

Lista de Exercícios – Engenharias

Programação I e II – Prof. Me. Joseffe Barroso de Oliveira

Módulo 01 – Entrada, Processamento e Saída

1. Entrar via teclado com a base e a altura de um retângulo, calcular e exibir sua área.
2. Calcular e exibir a área de um quadrado, a partir do valor de sua aresta que será digitado.
3. A partir dos valores da base e altura de um triângulo, calcular e exibir sua área.
4. Calcular e exibir a média aritmética de quatro valores quaisquer que serão digitados.
5. Entrar via teclado com o valor de uma temperatura em graus Celsius, calcular e exibir sua temperatura equivalente em Fahrenheit.
6. Entrar via teclado com o valor da cotação do dólar e uma certa quantidade de dólares. Calcular e exibir o valor correspondente em Reais (R\$).
7. Entrar via teclado com o valor de cinco produtos. Após as entradas, digitar um valor referente ao pagamento da somatória destes valores. Calcular e exibir o troco que deverá ser devolvido.

Módulo 02 – Estrutura de Decisão

8. Entrar via teclado, com dois valores distintos. Exibir o maior deles.
9. Entrar via teclado, com dois valores distintos. Exibir o menor deles.
10. Entrar com dois valores quaisquer. Exibir o maior deles, se existir, caso contrário, enviar mensagem avisando que os números são idênticos.
11. Calcular e exibir a área de um retângulo, a partir dos valores da base e altura que serão digitados. Se a área for maior que 100, exibir a mensagem “Terreno grande”.
12. Calcular e exibir a área de um retângulo, a partir dos valores da base e altura que serão digitados. Se a área for maior que 100, exibir a mensagem “Terreno grande”, caso contrário, exibir a mensagem “Terreno pequeno”.
13. Entrar via teclado com três valores distintos. Exibir o maior deles.
14. Entrar com o peso e a altura de uma determinada pessoa. Após a digitação, exibir se esta pessoa está ou não com seu peso ideal. Veja tabela da relação peso/altura².

Relação peso/altura ² (R)	Mensagem
$R < 20$	Abaixo do peso
$20 \leq R < 25$	Peso ideal
$R \geq 25$	Acima do peso

15. A partir de três valores que serão digitados, verificar se formam ou não um triângulo. Em caso positivo, exibir sua classificação: “Isósceles, escaleno ou equilátero”. Um triângulo escaleno possui todos os lados diferentes, o isósceles, dois lados iguais e o equilátero, todos os lados iguais. Para existir triângulo é necessário que a soma de dois lados quaisquer seja maior que o outro, isto, para os três lados.
16. Verificar se três valores quaisquer (A, B, C) que serão digitados formam ou não um triângulo retângulo. Lembre-se que o quadrado da hipotenusa é igual a soma dos quadrados dos catetos.
17. Entrar com o peso, o sexo e a altura de uma determinada pessoa. Após a digitação, exibir se esta pessoa está ou não com seu peso ideal. Veja tabela da relação peso/altura².

Peso/altura ² (R) - Femininos	Mensagem
< 19	Abaixo do peso
$19 \leq R < 24$	Peso ideal
$R \geq 24$	Acima do peso

Peso/altura ² (R) - Masculinos	Mensagem
$R < 20$	Abaixo do peso
$20 \leq R < 25$	Peso ideal
$R \geq 25$	Acima do peso

18. A partir dos valores da aceleração (**a** em m/s²), da velocidade inicial (**v0** em m/s) e do tempo de percurso (**t** em s). Calcular e exibir a velocidade final de automóvel em km/h. Exibir mensagem de acordo com a tabela:

Velocidade em Km/h (V)	Mensagem
$V \leq 40$	Veículo muito lento
$40 < V \leq 60$	Velocidade permitida
$60 < V \leq 80$	Velocidade de cruzeiro
$80 < V \leq 120$	Veículo rápido
$V > 120$	Veículo muito rápido

Fórmula para o cálculo da velocidade em m/s: $V = v0 + a \cdot t$

19. Uma escola com cursos em regime semestral, realiza duas avaliações durante o semestre e calcula a média do aluno, da seguinte maneira:

$$MEDIA = (P1 + 2P2) / 3$$

Fazer um programa para entrar via teclado com os valores das notas (P1 e P2) e calcular a média. Exibir a situação final do aluno (“Aprovado ou Reprovado”), sabendo que a média de aprovação é igual a cinco.

20. Uma escola com cursos em regime semestral realiza duas avaliações durante o semestre e calcula a média do aluno, da seguinte maneira:

$$MEDIA = (P1 + 2P2) / 3$$

Fazer um programa para entrar via teclado com o valor da primeira nota (P1) e o programa deverá calcular e exibir quanto o aluno precisa tirar na segunda nota (P2) para ser aprovado, sabendo que a média de aprovação é igual a cinco.

