
# Introdução ao Processamento de Dados (IPD)

## Segunda Lista de Exercícios – Algoritmos

1. Faça um programa para ler 50 valores de temperaturas em graus Celsius. Transformar essas temperaturas em Farenheit e imprimir a media das temperaturas em Celsius e Farenheit e quantas temperaturas ficaram acima da media em Farenheit.

2. Leia um vetor de 40 posições e conte quantos elementos pares se encontram no vetor.

3. Leia uma frase e imprima o total de vogais, o total de brancos e o total do resto.

4. Ler um vetor de números inteiros de 30 posições. Depois, ler um número inteiro X, imprimir quantas vezes o número X aparece no vetor.

5. Leia um vetor de 16 posições e troque as 8 primeiras posições pelas 8 últimas posições. Imprima o vetor original e o vetor trocado.

6. Um palíndromo é uma cadeia que pode ser lida de frente para trás e de trás para frente. Ex: ‘SOMOS’ ‘1234321’. Implemente a função palindromo(palavra). O parâmetro palavra é uma string. A função deverá retornar True se for um palíndromo e False caso contrário.

7. Um anagrama (do grego ana = "voltar" ou "repetir" + graphein = "escrever") é uma espécie de jogo de palavras, resultando do rearranjo das letras de uma palavra ou frase para produzir outras palavras, utilizando todas as letras originais exatamente uma vez. Um exemplo conhecido é o nome da personagem Iracema, claro anagrama de América, no romance de José de Alencar. Implemente a função anagrama(frase1,

frase2). Os parâmetros frase1 e frase2 são string. A função deverá retornar True se forem anagramas e False caso contrário. Despreze acentuação (por exemplo: ã = a e ç = c) e os espaços não devem ser computados para efeitos do anagrama.

8. Considere um vetor de trajetórias de 9 elementos onde cada elemento possui o valor do próximo elemento a ser lido.

| Índice | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| Valor | 4 | 6 | 5 | 8 | 1 | 7 | 3 | 0 | 2 |

Fazer um programa que leia esse vetor e imprima a trajetória correta: sequência de impressão 4, 1, 6, 3, 8, 2, 5, 7, 0.

9. Crie uma matriz 5x5 com 1 na diagonal principal e 0 nas outras posições. Imprima a matriz no formato tradicional de linhas e colunas.

10. Crie a função mat_transposta(matriz). A função deve receber uma matriz genérica bidimensional, de qualquer tamanho (não necessariamente quadrada) e retornar a matriz transposta, sem alterar a matriz original.

11. Crie a função mat_maior_10(matriz). A função deve receber uma matriz genérica, de qualquer tamanho (não necessariamente quadrada) e retornar a quantidade de elementos da matriz maiores que dez.

12. Cálculo aproximado de ∏ (Pi) pelo Método de Monte Carlo.
Dada a função

$$x^2 + y^2 <= 4,$$

calcule o valor aproximado de ∏ com $4.10^6$ interações. Dicas: use random.random() * 2 para gerar valores entre 0 e 2 para as variáveis x e y e conte o número de pontos (x,y) que satisfazem a equação dada e divida o resultado por $10^6$. Cuidado com o tipo de dados.

13. Crie a função multiplica_matriz(mat1, mat2) que deve retornar o produto de duas matrizes bidimensionais genéricas, sem alterar as matrizes originais. A função deve imprimir uma mensagem de erro e retornar um vetor vazio ([]) caso não for possível realizar o produto das duas matrizes.

14. Faça um programa que calcule o valor de PI pela soma dos n primeiros termos da série abaixo:

$$\sqrt{12 * (1 - \frac{1}{4} + \frac{1}{9} - \frac{1}{16} + \frac{1}{25} - \frac{1}{36} \ldots}$$

15. Escreva uma função para condensar os elementos de uma lista ordenada L, que contém inteiros repetidos. Por exemplo, para L = [3, 3, 3, 3, 7, 7, 13, 13, 23], a função retorna ['3^4', '7^2', '13^2', '23'] (repare que são strings). Note-se que no caso de um número aparecer uma única vez, não deve haver expoente unitário.

16. Crie um algoritmo que:

a) Leia e escreva o nome e a altura das moças inscritas em um concurso de beleza, até que seja digitada o nome 'MARIA', que marca o final da lista, mas é para ser computada no concurso.

b) Calcule e escreva as duas maiores alturas e quantas moças as possuem.

17. Uma certa firma fez uma pesquisa para saber se as pessoas gostaram ou não de um novo produto lançado no mercado. Para isso, forneceu o sexo do entrevistado e sua resposta (sim ou não). Sabendo-se que foram entrevistadas 2.000 pessoas, crie um algoritmo que calcule e escreva:

a) o número de pessoas que responderam sim;

b) o número de pessoas que responderam não;a porcentagem de pessoas do sexo masculino que responderam não.

