

Linguagem de Programação C

Lista de Exercícios

Variáveis, entrada e saída

Exercícios de Fixação

- 1) Faça um programa que imprima o produto entre 35 e 47.
- 2) Faça um programa que imprima a média aritmética entre os números 12, 25 e 53.
- 3) Faça um programa que leia 2 números da entrada e imprima o seu produto.
- 4) Faça um programa que leia 2 números da entrada e imprima a sua média aritmética.
- 5) Faça um programa que leia 2 números da entrada e imprima o resto da divisão inteira de um pelo outro.
- 6) Faça um programa que leia um número da entrada e imprima o seu antecedente e o seu sucessor.
- 7) Faça um programa que leia um valor de conta de restaurante, representando o gasto realizado pelo cliente e imprima o valor total a ser pago, considerando que o restaurante cobra 10% para o garçom.
- 8) Faça um programa que leia um valor de hora e informe quantos minutos se passaram deste o início do dia.
- 11) Faça um programa que leia 2 números e imprima uma mensagem dizendo se são iguais ou diferentes.
- 12) Faça um programa que leia 2 números e imprima uma mensagem dizendo o maior deles. Suponha que os números serão diferentes.
- 13) Faça um programa que leia 2 números e imprima uma mensagem dizendo o maior deles. Detalhe: se os números forem iguais, imprima uma mensagem avisando ao usuário.
- 14) Faça um programa que leia 2 números e os imprima em ordem crescente.
- 15) Faça um programa que leia 2 números e os imprima em ordem decrescente.
- 16) Faça um programa que leia 3 números e imprima o menor deles.
- 17) Faça um programa que leia 3 números e os imprima em ordem crescente.
- 18) Faça um programa que leia 3 números e os imprima em ordem decrescente.
- 19) Faça um programa que leia 3 números e imprima o valor intermediário, entre o menor e o maior número. Suponha que os números serão diferentes.
- 20) Faça um programa que leia 3 números e imprima uma das seguintes mensagens:
 - todos os números são iguais;
 - todos os números são diferentes ou
 - apenas dois números são iguais.
- 21) Faça um programa que leia três notas de um aluno, calcule sua média aritmética e imprima uma mensagem dizendo se o aluno foi aprovado, reprovado ou deverá fazer prova final. O critério de aprovação é o seguinte:
 - aprovado ($\text{média} \geq 7$);

Tomada de decisão

Exercícios de Fixação

- 9) Faça um programa que leia da entrada um número e imprima o seu valor absoluto (sem o sinal).
- 10) Faça um programa que leia um número e imprima uma mensagem dizendo se é par ou ímpar.

Linguagem de Programação C

- reprovado (média < 3) e
- prova final ($3 \leq \text{média} < 7$).

22) Faça um programa que leia a idade de uma pessoa e informe sua classe eleitoral:

- não eleitor (abaixo de 16 anos);
- eleitor obrigatório (entre 18 e 65 anos) e
- eleitor facultativo (entre 16 e 18 anos e acima dos 65 anos).

23) Faça um programa que leia um horário no sistema de 24 horas e imprima este horário no sistema de 12 horas.

Exemplo: 22:32 → 10:32 pm

Obs.: 0:00 → 12:00 am

6:00 → 6:00 am

12:00 → 12:00 pm

Repetição

Exercícios de Fixação

24) Faça um programa que imprima os números de 1 a 100.

25) Faça um programa que imprima os números de 100 a 1, nesta ordem.

26) Faça um programa que imprima os números pares no intervalo de 1 a 300.

27) Faça um programa que imprima os números de 64 a 256.

28) Faça um programa que imprima os números ímpares no intervalo de 531 a 750.

29) Faça um programa que leia um número da entrada e imprima os números naturais de 0 até este número. Suponha que o número lido da entrada será maior que zero.

30) Faça um programa que leia um número da entrada e imprima os números naturais de 0 até este número, pulando de dois em dois. Suponha que o número lido da entrada será maior que zero.

31) Faça um programa que leia um número da entrada (*limite*), um incremento (*incr*) e imprima os

números naturais de 0 até *limite* pulando de *incr*. Suponha que *limite* e *incr* são maiores que zero.

Exemplo:

valores lidos: 10 (*limite*) 3 (*incr*)

impressão: 0 3 6 9

32) Faça um programa que leia um número da entrada e imprima os números ímpares menores do que este número. Exemplo:

valor lido: 7 impressão: 1 3 5

33) Faça um programa que leia um número da entrada (*num*) e imprima os *num* primeiros números pares. Exemplo:

valor lido: 6

impressão: 2 4 6 8 10 12

34) Faça um programa que imprima a tabuada de 5 no formato:

5 x 1 = 5

5 x 2 = 10

... ..

5 x 10 = 50

35) Faça um programa que leia um número da entrada e imprima a tabuada deste número. Suponha que o número lido da entrada é maior que zero.

36) Faça um programa que imprima as tabuadas dos números de 1 a 10.

37) Faça um programa que imprima os números de 1 a 100 e o valor de seu somatório.

38) Faça um programa que leia um número da entrada (*num*), imprima os números de 1 a *num* e o seu somatório.

39) Faça um programa que imprima a soma dos números pares entre 25 e 100.

40) Faça um programa que leia um número da entrada (*num*) e imprima a soma dos números pares entre 1 e *num*. Suponha que *num* será maior que zero.

41) Faça um programa que leia um número da entrada (*num*) e imprima o somatório dos *num* primeiros números ímpares.

Lembre-se sempre

“A Sorte Ajuda aos Competentes”

Página 2 de 32

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

42) Faça um programa que leia um intervalo da entrada (*início* e *fim*) e imprima os números naturais neste intervalo. Exemplo:

valores lidos: 5 9
impressão: 5 6 7 8 9

43) Faça um programa que leia um intervalo da entrada (*início* e *fim*) e imprima os múltiplos de 5 entre eles. Exemplo:

valores lidos: 3 17
impressão: 5 10 15

Observação: o que acontece se *fim* for menor que *início*? Seu programa continua funcionando?

44) Faça um programa que leia um intervalo da entrada (*início* e *fim*) e um número (*num*). O programa deve imprimir os múltiplos de *num* entre *início* e *fim*. Exemplo:

valores lidos: 1 15 3 (*num*)
impressão: 3 6 9 12 15

Observação: o que acontece se *fim* for menor que *início*? Seu programa continua funcionando?

45) Faça um programa que leia um intervalo da entrada (*início* e *fim*) e imprima os números pares entre *início* e *fim* e seu somatório. Exemplo:

valores lidos: 3 15
impressão: 4 6 8 10 12 14
Soma = 54

46) Faça um programa que leia um número (*num*) da entrada e imprima os múltiplos de 3 e 5 ao mesmo tempo no intervalo de 1 a *num*. Exemplo:

valor lido: 47
impressão: 15 30 45

47) Faça um programa que leia um número da entrada, calcule e imprima o seu fatorial. Lembre-se que:

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 1$$
$$0! = 1 \text{ e } 1! = 1$$

48) Faça um programa que leia um número da entrada (*num*), a seguir leia *num* números da entrada e os imprima. Exemplo:

valores lidos: 5
3 10 12 2 1
impressão: 3 10 12 2 1

49) Faça um programa que leia um número da entrada (*num*), a seguir leia *num* números da entrada e imprima o maior deles. Suponha que todos os números lidos serão positivos. Exemplo:

valores lidos: 3
17 4 22
impressão: O maior número lido foi 22.

50) Faça um programa que leia um número da entrada (*num*), a seguir leia *num* números da entrada e imprima o maior deles. Exemplo:

valores lidos: 4
-3 -5 -15 -4
impressão: O maior número lido foi -3.

51) Faça um programa que leia um número da entrada (*num*), a seguir leia *num* números da entrada e imprima o maior, o menor e soma dos números lidos.

