



## LISTA DE EXERCÍCIOS

- Arrays -

### **Vetores**

1. Gere um vetor com 30 valores inteiros que representam a umidade relativa do ar em cada dia de um mês. (Alimente o vetor com valores aleatórios entre 12 e 80). Conte e escreva os dias com maior e menor umidade relativa, a média mensal, e a quantidade de dias acima da média.
2. Faça um programa que preencha um vetor com N números inteiros aleatórios, sorteados no intervalo entre 0 e X. O programa deve imprimir os números sorteados, e ao final o seguinte resumo:
  - a) Quantos números distintos foram sorteados;
  - b) O(s) número(s) mais vezes sorteado;
  - c) O(s) número(s) menos vezes sorteado;
3. Faça um programa C para ler do usuário dois valores inteiros X e Y. Alimente dois vetores de tamanhos X e Y com números inteiros aleatórios, não repetitivos, sorteados no intervalo entre 0 e X+Y (inclusive). Faça a impressão das seguintes informações:
  - a) Vetores X e Y;
  - b) Todos os números exclusivos do vetor X;
  - c) Todos os números exclusivos do vetor Y;
  - d) Os números existentes nos dois vetores;
4. O algoritmo LRU (*Least Recently Used*) invalida o item que foi acessado menos recentemente. Isso significa que se um item não foi acessado há muito tempo, ele terá prioridade na remoção. Considere um vetor com N elementos, sorteados no intervalo entre X e Y (inclusive). Considerando que os primeiros índices foram os acessados recentemente, informe qual deveria ser o elemento excluído pela LRU.
5. Gere dois vetores, V1 e V2, cada um com K valores aleatórios sorteados no intervalo entre 0 e X ( $0 < K < X$ ). O usuário deve informar o valor de K e X, desde que  $K < X$ . O programa **NÃO** deverá aceitar números repetidos no mesmo vetor, e nem o mesmo número no mesmo índice dos dois vetores. Imprima os valores sorteados nos dois vetores, e o resultado da multiplicação dos valores dos índices de V1 e V2.
6. Métodos: map, filter e reduce são bastante comuns em linguagens de alto nível para tratamento de Arrays. Faça um programa que implemente o método...

**MAP** => Dado um vetor V1, com N números inteiros sorteados aleatoriamente (entre 0 e 20), e um valor inteiro X (sendo N e X informados pelo usuário), crie um novo vetor V2, preenchendo-o com os valores do primeiro vetor elevado à potência X (não usar funções prontas para realizar a potenciação). Imprima V1 e V2.
7. Métodos: map, filter e reduce são bastante comuns em linguagens de alto nível para tratamento de Arrays. Faça um programa que implemente o método...

**FILTER** => Dado um vetor V1, com N números inteiros sorteados aleatoriamente (entre 0 e 99), e um valor inteiro X (sendo N e X informados pelo usuário), crie um novo vetor V2, preenchendo-o apenas com valores do primeiro vetor que são divisíveis por X. Um mesmo valor não deve ser filtrado 2 ou mais vezes. Imprima V1 e V2.

8. Métodos: `map`, `filter` e `reduce` são bastante comuns em linguagens de alto nível para tratamento de Arrays. Faça um programa que implemente o método...
- REDUCE** => Dado um vetor V, com N números inteiros sorteados aleatoriamente (entre 0 e 9), imprima a média dos valores sorteados, excluindo-se do cálculo o menor e o maior valor sorteado.
9. Faça um programa que preencha um vetor com N números aleatórios, sorteados no intervalo entre X e Y. O programa deve informar a posição (índice) onde se encontra o menor número sorteado, seguido da posição onde se encontra o segundo menor número sorteado, e assim sucessivamente...
10. Leia do usuário um valor X, tal que X seja um valor PAR. Gere aleatoriamente X números inteiros não-repetitivos e armazene-os em uma estrutura do tipo Array. Imprima os números gerados na sequência em que foram sorteados. Após a impressão, troque no vetor a posição da metade inicial dos números sorteados pela metade final. Repita a impressão do vetor.
11. Gere um vetor com N números inteiros sorteados no intervalo entre X e Y (inclusive). Imprima o vetor sorteado, e a maior sequência crescente deste vetor (em caso de empate, imprima a primeira sequência).
12. Gere um vetor de N números inteiros sorteados no intervalo entre X e Y (inclusive). Imprima o vetor sorteado e depois elimine os valores repetidos, mantendo a ordem original do vetor. (Obs. A cada exclusão, o vetor deve ser rearranjado trazendo os elementos à frente para a posição anterior. Não é permitido usar vetor auxiliar). Imprima novamente o mesmo vetor.
13. Faça um programa que sorteie aleatoriamente N números, no intervalo entre X e Y (inclusive). Após o sorteio, o programa deve imprimir os números sorteados originalmente. Após isso, e a cada iteração, o programa deve imprimir novamente a relação, mas agora trocando as posições do MAIOR elemento pelo MENOR elemento.