18. Crie um algoritmo que leia um número N e verifique se ele é primo. O seu programa deve fazer o MÍNIMO de interações possíveis.

19. Crie um algoritmo que leia um número N e imprima os N primeiros números primos. O seu programa deve fazer o MÍNIMO de interações possíveis.

20. Crie um algoritmo que leia um número N e imprima se ele é perfeito ou não. Um número é perfeito quando a soma dos seus divisores é igual a ele mesmo, e.g., 6 = 3 + 2 + 1.

21. Crie um algoritmo que imprima os 4 primeiros números perfeitos.

22. Crie um algoritmo que leia um número N e calcule:

$$\sum_{i=1}^{N} \frac{1}{i}$$

23. Crie um algoritmo de caixa eletrônico que lê a quantidade de dinheiro a ser sacado e imprime a menor quantidade de notas a ser dada ao usuário. A sua máquina de saque eletrônico é carregada com 1.000 (mil) notas de 100, 50, 20, 10, 5 e 1 cada. Imprimir também a quantidade de cada nota a ser dada ao usuário. Lembre-se que você tem uma quantidade finita de cada tipo de nota. O final da leitura é marcado pelo valor 0 que não deve ser calculado.
Exemplo: 98 = uma nota de 50, duas notas de 20, uma nota de 5, e três notas de 1.
Perceba que a cada saque há uma quantidade menor de cada tipo de notas.
Outra situação é quando não é possível sacar o dinheiro com a quantidade de notas existentes. Neste caso, o programa deve cancelar a operação e informar o usuário que não é possível realizar o saque.

24. Crie um algoritmo se imprima os N primeiros números que sejam primos e façam parte da série de Fibonacci UTILIZANDO OBRIGATORIAMENTE FUNÇÕES.

25. Crie um algoritmo que leia as taxas de juros (***r***) de um financiamento *price* (1% é igual a 0,01), os valores das parcelas (***pmt***) e o número de parcelas (***n***). Em seguida, calcule o valor presente do financiamento que é feito pela seguinte fórmula:

$$\sum_{i=1}^{n} \frac{pmt}{(1+r)^i}$$

26. Crie um algoritmo que leia N números inteiros positivos e responda se é possível formar um polígono com o mesmo (dica: maior número < soma dos demais números).

27. Crie um algoritmo que verifique se existe alguma raiz na equação $Ax^3 + Bx^2 + Cx + D$ no intervalor [-1.000, 1.000]. Seu algoritmo deve ler os coeficientes A, B, C e D. Dica: incremente o valor de $x$ de uma unidade a cada interação e verifique se houve uma mudança de sinal no resultado da equação, se o sinal mudou, existe a ocorrência de uma raiz (não é necessário calcular a raiz).

28. Crie um algoritmo que leia um número (com qualquer número de dígitos) em uma base numérica de ordem < 10 e calcule o número correspondente na base decimal. O número da ordem da base (e.g., 2 para binária, 3 para ternária, 8 para octal, etc.) deve também ser informado pelo usuário.

29. Crie um algoritmo que leia um número decimal (com qualquer número de dígitos) e o converta para a base hexadecimal. A resposta deve ser dada em string.

30. Seu algoritmo deve ler as variáveis inteiras A e B. Posteriormente, calcule o Máximo Divisor Comum (MDC) entre A e B, a quantidade de divisores comuns entre A e B e o Mínimo Múltiplo Comum (MMC) entre A e B.

31. Implemente seguinte algoritmo de criptografia: dada uma frase qualquer, se o código ASCII do caractere for par, subtraia três do código ASCII. Se for ímpar, some três (semelhante ao exemplo dado em sala).

32. Leia um número A real positivo. Calcule sua raiz real utilizando o método babilônico indo até a 10ª interação ($x_{10}$). Considere, por hipótese, $x_0 = 1$.

$$x_{n+1} = \frac{(x_n + \frac{A}{x_n})}{2}$$

33. Jogo de dados. Dois jogadores estão confrontando-se entre si, usando dois dados, numerados de 1 até 6 (dado ←random.randint(1,6)). Vence uma rodada quem obtiver o maior número no lançamento. Entretanto, caso um jogador obtiver um lançamento com dois dados iguais (por exemplo, 1 e 1, 2 e 2,....) e o outro jogador não, o jogador que tiver lançado a dupla ganha. Caso os dois jogadores fizerem o lançamento e o resultado for de duas duplas para os dois jogadores, ganha o jogador com a dupla maior. A disputa é em 11 lançamentos. Indique o jogador vencedor ou se houve empate.