52) Faça um programa que leia uma sequência de números terminada por 0 e imprima o maior, o menor e a média aritmética dos números. O número 0 (zero) não faz parte da sequência. Exemplo:

valores lidos: 20 -3 4 -7 1 0
impressão: Maior = 20 Menor = -7
Média = 3

53) Faça um programa que leia uma sequência de números terminada por 0 e imprima o número que for múltiplo de sua posição na sequência.

valores lidos: 3 7 8 16 5 0
posição: 1 2 3 4 5
impressão: 3 16 5

54) Faça um programa que leia uma sequência de números terminada por 0 e imprima a média aritmética dos ímpares e a soma dos pares.

55) Faça um programa que calcule e imprima o valor de 2^5 . Dica: $2^5 = 2 \times 2 \times 2 \times 2 \times 2$.

56) Faça um programa que leia dois números da entrada (*X* e *Y*) e calcule X^Y . Suponha que *X* e *Y* são positivos.

57) Faça um programa que calcule o produto de 2 números lidos da entrada (*num1* e *num2*) através do

Linguagem de Programação C

método de somas sucessivas. Suponha que *num1* e *num2* são positivos. Exemplo:

valores lidos: 3 4
impressão: 12
dica: $3 \times 4 = 3 + 3 + 3 + 3 = 12$

58) Faça um programa que leia 2 números da entrada e imprima o resto da divisão inteira do primeiro pelo segundo usando subtrações sucessivas.

59) Faça um programa que calcule o resultado de $\frac{num1}{num2}$ utilizando o algoritmo de subtrações sucessivas. *num1* e *num2* devem ser lidos da entrada. Informe ao usuário caso a divisão não seja exata. Exemplo:

valores lidos: 6 3
impressão: 2
dica: $6 - 3 = 3 - 3 = 0$
(2 subtrações)

60) Faça um programa que imprima os termos de uma P.A., sendo lidos da entrada: o primeiro termo (*a₀*), a razão (*r*) e o número de elementos (*num*).

61) Faça um programa que imprima os termos de uma P.G., sendo lidos da entrada: o primeiro termo (*a₀*), a razão (*q*) e o número de elementos (*num*).

62) Faça um programa que leia um número da entrada e imprima uma mensagem dizendo se o número é solução da equação $2x^2 - 7x + 3 = 0$.

63) Escreva um algoritmo que leia uma sequência de 50 números naturais e imprima o maior número da sequência múltiplo de 2 e de 3 ao mesmo tempo.

64) Faça um programa que imprima os 10 primeiros termos da série de *Fibonacci*.

Observação: $a_0 = 1$, $a_1 = 1$ e $a_n = a_{n-1} + a_{n-2}$.

65) Faça um programa que leia um número (*num*) da entrada e imprima os *num* primeiros termos da série de *Fibonacci*.

66) Faça um programa que leia um número da entrada e imprima os termos da série de *Fibonacci* menores que este valor.

67) Faça um programa que calcule $\prod_{i=N}^K \frac{1}{i}$, onde N e

K são lidos da entrada.

68) (Prova 1 - 96/1) Faça um programa que imprima a tabela de conversão de graus **Celsius-Fahrenheit** para o intervalo desejado pelo usuário. O programa deve solicitar ao usuário o limite superior, o limite inferior do intervalo e o decremento.

Fórmula de conversão: $C = 5(F - 32) / 9$

Exemplo:

valores lidos:	68	50	15
impressão:	Fahrenheit	Celsius	
	68	20	
	53	11	

Exercícios Propostos

69) (Prova 1 - 95/2) A série de **RICCI** é gerada da seguinte forma: os dois primeiros termos são fornecidos pelo usuário, a partir daí, os termos são gerados como a soma dos dois termos anteriores, ou seja, $a_i = a_{i-1} + a_{i-2}$ para $i \geq 3$.

Escreva um programa que imprima os N (N lido e ≥ 2) primeiros termos da série de **RICCI**. O programa deve imprimir, também a soma dos termos impressos.

70) (Prova 1 - 96/1) A série de **FETUCCINE** é gerada da seguinte forma: os dois primeiros termos são fornecidos pelo usuário; os termos seguintes são gerados como a soma ou subtração dos dois termos anteriores, ou seja:

$$a_i = a_{i-1} + a_{i-2} \implies \text{para } i \text{ ímpar}$$

$$a_i = a_{i-1} - a_{i-2} \implies \text{para } i \text{ par}$$

Escreva um programa que imprima os N (N lido e maior ou igual a 2) primeiros termos da série de **FETUCCINE**. O programa deve imprimir, também a soma dos termos impressos.

71) (Prova 1 - 96/2) Uma telenovela de grande sucesso tem o seu final previsto para o dia 1^º de novembro. Sabe-se que neste dia será revelado nome do personagem responsável por inúmeros assassinatos ocorridos no decorrer da trama. No entanto, no dia 1^º de outubro (um mês antes do

Lembre-se sempre

“A Sorte Ajuda aos Competentes”

Página 4 de 32

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

término da novela), um funcionário da emissora conta para dois amigos o desfecho do último capítulo. Com isto, no dia 1^o de outubro, 3 pessoas sabem do desfecho da novela. Supondo que cada nova pessoa a saber do final da novela contará para duas novas pessoas no dia seguinte (**e para mais ninguém**), mais 4 pessoas estarão envolvidas ao final do dia 2 de outubro (7 pessoas até este dia). No dia 3 de outubro, já serão 15 pessoas.

Faça um programa para ajudar o dono da emissora a decidir se deve demitir o funcionário fofoqueiro: se ao término do dia 25 de outubro mais de 20000 (vinte mil) pessoas estiverem sabendo do final da novela, o funcionário deverá ser demitido.

72) Faça um programa que calcule o M.M.C entre dois números lidos da entrada.

73) Faça um programa que calcule o M.D.C entre dois números lidos da entrada.

74) Faça um programa que calcule e imprima os termos da série $\frac{X^N}{N!}$, onde X e N devem ser lidos da entrada.

75) Faça um programa que calcule o resultado de $\sum_{I=N}^0 (-1)^I * I$, onde N é lido da entrada.

76) Escrever um algoritmo para escrever na saída todos os números (inteiros e positivos) entre 10 e 100 tais que:

- Não terminem em zero;
- Se o dígito da direita for removido, o número restante é divisor do número original.

Exemplos:

39: 3 é divisor de 39

48: 4 é divisor de 48

77) Dados N valores (inteiros e positivos) na entrada, escrever um algoritmo que teste quais valores estão dentro e quais estão fora do intervalo {1, 2, ..., 15}, escrevendo na saída o valor e a mensagem correspondente. O primeiro valor a ser lido na entrada é N e N não é um dos valores a serem testados. Além disso, obter:

- a quantidade, soma e média aritmética dos valores dentro do intervalo;
- a quantidade, soma e média aritmética dos valores fora do intervalo.

Exemplo:

valores lidos: 5 3 18 7 22 50
impressão: 3 está dentro do intervalo
18 está fora do intervalo
7 está dentro do intervalo
22 está fora do intervalo
50 está fora do intervalo
2 valores dentro do intervalo,
soma = 10 e média = 5
3 valores fora do intervalo,
soma = 90 e média = 30

78) Repare a seguinte característica do número 3025:

$$30 + 25 = 55 \quad \text{e} \quad 55^2 = 3025$$

Faça um programa que leia uma série de valores (números inteiros de 4 algoritmos, um de cada vez) e diga se possuem a mesma característica (repare que $3025 / 100 = 30$ c/ resto 25). O programa termina quando for lido um valor menor que 1000 ou maior que 9999.

79) Uma das maneiras de se conseguir a raiz quadrada de um número é subtrair do número os ímpares consecutivos a partir de 1, até que o resultado da subtração seja menor ou igual a zero. O número de vezes que se conseguir fazer a subtração é a raiz quadrada exata (resultado 0) ou aproximada do número (resultado negativo).

Exemplo: Raiz de 16

$$16 - 1 = 15 \quad 15 - 3 = 12 \quad 12 - 5 = 7 \quad 7 - 7 = 0$$

A raiz de 16 é 4.

80) Faça um programa que leia um conjunto de pedidos de compra e calcule o valor total da compra. Cada pedido é composto pelos seguintes campos:

- número do pedido;
- data do pedido (dia, mês, ano);
- preço unitário do item e
- quantidade.