21. Entrar via teclado com dois valores quaisquer. Após a digitação, exibir um seletor de opções (“menu”) com as seguintes opções: (Fazer esse exercício utilizando If..Else e/ou Case)

- 1 – Multiplicação
- 2 – Adição
- 3 – Divisão
- 4 – Subtração
- 5 – Fim de processo

Solicitar uma opção por parte do usuário, verificar se é ou não uma opção válida (letras ou números) e processar a respectiva operação. Enviar mensagem de erro se a opção escolhida não existir no seletor. Encerrar o programa somente quando o usuário escolher a opção de finalização. Repare que a opção de número três é de divisão e o programa deverá enviar mensagem de erro, (somente nesta opção) se o denominador for zero.

22. Exibir o seguinte seletor de opções e em função de uma escolha, solicitar os dados necessários para o cálculo da respectiva área. Enviar mensagem de erro se o usuário escolher uma opção inexistente. Encerrar o programa somente quando selecionada a opção de finalização. (Fazer esse exercício utilizando If..Else e/ou Case)

- 1 – Triângulo
- 2 – Quadrado
- 3 – Retângulo
- 4 – Círculo
- 5 – Fim de processo

Módulo 03 – Estrutura de Repetição

23. Criar uma rotina de entrada que aceite somente um valor positivo.
24. Entrar com dois valores via teclado, onde o segundo deverá ser maior que o primeiro. Caso contrário solicitar novamente apenas o segundo valor.
25. Entrar via teclado com o sexo de determinado usuário, aceitar somente “F” ou “M” como respostas válidas.
26. Exibir a tabuada do número cinco no intervalo de um a dez.
27. Entrar via teclado com um valor qualquer. Travar a digitação, no sentido de aceitar somente valores positivos. Após a digitação, exibir a tabuada do valor solicitado, no intervalo de um a dez.
28. Entrar via teclado com um valor (X) qualquer. Travar a digitação, no sentido de aceitar somente valores positivos. Solicitar o intervalo que o programa que deverá calcular a tabuada do valor digitado, sendo que o segundo valor (B), deverá ser maior que o primeiro (A), caso contrário, digitar novamente somente o segundo. Após a validação dos dados, exibir a tabuada do valor digitado, no intervalo **decrescente**, ou seja, a tabuada de X no intervalo de B para A.
29. Exibir a tabuada dos valores de um a vinte, no intervalo de um a dez. Entre as tabuadas, solicitar que o usuário pressione uma tecla.
30. Exibir a soma dos números inteiros positivos do intervalo de um a cem.
31. Exibir os trinta primeiros valores da série de Fibonacci. A série: 1, 1, 2, 3, 5, 8, ...
32. Exibir os vinte primeiros valores da série de Bergamaschi. A série: 1, 1, 1, 3, 5, 9, 17, ...
33. Calcular e exibir a soma dos “N” primeiros valores da sequência abaixo. O valor “N” será digitado, deverá ser positivo, mas menor que cem. Caso o valor não satisfaça a restrição, enviar mensagem de erro e solicitar o valor novamente.
A sequência: 2, 5, 10, 17, 26,
34. Calcular e exibir a soma dos “N” primeiros valores da sequência abaixo. O valor “N” será digitado, deverá ser positivo, mas menor que cinquenta. Caso o valor não satisfaça a restrição, enviar mensagem de erro e solicitar o valor novamente.
A sequência: $\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{4}$, $\frac{4}{5}$,
35. Calcular e exibir a soma dos “N” primeiros valores da sequência abaixo. O valor “N” será digitado, deverá ser positivo, mas menor que cinquenta. Caso o valor não satisfaça a restrição, enviar mensagem de erro e solicitar o valor novamente.
A sequência: 2, $\frac{5}{8}$, $\frac{10}{27}$, $\frac{17}{64}$,
36. Entrar via teclado com dez valores positivos. Consistir a digitação e enviar mensagem de erro, se necessário. Após a digitação, exibir:
 - a) O maior valor;
 - b) A soma dos valores;
 - c) A média aritmética dos valores;
37. Entrar via teclado com “N” valores quaisquer. O valor “N” (que representa a quantidade de números) será digitado, deverá ser positivo, mas menor que vinte. Caso a quantidade não satisfaça a restrição, enviar mensagem de erro e solicitar o valor novamente. Após a digitação dos “N” valores, exibir:
 - a) O maior valor;
 - b) O menor valor;
 - c) A soma dos valores;
 - d) A média aritmética dos valores;
 - e) A porcentagem de valores que são positivos;
 - f) A porcentagem de valores negativos;
38. Entrar via teclado com “N” valores quaisquer. O valor “N” (que representa a quantidade de números) será digitado, deverá ser positivo, mas menor que vinte. Caso a quantidade não satisfaça a restrição, enviar mensagem de erro e solicitar o valor novamente. Após a digitação dos “N” valores, exibir:

- a) O maior valor;
- b) O menor valor;
- c) A soma dos valores;
- d) A média aritmética dos valores;
- e) A porcentagem de valores que são positivos;
- f) A porcentagem de valores negativos;

Após exibir os dados, perguntar ao usuário de deseja ou não uma nova execução do programa. Consistir a resposta no sentido de aceitar somente “S” ou “N” e encerrar o programa em função dessa resposta.