34. Implemente o jogo *Blackjack* ou 21. No caso temos apenas dois jogadores, o cliente (humano) e a banca (computador). O ás pode valer 1 ou 11, de acordo com as cartas que o jogador possui. O valete (J), a dama (Q) e o rei (K) valem 10. A simulação da obtenção das cartas deve ser feita com o uso de números aleatórios

(Dica: carta ← random.randint(1,13), isto é, variando de 1 a 13).

Os números 11, 12 e 13 representam respectivamente J, Q e K. Não há a necessidade de representar um baralho real de 52 cartas. O cliente recebe inicialmente duas cartas aleatoriamente e decide se pede outra ou não, dependendo do somatório. Caso a soma das cartas seja maior que 21, o cliente perde. O cliente pode passar a vez para a banca (que não possui cartas no início do jogo), que, para vencer, terá que obter obrigatoriamente um número maior que o cliente e MENOR do que 21, caso contrário, o cliente ganha. O jogo tem apenas uma rodada. Informe quem ganhou ou se houve empate.

35. Jogo da subtração. Seu algoritmo deverá ler uma variável positiva T e posteriormente mais três variáveis positivas S1, S2 e S3, sendo que estas variáveis são obrigatoriamente menores que T. O jogo consiste de dois jogadores, J1 e J2. A cada rodada a variável T é subtraída por uma das variáveis S1 ou S2 ou S3, escolhida pelo jogador da rodada. Perde o jogo quando o jogador ao executar sua subtração, obtém um valor menor do que zero. O seu programa deve apontar o jogador VENCEDOR.

36. Elabore um programa que calcule a quantidade de dias existentes entre duas datas. Dica: utilize as variáveis D1, M1, A1, D2, M2, A2. Por hipótese, as variáveis dos anos não precisam considerar a correção do calendário gregoriano. Lembre-se que há regras especiais de anos bissextos conforme o ano específico.

37. Faça um programa que preencha uma matriz 3x3 com valores aleatórios e calcule o seu determinante. Uma vez a matriz lida, a parte que faz o cálculo do determinante não se pode ter mais de 4 linhas e não pode haver mais de três células especificadas por linha (em outras palavras, você não pode colocar em uma única linha a fórmula do determinante).

38. Implemente a função Cramer(matriz, vetor) que calcule as raízes de um sistema de 3 equações e 3 incógnitas pela Regra de Cramer e retorna a solução na forma de um vetor. Se não houver solução para o sistema, deve imprimir uma mensagem de erro e retornar um vetor vazio ([]). Sua função pode (e deve) chamar a função det(matriz) que retorna o valor do determinante da matriz. Exemplo

$$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 5 & 5 \\ 6 & 2 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 7 \end{bmatrix}$$

39. Leia um arquivo texto.txt e conte a quantidade de quantas vezes cada palavra ocorre no texto, criando dois vetores distintos ou utilizando a estrutura de dicionários. Você tem ainda que considerar que o texto possui pontuação e números, que não deverão fazer parte da contagem.

40. Crie um vetor com 100 elementos aleatórios. Após isso faça a ordenação dos mesmos, SEM UTILIZAR O MÉTODO sort().

41. Leia um arquivo entrada.txt e troque todas as letras minúsculas por maiúsculas e vice-versa. Grave o resultado em um arquivo saida.txt.

42. Leia um arquivo entrada.txt e conte quantas vezes cada letra do alfabeto (independente de ser maiúscula ou minúscula) aparece. Grave o resultado em um

arquivo saida.txt, com a letra e o número de ocorrências. O arquivo saída terá o formato, por exemplo:

A 13

B 28

Z 1

43. Implemente a função area_triangulo(matriz). O parâmetro matriz é uma matriz 3x2 que contém as coordenadas x e y do pontos do triângulo. Utilize fórmula de Heron para calcular a área.

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

Onde s é o semiperímetro do triângulo e a, b e c são os comprimento dos lados do triângulo.

44. Implemente a função area_triangulo(matriz). O parâmetro matriz é uma matriz que contém as coordenadas x e y dos pontos do triângulo ($P_1$, $P_2$ e $P_3$). Utilize o módulo do produto vetorial para calcular a área do triângulo.

$$\|\vec{u}\ X\ \vec{v}\|$$

Onde $\vec{u}$ é definido por $\overrightarrow{P_1P_2}$ e $\vec{v}$ é definido por $\overrightarrow{P_1P_3}$. Isto resultará no seguinte determinante:

$$\begin{vmatrix} 1 & 1 & 1 \\ x_2 - x_1 & y_2 - y_1 & 0 \\ x_3 - x_1 & y_3 - y_1 & 0 \end{vmatrix}$$

Após o cálculo do determinante, obtenha seu módulo (valor necessariamente positivo) e divida por 2 (compare a complexidade com o exercício anterior).