O programa deverá processar novos pedidos

Lembre-se sempre

“A Sorte Ajuda aos Competentes”

Página 5 de 32

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

até que o usuário digite 0 (zero) como número do pedido.

81) Refaça o exercício anterior, sendo que só serão processados os pedidos cujo valor seja maior que R\$ 1000,00.

82) Refaça o exercício 1 de modo que sejam processados somente os pedidos realizados nos anos de 1994 e 1996. Ao final do processamento o programa deverá indicar qual ano teve maior volume de compras.

83) Faça um programa que leia um número indeterminado de cartões contendo cada um a idade de uma pessoa e o sexo (0 - masculino, 1 - feminino). O último cartão (que não entrará nos cálculos) contém o valor da idade igual a zero. Calcule e imprima a idade média destas pessoas, quantas pessoas do sexo feminino estão na faixa dos 18 aos 25 anos e o número total de pessoas do sexo masculino.

84) Faça um programa que leia o preço de compra e o preço de venda de 100 mercadorias. O programa deverá imprimir quantas mercadorias proporcionam:

- lucro < 10 %
- 10 % <= lucro <= 20 %
- lucro > 20 %

85) Faça um programa que leia as seguintes informações sobre uma turma:

- número da turma;
- número de aulas dadas;
- número de alunos inscritos.

Deve ser lido também para cada aluno inscrito o seu número de faltas.

O programa deve imprimir o percentual de faltas para cada aluno e ao final do processamento o total de alunos reprovados por falta (25% de faltas).

86) Faça um programa que leia o salário bruto dos 500 funcionários de uma empresa, calcule o imposto retido e o salário líquido para cada um deles. Ao final do relatório o programa deverá imprimir os totais de salário bruto, salário líquido e imposto retido.

O imposto retido deverá ser calculado de acordo com a seguinte tabela:

Faixa de salário	Imposto
salário < 1000,00	0
1000,00 <= salário < 10000,00	10 %
10000,00 <= salário < 25000,00	15 %
salário >= 25000,00	20 %

87) Faça um programa que leia um conjunto de valores inteiros positivos e cujo último valor é -1. Dentre os valores lidos, o programa deve imprimir:

- o menor valor dentre os maiores que 100 e menores que 1000;
- a média desses valores dentre os maiores que 100 e menores que 1000;
- a soma desses valores dentre os maiores que 100 e menores que 1000;
- a soma de todos os valores lidos.

Observações:

- o valor -1 não deve ser considerado para nenhum cálculo.
- se nenhum valor estiver dentro do intervalo, o programa deve imprimir apenas uma mensagem para o usuário explicando o ocorrido.

88) As disciplinas do ciclo básico do nosso curso possuem códigos 108X, onde X pode variar de 0 a 9. Faça um programa que leia, para cada aluno, a sua matrícula (um inteiro de 1 a 2000), o código da disciplina, o código da turma (de 1 a 8) e a média final do aluno em cada disciplina. O programa deve imprimir, para a disciplina LTP1:

- o número de alunos da turma 1;
- a melhor média de alunos das turmas 1, 2, 3, 4 e 5;
- o total das médias maiores que 7 das turmas 6, 7, e 8.

Observações:

- A disciplina LTP1 possui código 1081 e 8 turmas.
- Considere um total de 2000 alunos e que todos os alunos cursaram todas as disciplinas (10).
- Os dados de um aluno serão digitados em sequência, ou seja, será lida a matrícula do aluno e, em seguida, todas as informações sobre ele.

Exemplo:

Lembre-se sempre
“A Sorte Ajuda aos Competentes”

Página 6 de 32

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

Matrícula do aluno: 33

Código da disciplina: 1086

Código da turma: 5

Média Final: 8

Código da disciplina: 1082

Código da turma: 1

Média Final: 6

... ..

Código da disciplina: 1085

Código da turma: 4

Média Final: 3

89) Faça um programa que calcule e imprima o CR do período para os alunos do P15. Para cada aluno, o programa deverá ler:

- número da matrícula;
- quantidade de disciplinas cursadas;
- notas em cada disciplina.

Além do CR de cada aluno, o programa deve imprimir o melhor CR dos alunos que cursaram 5 ou mais disciplinas.

Observações:

- o fim da entrada de dados é marcado por uma matrícula inválida (matrículas válidas: de 1 a 5000);
- o CR do aluno é igual a média aritmética de suas notas.

90) Faça um programa que controle o saldo bancário de um cliente. O programa lê o valor do saldo anterior e em seguida lê as operações realizadas na conta. As operações podem ser as seguintes:

- Saque em dinheiro (código 10);
- Depósito (código 33);
- Pagamento de cheque (código 4).

O programa lê o código das operações e realiza as atualizações na conta, imprimindo uma mensagem ao usuário caso seu saldo se torne negativo.

O programa deve continuar a leitura até que o código da operação seja zero. Códigos diferentes dos definidos devem ser ignorados.

Ao final do processamento o programa deve imprimir o saldo atual do cliente.

91) Uma biblioteca deseja fazer uma análise sobre a consulta a seu acervo. Faça um programa que leia o código de um livro, o número de exemplares e para cada um deles o número de empréstimos. O programa deve imprimir o total parcial de empréstimos para cada livro, e ao final deve imprimir o código do livro mais emprestado pela biblioteca. A entrada de dados termina com um código de livro negativo.

92) Uma empresa classifica seus funcionários em 3 níveis de acordo com um índice de produtividade. São eles: (1) Excelente, (2) Bom e (3) Regular. Cada nível acrescenta um abono ao salário base do funcionário, de acordo com a seguinte tabela:

Excelente	80 % do salário base
Bom	50 % do salário base
Regular	30 % do salário base

O programa deve ler a matrícula do funcionário, seu salário base, seu nível de abono e imprimir o salário a ser pago. O programa deve fornecer também a matrícula do funcionário de maior abono e a média do abono para os funcionários classificados como “Regular”.

O programa termina ao ler um valor de matrícula negativo.

93) A TELERJ deseja calcular as contas telefônicas de seus assinantes através do computador. A cobrança de seus serviços é feita da seguinte maneira:

1. Tarifa básica

- telefone residencial (código 1): R\$ 7,00
- telefone comercial (código 2): R\$ 20,00

2. Serviço local:

- R\$ 12,00 por pulso excedente (acima de 90 pulsos)

3. Serviço despertador:

- R\$ 5,00 por vez.

Na entrada de dados teremos:

- código do assinante;
- tipo do telefone (comercial ou residencial);
- número de pulsos registrados para chamadas locais e

Lembre-se sempre

“A Sorte Ajuda aos Competentes”

Página 7 de 32

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

- número de serviços de despertador prestados.

Faça um programa que leia os dados de um conjunto de assinantes (o código do assinante igual a zero encerra a entrada de dados), calcule e imprima:

- para cada assinante, o total de sua conta;
- valor da maior conta e o código do assinante que a pagou;
- o valor médio arrecadado por assinante no mês.

94) Faça um programa que ajude o DETRAN a saber o total de recursos que serão arrecadados com a aplicação de multas de trânsito.

O programa deve ler as seguintes informações para cada motorista:

- o número da carteira de motorista (de 1 a 4327);
- o número de multas;
- o valor de cada uma das multas.

Deve ser impresso o valor da dívida para cada motorista e ao final da leitura o total de recursos arrecadados (somatório de todas as multas). O programa deverá imprimir também o número da carteira do motorista que obteve o maior número de multas.

O programa termina ao ler a carteira de motorista de valor 0. Valores inválidos de carteira de motorista não devem ser processados, embora suas multas devam ser lidas.

