ e `False`, caso contrário.

   A seguinte função deve ser implementada:

```
def teste(a, b, n):
```

   (b) Utilize a função do item anterior e escreva uma outra função que receba como parâmetro um inteiro positivo n e verifique se n é pitagórico, devolvendo `True` caso n seja pitagórico e `False`, caso contrário.

   A seguinte função deve ser implementada:

```
def pitagorico(n):
```

5. Escreva uma função que receba uma lista de números reais e devolva a média aritmética dos números da lista.

   A seguinte função deve ser implementada:

```
def media(v):
```

6. Escreva uma função que receba uma lista de números reais e devolva o desvio padrão dos números da lista usando a seguinte fórmula:

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

   onde n é o número de elementos.

   A seguinte função deve ser implementada:

```
def desvioPadrao(v):
```

7. Escreva uma função chamada `sanduiche_primo` que receba como parâmetro um inteiro n e devolva uma tupla com dois valores inteiros $p1$ e $p2$, onde $p1$ é o maior número primo que é menor do que n e $p2$ é o menor número primo que é maior do que n.

   A seguinte função deve ser implementada:

```
def sanduiche_primo(n)
```

8. Escreva uma função que receba como parâmetro uma lista de inteiros. A função deve devolver uma tupla com dois valores inteiros $f1$ e $f2$, onde $f1$ é o elemento da lista com menor frequência (menor número de ocorrências na lista) e $f2$ é o elemento com maior frequência. Dica: use um dicionário para computar as frequências dos elementos da lista.

   A seguinte função deve ser implementada:

```
def frequencias(v)
```

9. Suponha que uma matriz binária `mat` represente ligações entre cidades, onde, se uma posição `mat[i][j]` possui o valor 1, então há uma estrada da cidade `i` para a cidade `j`. Seja o seguinte exemplo de matriz:

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Neste caso, há estradas da cidade 0 para as cidades 1 e 2, e da cidade 2 para a cidade 0. Para cada item abaixo escreva uma função `verifica(mat)` que receba como parâmetro uma matriz quadrada `mat`, indicando as estradas entre as cidades, e devolva uma lista `resposta`.

- Escreva uma função para determinar as cidades com estradas chegando, mas sem estradas saindo da cidade, indicando isto na lista `resposta`, tal que `resposta[i]` recebe `True` caso a cidade `i` satisfaça esta propriedade e `False`, caso contrário.
- Escreva uma função para determinar as cidades com estradas saindo, mas sem estradas chegando na cidade, indicando isto na lista `resposta`, tal que `resposta[i]` recebe `True` caso a cidade `i` satisfaça esta propriedade e `False`, caso contrário.
- Escreva uma função para determinar as cidades isoladas (sem estradas chegando ou saindo da cidade), indicando isto na lista `resposta`, tal que `resposta[i]` recebe `True` caso a cidade `i` satisfaça esta propriedade e `False`, caso contrário.

10. No jogo Sudoku temos uma matriz 9×9 dividida em 9 quadrados de 3×3 preenchidos previamente com alguns números entre 1 e 9 (veja o exemplo à esquerda abaixo). Uma solução para uma instância do jogo consiste no preenchimento de todas as posições vazias com números entre 1 e 9 respeitando-se as seguintes regras:

(a) Não pode haver números repetidos em um mesmo quadrado, ou seja, cada número entre 1 e 9 deve aparecer exatamente uma vez em cada quadrado.

(b) Não pode haver números repetidos em nenhuma linha da matriz.

(c) Não pode haver números repetidos em nenhuma coluna da matriz.

Escreva uma função que receba uma matriz 9×9 como parâmetro, que represente uma proposta de solução para um Sudoku, e teste se a matriz é uma solução válida para o jogo, devolvendo `True` em caso verdadeiro e `False`, caso contrário.

A seguinte função deve ser implementada:

```
def solucao(mat):
```

Veja abaixo um exemplo (à direita) de uma solução para um Sudoku.