Uma vez trocadas as posições, estes mesmos número não poderão ser mais trocados. Repita essa operação até quando for possível...

*Exemplo de Execução:*

N = 8

X = 1

Y = 10

Original: 4 – 6 – 8 – 9 – 3 – 2 – 7 – 8

1ª Iteração: 4 – 6 – 8 – 2 – 3 – 9 – 7 – 8

2ª Iteração: 4 – 6 – 3 – 2 – 8 – 9 – 7 – 8

3ª Iteração: 8 – 6 – 3 – 2 – 8 – 9 – 7 – 4

4ª Iteração: 8 – 7 – 3 – 2 – 8 – 9 – 6 – 4

14. Faça um programa que leia um número inteiro “K” e verifique se ele é palíndromo. P.Ex.: 57875, 131, 5665, 9, etc...
15. Declare um vetor de K números inteiros e preencha-o com números aleatórios entre 0 e 99 (inclusive). Imprima o vetor sorteado original, e o número de comparações e permutações necessárias para ordenação do vetor utilizando os seguintes algoritmos de ordenação: **Bubble Sort**, **Selection Sort** e **Insertion Sort**. Imprima também o vetor após o resultado das ordenações.

## Strings

16. Faça um programa em C que leia um valor inteiro correspondente ao seu número de matrícula no curso e, como saída, gera e imprime a string formada pelos dígitos deste número.
17. Escreva um programa em C para ler uma frase. A seguir, imprima a frase com: (a) todas as letras maiúsculas, (b) todas as letras minúsculas, (c) início de cada palavra em maiúsculo.
18. Faça um programa que verifique se um endereço de e-mail é válido ou não. Para um e-mail ser válido, deve possuir um único símbolo '@' (mas que não seja o primeiro), pelo menos um símbolo ponto '.' após o símbolo '@' (mas que não seja o último), e não possuir espaços em branco nem outros caracteres especiais. Faça várias validações na mesma execução.
19. Desenvolva um programa em C que faça a geração de senhas fortes. Uma senha forte é uma string contendo entre 8 e 16 caracteres, com obrigatoriamente: 1 caractere numérico, 1 caractere maiúsculo, 1 minúsculo e 1 caractere especial. Imprima 10 senhas fortes geradas aleatoriamente.
20. Faça um programa que simule o sorteio de um amigo oculto com N participantes. Leia o nome de cada amigo e imprima o resultado do sorteio. (Lembre que um amigo não pode tirar ele mesmo).
21. Faça um programa que leia, em formato de string, dois números inteiros excepcionalmente grandes e imprima a soma destes números. P.ex.: "25996478547851225" e "1452565475541" = 25997931113326766
22. Faça um programa que leia, em formato de string, um valor numérico representado na base binária. O programa deve validar se o valor informado pelo usuário realmente é um número binário. Em caso positivo, o programa deve informar o valor correspondente na base decimal.
23. Faça um programa que leia uma string A e outra string B. O programa deve informar a seguinte informação, conforme o caso: "**String A está contida em B**" ou "**String B está contida em A**" ou "**As strings são incompatíveis**".
24. O código de barras é essencial na automatização do processo de vendas. Esse código possui um formato conhecido por EAN-13, contendo 13 dígitos numéricos, sendo que os 3 primeiros identificam o País de origem (Brasil == 789), os 9 dígitos seguintes identificam a empresa fabricante e o produto em si, e o último - 13º dígito - é um validador, chamado de "Dígito Verificador - DV" que é calculado com base na sequência anterior, e que serve para dar segurança ao processo de leitura do código de barras. O cálculo do DV é o seguinte: Considerando do 1º ao 12º dígito, somar todas as posições pares e multiplicar a soma pelo valor 3. Ao resultado do passo anterior, somar todas as posições ímpares da sequência. O DV deverá ser o valor que tornará toda a soma alcançada em um múltiplo de 10. Faça um programa que valide Códigos de Barras lidos pelo usuário.
25. Semelhante ao problema anterior, pesquise e implemente o algoritmo para realizar a validação de um número de CPF.
26. Semelhante ao problema anterior, pesquise e implemente o algoritmo para realizar a validação de um número de cartão de crédito.
27. Faça um programa que calcule quantos segundos já se passaram no dia de hoje.  
Obs.: Para obter uma *string* com a data/horário atual utilize as expressões abaixo...  

```
time_t horario = time(NULL);  
printf("%s", asctime(localtime(&horario)));
```