95) Faça um programa que implemente o jogo da senha (para 2 pessoas):

- a) o jogador 1 digita uma senha (valor inteiro entre 0 e 100) sem o conhecimento do jogador 2
- b) o jogador 2 tem 5 chances para descobrir a senha
- c) a cada tentativa do jogador 2, o programa deve avisar se o valor digitado é maior, menor ou igual a senha
- d) se o jogador 2 acertar a senha, o programa não deve pedir mais nenhuma tentativa

96) Faça as seguintes alterações no jogo da senha (uma de cada vez):

- a) se o valor digitado em uma tentativa tiver uma diferença igual a 1 para a senha, o programa deve avisar que “TÁ QUENTE!”. Exemplos:
 - senha 43 e valor digitado 42

- senha 43 e valor digitado 44

b) ao final do jogo, se for o caso, enviar a mensagem “Você perdeu. Tente novamente depois”

c) não permita que o jogador 1 digite valores fora da faixa esperada (entre 0 e 100)

d) ao final de uma partida, permita ao usuário jogar novamente

97) (Prova 1 - 95/2) Faça um programa que calcule quantos litros de gasolina são usados em uma viagem, sabendo que um carro faz 10 Km/litro. O usuário fornecerá a velocidade do carro e o período de tempo que viaja nesta velocidade, para cada trecho do percurso. Então, usando as fórmulas $\text{distância} = \text{tempo} \times \text{velocidade}$ e $\text{litros consumidos} = \text{distância} / 10$, o programa computará, para todos os valores não negativos de velocidade, os litros de combustível consumidos. O programa deverá imprimir a distância e o número de litros de combustível gastos naquele trecho. Deverá imprimir, também, o total de litros gastos na viagem. O programa pára quando for digitado um valor negativo de velocidade.

98) Escreva um algoritmo que leia valores de venda de discos, CD's e fitas K7's de uma loja em um dia e desconte a comissão do vendedor (3%) e o repasse ao fornecedor (50%) para gerar o ganho da loja na venda de cada peça. O programa deverá gerar uma listagem contendo o valor da venda, a comissão, o repasse e o ganho da loja para cada venda, totalizando os valores ao final. Os dados da entrada terminam com o valor de venda igual a 0 (zero).

99) (Prova 1 - 96/2) Faça um programa que calcule o imposto de renda de um grupo de contribuintes, considerando que:

- a) os dados de cada contribuinte (CIC, número de dependentes e renda bruta anual) serão fornecidos pelo usuário via teclado;
- b) para cada contribuinte será feito um abatimento de R\$ 600 por dependente;
- c) a renda líquida é obtida diminuindo-se o abatimento com os dependentes da renda bruta anual;
- d) para saber quanto o contribuinte deve pagar de imposto, utiliza-se a tabela abaixo:

RENDA LÍQUIDA
até R\$ 1000

IMPOSTO
0% (isento)

Lembre-se sempre

“A Sorte Ajuda aos Competentes”

Página 8 de 32

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

de R\$ 1001 a R\$ 5000 15%
 acima de R\$ 5000 25%

- e) o valor de CIC igual a zero indica final de dados;
 f) o programa deverá imprimir, para cada contribuinte, o número do CIC e o imposto a ser pago;
 g) ao final, o programa deverá imprimir o total do imposto arrecadado pela Receita Federal e o número de contribuintes isentos;
 h) leve em consideração o fato do primeiro CIC informado poder ser zero.

100) (Prova 1 - 96/1) Responda as questões abaixo, baseado no algoritmo:

```
#include <stdio.h>
void main (void)
{
    int x, w, y, z;
    scanf ("%d%d%d%d", &w, &x, &y, &z);
    if (w>=x)
    {
        while (y>0)
        {
            y = y - 1;
            if (y>w)
            {
                printf ("local 1");
                while (z<11)
                {
                    z = z + 3;
                    if (z>x)
                        z = 15;
                    printf ("local 2");
                }
                printf ("local 3");
            }
        }
        printf ("local 4");
        while (w>x)
        {
            w = w - 5;
            printf ("local 5");
        }
    }
}
```

Escreva exatamente o que será impresso pelo algoritmo acima para cada entrada abaixo:

a) 2, 4, -15, -3

- b) 7, 2, 4, 0
 c) 3, 3, 5, 0
 d) -1, -3, 4, 0

101) (Prova 1 - 96/2) Faça um programa que leia um conjunto de notas de prova (0 a 10) de um conjunto de turmas. O programa deverá ler a quantidade de turmas e, para cada turma:

- o código da turma (inteiro positivo)
 - a quantidade de notas a serem lidas (inteiro positivo)
 - as notas da turma (inteiro positivo)
- O programa deverá imprimir:
- a média aritmética das notas de cada turma
 - a maior nota de todas as turmas e o código da turma a que ela pertence

102) (Prova 2 - 94/1) Sabe-se que o produto de dois números inteiros pode ser obtido através de sucessivas multiplicações (por 2), divisões (por 2) e somas. Enquanto um dos números é dividido por 2, o outro é multiplicado por 2. Quando o resultado da divisão for ímpar, o outro número correspondente (resultado da multiplicação) é acumulado; caso contrário, passa-se para a próxima multiplicação e divisão. Quando o resultado da divisão for 0 (zero), o valor acumulado corresponderá ao resultado final.

Exemplo:

Primeiro número	Segundo número	Valor acumulado
37 ímpar	41	41
37:2=18 par	41x2=82	41
18:2=9 ímpar	82x2=164	41+164=205
9:2=4 par	164x2=328	205
4:2=2 par	328x2=656	205
2:2=1 ímpar	656x2=1312	205+1312=1517
1:2=0 fim		Resultado = 1517

a) Faça uma função que receba como parâmetros 3 números inteiros e retorne a soma do segundo com o terceiro, caso o primeiro seja ímpar; senão, retorna o terceiro.

b) Usando a função do item a), faça um programa que leia dois números inteiros e calcule seu produto através do método especificado.

Linguagem de Programação C

103) (Prova 1 - 97/1) Faça um programa em “C” que calcule e imprima a média e a variância das notas de todos os alunos inscritos em LTP1. As notas devem ser lidas via teclado. O fim da leitura é marcado por uma nota negativa. A média e a variância são calculadas da seguinte forma:

$$media = \frac{\sum notas}{numero_de_alunos}$$

$$variancia = \frac{\sum (notas^2)}{numero_de_alunos} - media^2$$

Observar que $\sum(notas^2) \neq (\sum notas)^2$!!!!!!!

104) (Prova 2 - 94/2) Devido às confusões dos programadores do TRE, a Justiça Eleitoral resolveu contratá-lo para realizar a apuração das eleições.

Faça um programa que leia o código dos candidatos e informe o número de votos recebidos por cada um, bem como o candidato mais votado.

Você sabe que 100 candidatos disputam uma vaga. E que a entrada termina assim que for lido um candidato de código negativo.

Exemplo:

Entrada: 0 0 1 2 2 -1

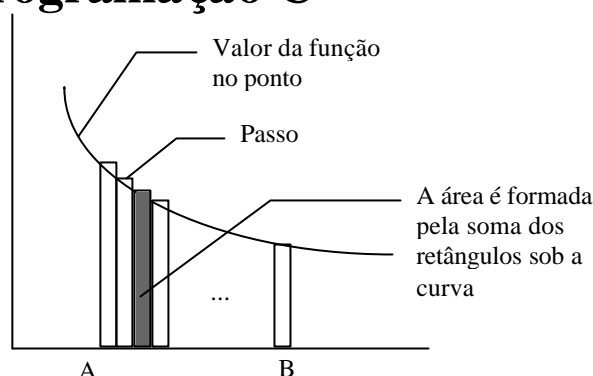
Saída: 0 3 votos

1 1 voto

2 2 votos

O candidato 0 foi eleito.

105) (Prova 2 - 96/1) Faça um programa em linguagem C que calcule o valor da área sob uma curva $y = 1/x$ entre dois pontos. Tais pontos deverão ser fornecidos pelo usuário assim como o passo. O valor da integral indica a área sob a curva, entre duas coordenadas X. Por exemplo, seja o gráfico a seguir:



106) (Prova 1 - 97/1) Faça um programa em “C” que permita a uma empresa atacadista de cimento controlar os pedidos de compra e o estoque do produto. O programa inicialmente irá ler do teclado a quantidade de sacos de cimento disponíveis no estoque da empresa (tipo inteiro) e o preço de cada saco (tipo real). Em seguida, o programa processará os pedidos de compra dos clientes da empresa.