39. Calcular o fatorial de um valor que será digitado. Este valor não poderá ser negativo. Enviar mensagem de erro e solicitar o valor novamente, se necessário. Perguntar se o usuário deseja ou não fazer um novo cálculo, consistir a resposta em “S” ou “N”.

$$N! = N \times N-1 \times N-2 \times N-3 \times \dots \times (N - (N-1))$$

$$\text{Ex: } 5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

40. O jogo da mega-sena consiste em acertar seis dos sessenta números disponíveis em um volante. Fazer um programa para calcular a quantidade de jogos que temos que fazer, para com certeza acertar o resultado da mega-sena. Admitindo que faremos jogos de seis números por volante, o programa deverá exibir quais seriam estes números em cada volante, ou seja, exibir todos os resultados possíveis.

Módulo 04 – Vetores e Matrizes

- 41. Armazenar dez números na memória do computador. Exibir os valores na ordem inversa à da digitação.
- 42. Armazenar dez valores na memória do computador. Após a digitação dos valores, criar uma rotina para ler os valores e achar e exibir o maior deles.
- 43. Armazenar vinte valores em um vetor. Após a digitação, entrar com uma constante multiplicativa que deverá multiplicar cada um dos valores do vetor e armazenar o resultado no próprio vetor, na respectiva posição.
- 44. Armazenar vinte valores na memória. Após a digitação, entrar com uma constante multiplicativa que deverá multiplicar cada um dos valores do vetor e armazenar o resultado em outro vetor, porém em posições equivalentes. Exibir os vetores na tela.
- 45. Armazenar um máximo de 20 valores em um vetor. A quantidade de valores a serem armazenados será escolhida pelo usuário. Enviar mensagem de erro, caso a quantidade de valores escolhida esteja fora da faixa possível e solicitar a quantidade novamente. Após a digitação dos valores, criar uma rotina de consulta, onde o usuário digita um número e o programa exibe em qual posição do vetor este número está localizado. Se o número não for encontrado, enviar mensagem “Valor não encontrado!”. Perguntar ao usuário se deseja ou não fazer uma nova consulta, consistir a resposta e encerrar o programa em caso negativo.
- 46. Armazenar o nome e idade de cem pessoas. Após a digitação, exibir os dados (nome e idade) de todas as pessoas.
- 47. Armazenar o nome, sexo e idade de cem pessoas. Consistir as entradas no sentido de aceitar apenas “F” ou “M” para o sexo e valores positivos para a idade. Após a digitação dos dados, exibir os dados (nome, sexo e idade) de todas as pessoas do sexo feminino.
- 48. Armazenar o nome, sexo e idade de cem pessoas. Consistir as entradas no sentido de aceitar apenas “F” ou “M” para o sexo e valores positivos para a idade. Após a digitação, exibir os dados (nome, sexo e idade) de todas as pessoas com idade superior a dezoito anos. Ao final da lista, exibir a quantidade de pessoas listadas.
- 49. Armazenar o nome, sexo e idade de cem pessoas. Consistir as entradas no sentido de aceitar apenas “F” ou “M” para o sexo e valores positivos para a idade. Após a digitação dos dados, exibir os dados de todas as pessoas com idade superior a vinte anos. No final da listagem, exibir a quantidade de pessoas que foram listadas e a porcentagem que este valor representa em relação ao total de pessoas digitadas.
- 50. Alterar o programa anterior, no sentido de controlar o layout final de tela, para que o usuário pressione uma tecla para visualizar os dados das pessoas passo a passo, por exemplo, de dez em dez pessoas.
- 51. Armazenar vinte valores em um vetor. Após a digitação, exibir os valores em ordem crescente.