28. Faça um programa que leia a data de nascimento do usuário no seguinte formato *string* DD/MM/AAAA. O seu programa deve validar a entrada, tanto no formato indicado, quanto na real existência da data informada. Após isto, obtenha a data atual, e calcule a idade completa do usuário (anos, meses e dias).
29. Escreva um programa em C para ler uma frase qualquer e contar o número de palavras existentes na frase. Considere uma palavra como sendo um conjunto maior que dois caracteres separados por um ou mais espaços em branco.
30. Faça um programa que leia uma string S, e criptografe-a com o seguinte algoritmo: sorteie um vetor de  $\text{strlen}(S)$  números, com valores aleatórios e não-repetitivos entre 0 e  $\text{strlen}(s)$ . Após o sorteio, embaralhe as letras da mensagem original, de acordo com as posições sorteadas no vetor. Atenção, todos os caracteres têm que ter sua posição inicial alterada!

*Exemplo de Execução:*

String:	AULA DE ED
Vetor Sorteado:	1369847502
Texto Cifrado:	EADUD LE A

31. Considerando o problema anterior, faça um programa que leia uma String S criptografada, e um vetor de  $\text{strlen}(s)$  números inteiros. Valide a consistência do vetor (não deve ter números repetidos e devem estar no intervalo adequado). Se tudo estiver em conformidade, informe a mensagem decifrada...

*Exemplo de Execução:*

Texto Cifrado:	EADUD LE A
Vetor (Chave):	1369847502
Texto Decifrado:	AULA DE ED

32. Em uma eleição para representante da turma, existem K candidatos. Faça com que seu programa leia o valor de K e o nome de cada um dos candidatos.
- Após isso, programe uma espécie de urna eletrônica, onde o voto do candidato é representado pelo seu número de sequência de cadastro na urna (p.ex.: 1 - 2 - 3 - ... - K).
- O valor 0 corresponde a um voto em BRANCO, e um valor superior a K representa um voto NULO.
- A votação deve ser encerrada assim que for informado um voto com valor negativo.

Ao final da votação, seu programa deve gerar o seguinte relatório:

- A quantidade total de votos;
- O nome e a quantidade de votos para cada candidato;
- A quantidade de votos nulos;
- A quantidade de votos em branco;
- O nome do vencedor das eleições;
- O percentual do vencedor em relação ao número de votos válidos;

### **Matrizes**

33. Gere aleatoriamente uma matriz N x N de inteiros sorteados no intervalo entre X e Y (inclusive). Imprima-a em formato de tabela (linhas e colunas) e informe a localização do **maior** e do **menor** valor da matriz.
34. Gere aleatoriamente uma matriz N x N de inteiros sorteados no intervalo entre X e Y (inclusive). Imprima-a em formato de tabela, e os valores que compõem a coluna que resulta na **maior** soma, e os valores da linha que resulta na **menor** soma.

- 35.** Gere aleatoriamente uma matriz  $N \times N$  de inteiros sorteados no intervalo entre 0 e  $N^2$  (inclusive). Todo número sorteado, obrigatoriamente deve ser exclusivo em sua linha e em sua coluna. Imprima a matriz em formato de tabela (linhas/colunas).
- 36.** Gere aleatoriamente uma matriz  $N \times N$  de inteiros sorteados no intervalo entre X e Y (inclusive). Peça ao usuário o índice de duas colunas, e faça a troca das respectivas colunas. Faça o mesmo com duas linhas escolhidas pelo usuário. Imprima a matriz após realizadas as trocas.
- 37.** Faça um programa que leia o Nome e a Idade de N pessoas. Imprima a relação de nomes ordenada pelas idades de forma decrescente.
- 38.** Gere uma cartela de bingo  $5 \times 5$ , com números aleatórios entre 1 e 75. (Em uma cartela de bingo, não há números repetidos e os números são apresentados em ordem crescente). Imprima a cartela e faça com que o programa sorteie as bolas, uma a uma, e realize a “marcação” da cartela em tempo real.
- 39.** Crie um programa que leia o nome de N alunos. Após isto, para cada aluno cadastrado, imprima o nome e solicite ao usuário os valores das notas da prova 1, prova 2 e prova 3. Ao final, o programa deverá gerar um relatório em formato de lista, contendo NOME, NOTA 1, NOTA 2, NOTA 3, MÉDIA, e se está APROVADO ou REPROVADO.
- 40.** Gere uma matriz  $10 \times 100$ , com números aleatórios entre 1 e 4, que se refere às respostas de 10 questões de múltipla escolha, referentes a 100 alunos. Leia (a partir do usuário) um vetor de 10 posições contendo o gabarito de respostas corretas. O seu programa deverá comparar as respostas de cada candidato com o gabarito, e emitir a nota final de cada aluno. O programa também deverá informar: qual foi a média das notas, a maior nota, quantos alunos atingiram a maior nota, a menor nota, e quantos alunos atingiram a menor nota.