As informações a serem lidas do teclado, para cada pedido, são as seguintes:

- código do cliente (tipo inteiro)
- quantidade de sacos de cimento a serem comprados (tipo inteiro)

Um pedido só poderá ser aceito se a quantidade de sacos disponíveis no estoque for maior ou igual à quantidade de sacos do pedido e se a quantidade de sacos do pedido não ultrapassar 10% (dez por cento) do total de sacos disponíveis no estoque.

Se o pedido for aceito, o programa deverá imprimir o código do cliente, a quantidade de sacos pedidos e o valor do pedido, além de subtrair do estoque a quantidade de sacos vendidos.

Se o pedido for rejeitado, as mensagens Estoque insuficiente ou Ultrapassado o máximo permitido deverão ser exibidas no vídeo, conforme o motivo da rejeição do pedido.

Ao final dos pedidos, que será detectado quando o código do cliente for zero ou o estoque mínimo de cem sacos for atingido, deverá ser impressa a quantidade de sacos em estoque.

Linguagem de Programação C

107) (Prova 1 - 97/1) Faça um programa em “C” que gerencie as contas correntes dos 1000 clientes do Banco Oir Cup, um banco ítalo-anglo-franco-luso-nipo-brasileiro. O programa deverá ler, para cada cliente, o código do cliente (tipo inteiro), saldo anterior (tipo real) e as movimentações da conta. Cada movimentação é composta por um código (tipo caractere, **C**, **D** ou **F**, indicando **Crédito**, **Débito** ou **Fim** das movimentações deste cliente) e um valor (tipo real). Deverá ser impresso, para cada cliente, o seu código e o saldo atual (após o processamento das movimentações). Ao final, deverá ser impresso o total de dinheiro em caixa no banco (soma dos saldos de todos os clientes) e o código do cliente que possui o maior saldo.

Observação: o menor saldo possível é de R\$ 1000 negativo.

Seleção múltipla

Exercícios de Fixação

108) Faça um programa que implemente uma máquina de calcular com operações +, -, *, / e %, através da leitura de dois operandos numéricos inteiros e do operador (tipo caractere), no formato: operando1 operador operando2. Verifique casos de operações que não podem ser realizadas (antes de operá-las).

109) Escreva um programa que leia um peso na Terra e o número de um planeta e imprima o valor do seu peso neste planeta. A relação de planetas e dada a seguir juntamente com o valor das gravidades relativas em relação à Terra.

#	Gravidade relativa	Planeta
1	0,37	Mercúrio
2	0,88	Vênus
3	0,38	Marte
4	2,64	Júpiter
5	1,15	Saturno
6	1,17	Urano

Para calcular o peso no planeta use a fórmula:

$$P_{planeta} = \frac{P_{Terra}}{10} * g_{planeta}$$

110) Faça um programa que, dada a idade de um nadador, classifique-o em uma das seguintes categorias:

Categoria	Idade
Infantil A	5 a 7 anos
Infantil B	8 a 10 anos
Juvenil A	11 a 13 anos
Juvenil B	14 a 17 anos
Senior	maiores de 18 anos

111) Em uma eleição presidencial, existem quatro candidatos. Os votos são informados através de código. Os dados utilizados para a escrutinagem obedecem à seguinte codificação:

- 1, 2, 3, 4 = voto para os respectivos candidatos;
- 5 = voto nulo;
- 6 = voto em branco;

Elabore um programa que calcule e imprima:

- total de votos para cada candidato;
- total de votos nulos;
- total de votos em branco;
- percentual dos votos em branco e nulos sobre o total;
- situação do candidato vencedor sobre os outros dois, no caso, se ele obteve ou não mais votos que os outros dois somados.

Como finalizador do conjunto de votos, tem-se o valor 0.

112) Faça um programa que calcule a área de determinadas figuras geométricas. O programa deverá apresentar um menu com as seguintes opções:

1. quadrado ($lado^2$)
2. retângulo ($comprimento * largura$)
3. círculo ($3,14 * raio^2$)
4. trapézio ($(base maior + base menor) * altura / 2$)

De acordo com a opção, o programa deve pedir os

Lembre-se sempre

Página 11 de

Linguagem de Programação C

valores necessários para realizar o cálculo. Utilize o comando **switch**.

113) Faça o chinês do programa abaixo, mostrando exatamente o que seria impresso se fossem digitados os seguintes caracteres na entrada: EVA e ADAO#

```
#include <stdio.h>
void main (void)
{
    char ch;
    int i = 0;

    while ((i<=10) && (ch=getch())!='#')
    {
        switch (ch)
        {
            case 'A':
            case 'O':
                i++;
                putchar ('E');
                break;

            case ' ':
                putchar ('!');

            case 'L':
                i = i +2;
                putchar ('$');
                break;

            default:
                putchar ('?');
                i++;

        }
    }
    printf ("\nAcabou");
}
```

114) (Prova 1 - 95/2) Uma equipe de vôlei paga aos seus jogadores, além do salário, um valor adicional ao salário mensal (bicho) que é função da produtividade de cada um.

Essa produtividade é paga de acordo com a tabela a seguir:

classe	nível	valor adicional
1	excelente	+100%
2	bom	+80%
3	médio	+50%
4	regular	+30%
5	precisa treinar mais	+10%
6	te cuida	+5%
7	tsktsk	nada

Faça um programa que, dados o salário e o código da classe de um jogador, calcule e imprima o seu salário final e o nome da sua classe (nível).

115) (Prova 3 - 96/1) Um aluno de LTP1 escreveu esta função em uma das listas. Ajude o professor a corrigir (como você gostaria que eu corrigisse a sua prova) apontando os sete erros. Esta função implementa uma calculadora, retornando -1 se a operação não puder ser realizada ou 0 (zero) em caso contrário. O resultado da operação será armazenado em *op1. Rescreva corretamente apenas as linhas erradas, identificando-as pelo seu número.

```
1. void calc (float op1, float op2,
2.           char *operador)
3. {
4.     int aux;
5.     while (operador)
6.     {
7.         case '+':
8.             *op1 = *op1 + op2;
9.             break;
10.        case '-':
11.            *op1 = *op1 - op2;
12.            break;
13.        case '*':
14.            *op1 = *op1 * op2;
15.            break;
16.        case '/':
17.            if (*op2 != 0)
18.                *op1 = *op1 / op2;
19.            else
20.                aux = -1;
21.        return aux;
22.    }
23. }
```

Linguagem de Programação C

Caracteres

Exercícios de Fixação

116) Faça um programa que imprima as letras minúsculas do alfabeto e seus respectivos códigos decimais da tabela ASCII.

117) Faça um programa que leia uma sequência de caracteres terminada por '#' e imprima o número de caracteres lidos.

118) Faça um programa que leia uma sequência de caracteres terminada por um <Enter> (linha de texto) e imprima a quantidade de caracteres lidos.

119) Faça um programa que leia uma sequência de caracteres terminada por um <Enter> (linha de texto) e os imprima novamente na tela com exceção dos caracteres de espaço em branco que não devem ser impressos.

120) Faça um programa que leia uma sequência de caracteres terminada por um <Enter> (linha de texto) e imprima a quantidade de vogais, a quantidade de consoantes e a quantidade de caracteres não letras encontrados. Considere apenas as letras minúsculas.

121) Faça um programa que leia uma sequência de caracteres terminada por um <Enter> (linha de texto) e imprima a quantidade total de caracteres lidos e a porcentagem de caracteres do tipo letra encontrados.

122) Faça um programa que leia uma linha de texto da entrada e imprima o número de letras maiúsculas e o número de letras minúsculas encontradas.

123) Durante a segunda guerra mundial, havia necessidade de codificação das mensagens transmitidas, para isso eram necessários códigos secretos, um desses códigos era o POLAR/ZENIT, que consistia em trocar as ocorrências da letra P pela letra Z e vice-versa, as letras O pelas letras E, e vice-versa, e assim por diante. Faça um programa que leia uma linha de texto da entrada e imprima os

caracteres codificados segundo o código POLAR/ZENIT.