52. Armazenar vinte valores em um vetor. Após a digitação, exibir os valores em ordem decrescente.
53. Armazenar o nome, sexo e idade de vinte pessoas. Após a digitação, exibir os dados (nome, sexo e idade) em ordem decrescente de idade.
54. Armazenar o nome, sexo e idade de vinte pessoas. Após a digitação, exibir os dados (nome, sexo e idade) classificados por sexo (crescente), depois por idade (decrescente) e finalmente por nome (alfabética). Por exemplo:

Com as seguintes entradas:

SOLANGE	F	34
MARCELO	M	36
JOAO	M	11
MAGNA	F	34
JUNIOR	M	6
MARIA	F	35
LUCAS	M	10
GABRIEL	M	7
FELIPE	M	10

Teremos a seguinte saída:

MARIA	F	35
MAGNA	F	34
SOLANGE	F	34
MARCELO	M	36
JOAO	M	11
FELIPE	M	10
LUCAS	M	10
GABRIEL	M	7
JUNIOR	M	6

55. Armazenar um máximo de 20 valores em um vetor. A quantidade de valores a serem armazenados será escolhida pelo usuário. Enviar mensagem de erro, caso a quantidade de valores escolhida esteja fora da faixa possível e solicitar a quantidade novamente. Após a digitação, calcular a *mediana* destes valores, sabendo que quando a quantidade de valores for par, a mediana é a média aritmética dos dois valores intermediários da lista e quando a quantidade de valores for ímpar, a mediana é o valor que está no ponto médio da lista. Nos dois casos, a lista deverá estar ordenada. Por exemplo:

Suponha os valores digitados: 7, 5, 2, 4 → mediana = $(4+5)/2 = 4,5$

Suponha os valores digitados: 7, 5, 2, 4, 8 → mediana = 5

56. Armazenar seis valores em uma matriz de ordem 2x3. Apresentar os valores na tela.
57. Armazenar seis valores em uma matriz de ordem 3x2. Apresentar os valores na tela.
58. Armazenar seis nomes em uma matriz de ordem 2x3. Apresentar os nomes na tela.
59. Entrar via teclado com doze valores e armazená-los em uma matriz de ordem 3x4. Após a digitação dos valores solicitar uma constante multiplicativa, que deverá multiplicar cada valor matriz e armazenar o resultado na própria matriz, nas posições correspondentes.
60. Entrar via teclado com doze valores e armazená-los em uma matriz de ordem 3x4. Após a digitação dos valores solicitar uma constante multiplicativa, que deverá multiplicar cada valor matriz e armazenar o resultado em outra matriz de mesma ordem, nas posições correspondentes. Exibir as matrizes na tela, sob a forma matricial, ou seja, linhas por colunas.
61. Entrar com uma matriz de ordem MxN, onde a ordem também será escolhida pelo usuário, sendo que no máximo 10x10. A matriz não precisa ser quadrática. Após a digitação dos elementos, criar uma rotina de consulta, onde o usuário digita um valor e a rotina exibe em qual (ou quais) posição da matriz, o valor escolhido se encontra. Enviar mensagem comunicando se por acaso o valor não estiver armazenado na matriz. Perguntar ao usuário, se deseja ou não fazer nova consulta.
62. Criar um programa que represente um “Jogo da Velha”, onde o programa solicita os nomes dos jogadores e passa a administrar a jogada de cada um. Solicitar de cada jogador, a posição do “tabuleiro” que deseja jogar e consistir esta posição. Evidentemente que não poderão ser feitas jogadas em posições já ocupadas. Exibir mensagem divulgando o nome do vencedor ou então “empate”, se for o caso. Perguntar se o usuário deseja ou não uma nova partida.
63. Vamos fazer uma excursão para Penápolis. Para isto, vamos de ônibus, e precisaremos controlar as reservas de lugares do ônibus. Sabe-se que o ônibus possui quatro fileiras de dez cadeiras cada. Fazer um programa para solicitar o nome do usuário e o lugar que pretende reservar (fileira e cadeira), e se este lugar estiver disponível o programa deverá concretizar a reserva, caso contrário, enviar mensagem comunicando que esta poltrona já está ocupada. Perguntar se existe mais alguém que queira viajar para a metrópole, e em caso negativo exibir um “mapa” mostrando os nomes e lugares de cada passageiro que efetuou a reserva, assim como os lugares que permanecem livres. Lembre-se que o ônibus possui uma capacidade limitada de poltronas e que o programa deverá encerrar estas reservas, caso esta capacidade tenha sido alcançada.

64. Criar um programa para controlar as reservas de poltronas de uma peça teatral, “*Marcelo e suas Mainframes*”, sabendo que a peça será apresentada em três sessões, manhã, tarde e noite e que o teatro possui “X” fileiras de “Y” cadeiras cada. Os valores de “X” e “Y” serão digitados. O usuário digita seu nome, lança a sessão, o número da fileira e da cadeira que pretende reservar, e se estiver livre a reserva será efetuada, caso contrário, o programa deverá enviar mensagem comunicando que a poltrona está ocupada e solicitar outro lugar. Perguntar ao usuário se mais alguém pretende fazer reservas. As reservas poderão ser efetuadas até completar um máximo de quatro quintos da capacidade total do teatro. No final do programa de reservas, exibir um “mapa” mostrando as poltronas do teatro, por sessão, com os nomes de cada ocupante, ou ainda com a informação “Poltrona livre”.