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

Sudoku não resolvido

4	2	*6*	5	*7*	1	*3*	9	*8*
8	*5*	*7*	2	*9*	3	*1*	*4*	6
1	3	*9*	*4*	6	*8*	*2*	7	*5*
9	*7*	1	*3*	*8*	*5*	6	*2*	*4*
5	4	*3*	*7*	*2*	*6*	*8*	1	9
6	*8*	2	*1*	*4*	*9*	7	*5*	*3*
7	9	*4*	*6*	3	*2*	*5*	8	*1*
2	*6*	*5*	8	*1*	4	*9*	*3*	7
3	1	*8*	9	*5*	7	*4*	6	*2*

Solução

MC102 - Algoritmos e Programação de Computadores

Lista de Exercícios 6

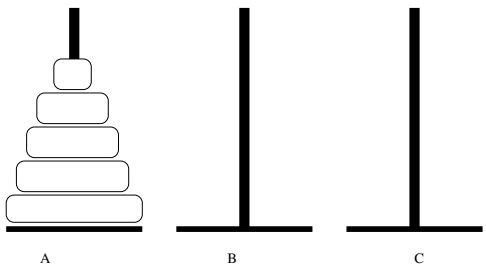
1. Faça uma função recursiva que calcule a média dos elementos de uma lista de números.
2. Mostre uma representação passo a passo da memória do computador, considerando as chamadas da função recursiva para o cálculo dos números da série de Fibonacci (vista em aula).
3. Determine o que a seguinte definição recursiva para uma função f calcula:
   - Se $n = 0$, retorne 0.
   - Se $n > 0$, retorne $n + f(n - 1)$.
4. Escreva uma função recursiva que calcule o somatório dos n primeiros valores da série harmônica. A série harmônica é definida como:

$$\sum_{k=1}^{\infty} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} \cdots$$

5. Escreva uma função recursiva que determine se um número inteiro n é primo.
6. Escreva uma função recursiva que calcule $\lfloor \lg n \rfloor$, ou seja, o *piso* do logaritmo de n na base 2. Por exemplo, $\lfloor \lg 100 \rfloor = 6$.
7. Considere a seguinte variação do problema das Torres de Hanói. O objetivo continua sendo mover os n discos do pino A para o pino C com as restrições: (i) apenas o disco do topo de um pino pode ser movido e (ii) nunca um disco de diâmetro maior pode ficar sobre um disco de diâmetro menor. Agora adicionamos uma nova restrição: (iii) não é possível mover um disco diretamente do pino A para o pino C (ou de C para A). Ou seja, para mover um disco de A para C (ou de C para A), o disco precisa primeiro ser movido para o pino B. Escreva um algoritmo que gere a solução para este novo problema.

   A seguinte função deve ser implementada:

```
hanoi(n, inicial, final, auxiliar)
```



8. Suponha que uma matriz binária quadrada `M` represente a ligação entre um conjunto de n cidades. Desta forma, `M[i][j] = 1` indica que existe uma estrada da cidade `i` para a cidade `j`, e `M[i][j] = 0`, caso contrário. Por exemplo, na matriz abaixo temos que a cidade 0 possui estradas para as cidades 1 e 2, já a cidade 1 possui estrada apenas para a cidade 2. Note que existe uma estrada saindo da cidade 0 em direção à cidade 1, mas não há estrada saindo da cidade 1 em direção à cidade 0, isso porque nem sempre uma estrada que liga duas cidades possui vias de ida e volta.

```
0 1 1 0
0 0 1 0
1 1 0 1
1 0 1 0
```

   Escreva uma função recursiva que, dada uma matriz `M` e uma cidade `i`, determine todas as cidades que podem ser alcançadas a partir de `i`.

9. Escreva funções recursivas para realizar a tarefa de caça-palavras em um diagrama. Dada uma matriz M de caracteres, uma posição $pos = (x, y)$ da matriz e uma palavra p, as funções devem determinar se é possível encontrar a palavra p na matriz M a partir da posição pos e seguindo uma determinada direção (vertical, horizontal ou diagonal). Dica: escreva uma função recursiva para cada uma das possíveis direções. Exemplo:

```
M:
g z h y z l y n s g q j u
i n w a k b n x w w t n y
i w o u i p o f d b o y o
m k o h k l h r k h j w u
j l i o t r t c t b c o p
a h e d n y y y e g t p t
f n o h t y p y t h o n f
r l q o s y y y h n a f x
c e z x t k t m t x e u z
t g m h c v h x c h j n a
s n o w t q o b v n o z j
u n u x x a n p u i g n b
s i x o d j v s f e o n z
p: python
pos: (6,6)
```

```
Resultado:
. . . . . .
. n . . . . n . . . . n .
. . o . . . o . . . o . .
. . . h . . h . . h . . .
. . . . t . t . t . . . .
. . . . . y y y . . . . .
. n o h t y p y t h o n .
. . . . . y y y . . . . .
. . . . t . t . t . . . .
. . . h . . h . . h . . .
. . o . . . o . . . o . .
. n . . . . n . . . . n .
. . . . . .
```

10. Escreva uma função recursiva para calcular $\binom{n}{k}$ sabendo que:

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}, \binom{n}{n} = 1 \text{ e } \binom{n}{1} = n$$