124) Faça um programa que leia uma linha de texto e imprima o número de palavras que compõe a linha. Considere que cada palavra esta separada por exatamente um único caracter de espaço.

125) Faça um programa que leia um caracter e uma linha de texto e imprima a posição do caracter na linha a cada vez que ele for digitado. O caracter pode aparecer mais de uma vez na linha.

126) Faça um programa que leia uma sequência de linhas de texto (um texto) terminada com uma linha vazia e imprima a quantidade total de caracteres lidos.

127) Faça um programa que leia uma sequência de linhas de texto terminada com uma linha vazia e imprima a quantidade de caracteres da maior linha do texto.

128) Faça um programa que leia uma sequência de linhas de texto terminada com uma linha vazia e imprima a quantidade média de caracteres encontrados em cada linha.

129) Faça um programa que procure por um determinado caracter em um texto (conjunto de linhas), lido da entrada, indicando os números das linhas em que o caracter aparecer e também a respectiva posição em cada linha (em número de caracteres). O caracter pode aparecer mais de uma vez em uma linha.

130) Faça um programa que leia uma sequência de linhas de texto e imprima os caracteres lidos novamente na tela com exceção dos caracteres de espaço em branco, que não devem ser impressos.

131) Faça um programa que leia um texto (conjunto de linhas) e imprima a quantidade de vogais, a quantidade de consoantes e a quantidade de caracteres não letras encontrados. Considere apenas as letras minúsculas.

Linguagem de Programação C

132) Faça um programa que leia um texto da entrada e converta os caracteres maiúsculos em minúsculos.

133) Faça um programa que leia um texto da entrada e converta os caracteres maiúsculos para minúsculos e vice-versa.

134) Faça um programa que leia três caracteres numéricos ('0' a '9') da entrada e imprima o número inteiro correspondente usando **printf** com formato **"%d"**. Lembre-se que você deve compor o número, o primeiro caractere será a centena, o segundo a dezena e o terceiro a unidade.

135) Faça um programa que leia um texto da entrada e imprima o número de palavras encontradas no texto.

Exercícios Propostos

136) Faça um programa que calcule o resultado de uma expressão aritmética lida da entrada. A expressão terá um número indeterminado de termos envolvendo as operações de soma, subtração, divisão e multiplicação. **Não leve em consideração a precedência dos operadores. É proibido o uso da função scanf para a leitura dos operadores e operandos.** Exemplo: $10 + 5 * 2 = 30$

137) Faça um programa em "C" que leia quatro caracteres da entrada padrão (representado um número na base 16) e um caractere representando um sinal e imprima o valor numérico decimal correspondente. O caractere que representa o sinal ('+' ou '-') pode aparecer em qualquer parte do número. Os demais caracteres serão dígitos ('0', '1', '2', ..., '9') ou letras maiúsculas ('A', 'B', 'C', 'D', 'E', 'F'). Para facilitar assumo que o usuário irá digitar exatamente o que o programa espera ler.

Exemplo: 110-B \Rightarrow -4363

138) (Prova 4 - 96/1) Faça um programa que leia uma linha (terminada por <Enter>) com um texto compactado e a imprima de forma descompactada. Ao aparecer no texto o caractere '@', significa que os dois caracteres seguintes são, respectivamente, o

número de vezes que o caractere se repete e o caractere a ser repetido.

Exemplo:

texto compactado: *x@3ux@5a*
texto descompactado: *uuuuxaaaaa*

139) (Prova 2 - 94/1) Faça um programa que leia da entrada padrão (teclado) uma sequência de caracteres, terminada pelo caractere '#' e imprima quantos caracteres, quantas palavras e quantas frases foram lidas.

Observações: Para facilitar, considere que: cada frase é delimitada por '?', '!' ou '!'; cada palavra é delimitada por '\n', '\t' ou espaço; não existem espaços antes e depois de um sinal de pontuação; cada palavra é separada das demais por um único espaço; o texto começa por uma letra; não existem sinais de pontuação consecutivos.

140) (Prova 3 - 96/2) O programa abaixo funciona como esperado? Justifique sua resposta.

```
#include <stdio.h>
#include <conio.h>
void main (void)
{
    int i;
    char nome[10];
    for (i = 0; i < 10; i++)
        nome[i] = getch();
    puts (nome);
}
```

141) (Prova 1 - 97/1) Faça um programa em "C" que leia uma linha de texto e substitua cada vogal lida por N vogais iguais, onde N representa o número de vogais **lidas** até então. Para os demais caracteres, o programa deve imprimir apenas o próprio caractere. Considere que o usuário digitará apenas caracteres minúsculos. O programa deverá imprimir, ao final, o percentual de consoantes em relação ao total de caracteres digitados. Exemplo:

Texto lido: apenas um exemplo!!!!!!

Impressão: apeenaaas uuuum
eeeeexeeeeeemploooooo!!!!!!

Percentual: 33.333333

Lembre-se sempre

32

"A Sorte Ajuda aos Competentes"

Página 14 de

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

142) (Prova 4 - 96/2) Faça um programa que compacte um texto lido segundo o seguinte algoritmo:

- a) se houver de 3 a 9 caracteres repetidos, o programa deverá imprimir o caracter '@' seguido do número de caracteres repetidos e do caracter repetido,
- b) se não houver mais do que três repetições, o próprio caracter deverá ser impresso tantas vezes quanto ele apareceu,
- c) se houver mais do que 9 repetições, o programa deverá imprimir seqüências semelhantes a do item a até totalizar o número de repetições necessárias (ver exemplo da letra k).

Exemplo:

Entrada de dados:

abcccdeeeee
fghhhhhhhhhh
ijklkkkkkkkkkkklm

Saída de dados:

ab@3cd@5e
fg@9hh
ij@9k@6klm

143)

- a) Faça uma função que leia um valor, em algarismos romanos (considere válidos apenas os algarismos I, V e X), e o converta para seu número decimal correspondente, devolvendo o valor calculado. O número em romanos será lido como uma seqüência de caracteres terminada com um <Enter>.
- b) Faça uma função que receba 2 valores decimais e um operador (+, -, *, /) e calcule o resultado da operação.
- c) Faça um programa, utilizando as funções dos itens anteriores, que leia 2 (dois) números em algarismos romanos e um operador (+, -, *, /) e calcule o resultado da operação.

Funções

Exercícios de Fixação

144) Faça uma função que:

- a) receba um inteiro (de um dígito) como parâmetro e retorne o caractere correspondente.
- b) receba um caractere como parâmetro e retorne o seu correspondente em maiúsculo (se minúsculo) ou o próprio caractere.
- c) receba um caractere como parâmetro e retorne o seu correspondente em minúsculo (se maiúsculo) ou o próprio caractere.
- d) receba um caractere como parâmetro e retorne o seu correspondente código ASCII (inteiro).
- e) receba um caractere como parâmetro e retorne verdadeiro (1) se for letra e falso (0), caso contrário.
- f) receba um caractere como parâmetro e retorne verdadeiro (1) se for dígito e falso (0), caso contrário.
- g) receba um caractere como parâmetro e retorne verdadeiro (1) se for espaço e falso (0), caso contrário.
- h) receba um caractere como parâmetro e retorne verdadeiro (1) se for letra maiúscula e falso (0), caso contrário.
- i) receba um caractere como parâmetro e retorne verdadeiro (1) se for letra minúscula e falso (0), caso contrário.
- j) retorne o resultado de um número “elevado” a outro (ambos recebidos como parâmetros).
- k) retorne o resto da divisão inteira de um número por outro.

145) Faça uma função que receba um caracter como parâmetro e retorne 1 caso seja uma consoante e 0 em caso contrário.

146) Faça uma função que receba um caracter como parâmetro e retorne 1 caso seja uma vogal e 0 em caso contrário.

147) Faça uma função que receba um número inteiro como parâmetro e retorne seu valor absoluto.

148) Faça uma função que receba um número inteiro como parâmetro e retorne 1 se sua raiz quadrada é exata e 0 em caso contrário. A função de retornar, através de um parâmetro passado por referência, o valor aproximado da raiz.