45. Implemente a função area_poligono(matriz). Esta função calculará a área de um polígono convexo cujos vértices estão na matriz (N x 2 – N indeterminado). Dicas: utilize a função area_triangulo(mat) que calcula a área de um triângulo. Em um quadrilátero com vértices $X_1$, $X_2$, $X_3$ e $X_4$, a área do mesmo consiste da soma dos triângulos com vértices ($X_1$, $X_2$ e $X_3$) e ($X_1$, $X_3$ e $X_4$).

46. Escreva um programa para converter números inteiros, menores do que 4000 e escritos em algarismos arábicos, para romanos. Obs: evite escrever mais do que nove "if"s.

Dicas:
- A ideia é usar um comando while para analisar cada casa decimal e gerar os caracteres romanos diferentemente para cada iteração.
- Use uma string ou dicionários para armazenar as letras correspondentes a cada casa decimal.

Exemplo: 1666 corresponde a string "MDCLXVI", onde:
as letras nas posições 5 e 6 correspondem às unidades (VI),
as letras nas posições 3 e 4 correspondem às dezenas (LX),
as letras nas posições 1 e 2 correspondem às centenas (DC),
a letra na posição 0 corresponde aos milhares (M).

47. Escreva um programa para converter números escritos em algarismos romanos (menores que 4000) para arábicos.

48. Escreva uma função fat_primo(num) que decompõe num em fatores primos e retorna um vetor com a decomposição.
Exemplo: 564757684810891538218830 27 = ['3^33', '7^4', '11^4', '17^2']

49. Considere alguma sequência de elementos (difere de um mero conjunto de elementos por definir uma ordem entre os seus membros). Enumere, então, todas as subsequências não contínuas de uma dada sequência.
Dicas:
- Uma subsequência sempre contém algum subconjunto de elementos da sequência, na mesma ordem.
- Uma subsequência contínua é aquela em que nenhum elemento está faltando, entre o primeiro e o último elemento da subsequência.
Exemplo: (1,2,3,4) → [[1, 2, 4], [1, 3, 4], [1, 3], [1, 4], [2, 4]]

50. Torres de Hanoi. Implemente a função hanoi(num), onde num é a quantidade de blocos do problema da torre de hanoi. A sua função deve dizer qual o movimento realizado e quantos blocos tem em cada torre após cada movimento.

51. Implemente a função resolve_abc(a,b,c). a, b e c são 3 string de mesmo comprimento para a equação a-b = c. Suponhamos que a = '7x2', b='28y' e c = 'x36'.Neste caso o resultado seria return 722, 286 , 436. Por hipótese, considere a>b e a> c.

52. Sudoku. Implemente a função Sudoku(matriz). matriz é uma matriz 9x9 com números de 0 a 9, onde 0 representaria uma posição vazia num jogo de Sudoku. A função deve trocar todas os números errados, segundo o critério do Sudoku por 0. Você pode utilizar os métodos embutidos nas classes list e set do Python.

53. Jogo da memória com 18 blocos. Crie um vetor com 18 posições, numeradas de 0 a 9 e 0 a 9 novamente. Misture estas posições. Crie um vetor de 18 posições com valores boleanos setados como False. Preencha uma numeração bem formatada de 0 a 17 no topo da tela e logo abaixo dela preencha com asteriscos, um para cada posição. O jogador então escolherá duas posições, de 0 a 17. Caso os números sejam idênticos, o a posição de valores booleanos será setada para True e a posição descoberta na próxima rodada. O jogo termina quando todas as posições forem reveladas. Forneça ao usuário o número de rodadas necessárias para completar o jogo.

54. Soma de números fracionários. Implemente a função soma_fracionarios(matriz). Matriz contem os números fracionais que serão somando. Por exemplos matriz = [[1,2],[-3,5],[2,7]]. Isto representaria a soma:

$$\frac{1}{2}+\frac{-3}{5}+\frac{2}{7}$$

O resultado desta soma seria:

$$\frac{35-42+20}{70}=\frac{13}{70}$$

A função neste caso retornaria [13, 70]. A matriz de retorno deve ser a mais simples possível. Suponha que o resultado fosse [4,10]. Neste caso, o valor de retorno deveria ser [2, 5].

55. Paradoxo do aniversário. Em teoria das probabilidades, o paradoxo do aniversário afirma que dado um grupo de 23 (ou mais) pessoas escolhidas aleatoriamente, a chance de que duas pessoas terão a mesma data de aniversário é de mais de 50%. Implemente um programa que mostra esta teoria. Podem ser utilizados os métodos embutidos nas classes list e set.