Linguagem de Programação C

149) Faça uma função que receba um caracter como parâmetro por referência. Se o caracter for uma letra minúscula, a função deve convertê-la para maiúscula. A função deve retorna 1 se o caracter for uma letra e 0 em caso contrário.

150) Faça uma função que receba um caracter como parâmetro por referência. Se o caracter for um dígito, a função deve convertê-lo para o seu respectivo valor numérico. Lembre-se '0' caracter é diferente de 0 numérico. A função deve retorna 1 se o caracter for um dígito e 0 em caso contrário.

151) (Prova 3 - 96/2) Faça os protótipos (**apenas!!**) das funções f(), g() e h() utilizadas pela função main() abaixo:

```
void main (void)
{
    int v[6]={0,10,20,40,50,60}, y;
    float x, z;
    x = f (10);
    g (x, &y);
    g (z, &v[4]);
    z = h (v[5], v);
    ...    ...    ...
}
```

152) (Prova 3 - 97/1) Faça os protótipos (**apenas!!**) das funções abaixo, baseado nas informações dadas:

a)

```
void main (void)
{
    int a, b[10];
    char d, e[10];
    float g, h[10];

    ...    ...    ....
    faz_tudo (a, &d, &g);
    g = faz_nada (a, b[1], b);
    não_sei (3.14, g, h[2], &h[3]);
    e[a] = ta_certo ();
    ...    ...    ....
}
```

b)

```
void main (void)
{
    int v[6]={0, 10, 20, 40, 50, 60},y;
    float x, z;

    ...    ...    ...
    x = f (10);
    g (x, &y);
    g (z, &v[4]);
    z = h (v[5], v);
    ...    ...    ...
}
```

153) (Prova 3 - 97/1) Faça as funções necessárias para que os programas abaixo funcionem conforme o esperado:

a)

```
#include <stdio.h>
void main (void)
{
    int a, b, c, erro;
    scanf ("%d%d", &a, &b);
    c = divide (a, b, &erro);
    if (erro)
        printf ("Erro: divisão por zero\n");
    else
        printf ("%d dividido por %d = %d\n", a, b, c);
}
```

b)

```
#include <stdio.h>
void main (void)
{
    int v[5]={1, 2, 3, 4, 5}, i, j;
    printf ("\nDigite as posições do vetor que devem ser trocadas");
    scanf ("%d%d", &i, &j);
    troca (v, i, j);
    for (i=0; i<5; i++)
        printf ("%d\n", v[i]);
}
```

Exercícios Propostos

154) Elabore uma função que retorne o reverso de um número inteiro, por exemplo 932 → 239.

Lembre-se sempre

Página 16 de

Linguagem de Programação C

155)

- a) Faça uma função chamada *fatorial* que deve receber um valor inteiro como parâmetro e retornar o fatorial deste número.
- b) Faça um programa que calcule o valor da combinação de n elementos r a r . A fórmula da combinação é a seguinte: $C_n^r = \frac{n!}{r! * (n-r)!}$. Leia os valores de n e r da entrada e use a função do item anterior para calcular o resultado.

156) Construa uma função que, dado um número de conta corrente com três dígitos, retorne o seu dígito verificador, o qual é calculado da seguinte maneira:

Exemplo: número da conta: 25678

- Somar o número da conta com o seu inverso: $256 + 652 = 908$;
- multiplicar cada dígito pela sua ordem posicional e somar este resultado: 9 0 8

$$\begin{array}{r} 9 \quad 0 \quad 8 \\ \times 1 \quad \times 2 \quad \times 3 \\ \hline 9 \quad + \quad 0 \quad + \quad 24 = 33 \end{array}$$

- o último dígito deste resultado é o dígito verificador da conta ($33 \rightarrow 3$).

157) Faça uma função que calcule o peso ideal de uma pessoa. A função deve receber o valor da altura da pessoa e um valor numérico indicando o sexo (0 - masculino e 1 - feminino). A função deve calcular o peso ideal de acordo com as seguintes fórmulas: homens $\Rightarrow (72.7 * \text{altura}) - 58$ e mulheres $\Rightarrow (62.1 * \text{altura}) - 44.7$. Faça a função *main* para completar o seu programa. A função *main* deve ler a altura e o sexo de um conjunto de pessoas e imprimir o peso ideal até que seja digitado um valor negativo para altura.

158)

- a) Faça uma função (chamada **conv**) que leia um dígito (caractere) e retorne o valor numérico correspondente.
- b) Faça uma função (chamada **le_converte**) que leia 5 dígitos (caracteres), representando uma sequência de bits, e retorne o valor numérico

decimal correspondente. É obrigatório o uso da função anterior. Você pode fazer esta questão mesmo que não tenha feito a anterior.

- c) Seja a função **main** abaixo:

```
void main (void)
{
    int a, b, c;
    a = le_converte();
    b = le_converte();
    c = a + b;
    printf ("\n%d", c);
    imprime_em_binario (c);
}
```

Faça a função **imprime_em_binario**.

159) (Prova 2 - 96/1) Crie as funções necessárias ao funcionamento correto do programa abaixo. Não é permitido:

- alterar a função *main*();
- usar *printf*();
- usar *scanf*();

Linguagem de Programação C

```
void main (void)
{
    int x, y, z;
    int n;
    scanf ("%d", &n);
    x = funcao_q_le (n);
    y = funcao_q_le (n);
    z = x + y;
    funcao_q_escreve (z, n+1);
}
```

a) a função `funcao_q_le(i)` deverá ler **i** caracteres e retornar o seu valor decimal.

b) a função `funcao_q_escreve (a, i)` deverá escrever o valor **a** de até **i** dígitos.

160) O número irracional π pode ser calculado através da expressão $\pi = 4 \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1}$. Construa uma

função em C que receba como parâmetro (**passagem por valor**) o número de termos a serem usados no cálculo de π , e retorne o valor calculado para π .

161) As raízes de uma equação do segundo grau da forma $ax^2 + bx + c = 0$ podem ser calculadas através da expressão $x = \frac{-b \pm \sqrt{\Delta}}{2a}$, onde

$\Delta = b^2 - 4ac$. Faça um programa completo em C que leia um número indeterminado de coeficientes **a**, **b** e **c** do teclado (o programa deverá terminar quando o valor do coeficiente **a** for **zero**) e, para cada grupo de coeficientes, chame uma **função** que calcule as raízes da equação e as exiba no monitor, caso elas sejam reais, ou a mensagem **Raízes Imaginárias**, caso elas sejam imaginárias.

A função deverá receber como parâmetros os coeficientes **a**, **b** e **c** (passagem por valor) e as raízes **x1** e **x2** (passagem por referência). Os valores devolvidos pela função através dos parâmetros serão os seguintes:

a) se $\Delta > 0$, **x1** e **x2** conterão duas raízes reais e distintas;

b) se $\Delta = 0$, **x1** e **x2** conterão duas raízes reais e

iguais;

c) se $\Delta < 0$, **x1** e **x2** não deverão ser alteradas pela função.

O valor de retorno da função será o valor de Δ , indicando a quem chamou a função a natureza das raízes (reais e distintas, reais e iguais ou imaginárias).

Obs: use a função `pow(x,y)` para calcular x^y , onde **x** e **y** são do tipo `float`. Lembre-se de que a raiz quadrada de um número real **x** é igual a $x^{0.5}$.

Vetores

Exercícios de Fixação

162) Faça um programa que leia e imprima o resultado da soma entre dois vetores inteiros de 50 posições.

163) Elabore um algoritmo que, dados dois vetores inteiros de 20 posições, efetue as respectivas operações indicadas por um outro vetor de 20 posições de caracteres também fornecido pelo usuário, contendo as quatro operações aritméticas em qualquer combinação, armazenando os resultados em um quarto vetor.

164) Construa um algoritmo que preencha um vetor de 100 elementos inteiros colocando 1 na posição correspondente a um quadrado perfeito e 0 nas demais.

165) Faça um programa que leia um conjunto de linhas de texto, terminadas com uma linha vazia e imprima a quantidade de cada uma das letras minúsculas encontradas no texto.

166) Faça um programa que leia uma linha de texto, terminada como um `<Enter>` e imprima o inverso da linha que foi lida.

Exemplo: Isto e apenas um exemplo
olpmexe mu sanepa e otsI

167) Faça uma função chamada *inverte*, que receba um vetor de números inteiros como parâmetro e seu tamanho. A função deve inverter a ordem dos

Lembre-se sempre

Página 18 de

Linguagem de Programação C

elementos do vetor, de modo que o primeiro elemento passe a ser o último, o segundo passe a ser o penúltimo e assim por diante. Faça um programa para testar a função *inverte*.

168) Faça um programa que leia um conjunto de linhas de texto e imprima a linha que teve o maior número de vogais. Considere que cada linha terá um máximo de 80 caracteres.

169) Faça um programa que leia uma sequência de 10 números e informe o total de ocorrências do último número lido.

Exemplo: supondo a seguinte sequência de números

38 4 23 5 6 7 4 12 4

o resultado será:

O número 4 apareceu 3 vezes

170) Faça um programa que leia um texto (sequência de linhas) terminado por uma linha vazia (uma linha com apenas um <Enter>) e informe o total de ocorrências de cada dígito (de '0' até '9'). Os dígitos que não ocorreram no texto não devem ser listados.

Exemplo: supondo a seguinte sequência de caracteres:

Este eh 1 exemplo.

Esta eh apenas a linha 2 do exemplo.

A linha 3 vem depois da linha 2.

O programa deve funcionar para muito mais do que 239 casos.

A saída será:

O dígito 1 apareceu 1 vez(es).

O dígito 2 apareceu 3 vez(es).

O dígito 3 apareceu 2 vez(es).

O dígito 9 apareceu 1 vez(es).

171) Faça um programa que leia um vetor ordenado de 10 números inteiros e uma chave, informe a posição da chave no vetor ou uma mensagem de erro se o número não for encontrado.

Obs.: faça uma função que realize a busca. Esta função não deverá imprimir nada.

Exemplo: supondo a seguinte sequência:

1 2 3 8 97 412 446 957 8912 8974

se a chave for 446, o resultado será: Chave na

posição 6.

se a chave for 413, o resultado será: Chave não encontrada.

172) Faça um programa que leia a matrícula e a nota final de 10 alunos de uma turma de LTP1 e imprima os dados dos alunos (matrícula e nota) ordenados por nota (ordem decrescente).

173) Faça uma função que receba um vetor de caracteres chamado *linha* e um caracter chamado *ch* como parâmetros. Retire todas as ocorrências do caracter *ch* do vetor *linha*. O fim do vetor *linha* será indicado pelo caracter '\n'. Faça um programa para testar a sua função.

174) Faça uma função que receba um vetor de caracteres, seu tamanho e um caracter e retire todas as ocorrências desse caracter no vetor. A função deve retornar o total de caracteres restantes no vetor. Faça uma main() para testar sua função, do jeito que você achar necessário.

175) Faça uma função que receba um vetor de inteiros (ordenados de forma crescente) e seu tamanho e retorne o maior valor menor que a média de seus elementos. Faça uma main() para testar sua função, do jeito que você achar necessário.

176) Faça um programa que leia um texto (sequência de linhas) da entrada e imprima as 5 letras minúsculas de maior ocorrência no texto. Dica: Faça um vetor de contadores, um para cada letra e ordene este vetor em ordem decrescente.

177) Faça uma função que receba um vetor de caracteres, seu tamanho e dois caracteres e substitua todas as ocorrências do 1º caracter pelo 2º. Faça uma main() para testar sua função, do jeito que você achar

178) Faça uma função que receba um vetor de caracteres, seu tamanho e um inteiro. Se o caracter da posição indicada pelo inteiro for letra, retornar o total de letras do vetor; senão, retornar o total de caracteres não-letra. Faça uma main() para testar sua função, do jeito que você achar necessário.

Lembre-se sempre

32

“A Sorte Ajuda aos Competentes”

Página 19 de

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

179) Faça uma função que receba um vetor de inteiros positivos, um vetor de caracteres e o tamanho (único) dos vetores. A função deve imprimir cada um dos caracteres do 2º vetor **n** vezes, onde **n** é o conteúdo da posição correspondente no vetor de inteiros. Faça uma main() para testar sua função, do jeito que você achar necessário.

180) Faça uma função que receba um vetor de inteiros e seu tamanho e retorne 1 (um) se o vetor estiver ordenado de forma crescente ou 0 (zero) se não estiver. Faça uma main() para testar sua função, do jeito que você achar necessário.

181) Faça uma função que receba dois vetores de inteiros e seus tamanhos e imprima os elementos comuns aos dois vetores. Faça uma main() para testar sua função, do jeito que você achar necessário.

182) Faça uma função que receba um vetor de inteiros e seu tamanho e retorne o menor dos elementos pares do vetor. Faça uma main() para testar sua função, do jeito que você achar necessário.

183) Faça uma função que implemente uma busca binária. Esta função recebe um vetor de caracteres, seu tamanho e uma chave e retorna a posição da chave no vetor. Faça uma main() em que, se a chave for encontrada no vetor, ela será substituída por um * (asterisco).

184) Faça uma função que receba um vetor de inteiros e seu tamanho e retorne 1 (um) se o vetor estiver ordenado de forma decrescente ou 0 (zero) se não estiver. Faça uma main() para testar sua função, do jeito que você achar necessário.

185) Faça uma função que receba como parâmetros dois **vetores** de **reais**, e um **inteiro** indicando a **quantidade** de elementos que os dois vetores têm. A função deverá retornar o produto interno dos dois vetores.

Exemplo:

X = (1,2,3,4)

Y = (4,3,2,1)

X.Y = 4 + 6 + 6 + 4 = 20

186) Faça uma função que receba um vetor de **n** números inteiros positivos e obtenha a maior diferença (em módulo) entre dois elementos consecutivos deste vetor. O valor da maior diferença deverá ser retornado pela função.

Exemplo:

V = (1,2,4,7,8)

A maior diferença em módulo é 3.

Exercícios Propostos

187) Faça um programa que leia um texto e uma chave, e embaralhe as letras do texto conforme a ordem dada pela chave.

Por exemplo:

texto: "abcdefghij"

chave: 31524

saída: "caebdhfjgi"

Ou seja, o texto é dividido em pedaços do tamanho da chave (nesse caso 5). E cada pedaço é embaralhado da seguinte forma:

em 1º lugar a 3ª letra

em 2º lugar a 1ª letra

em 3º lugar a 5ª letra

em 4º lugar a 2ª letra

em 5º lugar a 4ª letra

Faça seu programa para uma chave de 5 elementos e um texto de até 80 caracteres. Se o texto lido não tiver um número de caracteres múltiplo de 5, complete com espaços em branco.

Por exemplo:

texto: "abcdefgh"

novo texto: "abcdefgh "

Para esse caso a saída seria:

saída: "caebd f g "

Linguagem de Programação C

188) A coordenação do TPD deseja saber quantos alunos estão cursando ao mesmo tempo as cadeiras de LTP1 e Organização de Computadores. Faça um programa que leia os códigos de matrícula dos alunos de ambos os cursos, sabendo-se que existem 150 alunos cursando LTP1 e 120 alunos cursando Organização de Computadores. O programa deve imprimir o código das matrículas dos alunos que estão cursando ambas as disciplinas.

189) Faça um programa que leia uma linha de texto, terminada com um `<Enter>` e imprima a linha com suas palavras na ordem inversa. Considere que cada palavra é separada por apenas um caracter de espaço.

Exemplo: Isto e apenas um exemplo
exemplo um apenas e Isto

190) Faça um programa que realize a reserva de passagens aéreas de uma companhia. Além da leitura do número de vôos e quantidade de lugares disponíveis, ler vários pedidos de reserva, constituídos do numero da carteira de identidade do cliente e do número do vôo desejado.

Para cada cliente, verificar se há possibilidade no vôo desejado. Em caso afirmativo, imprimir o número da identidade do cliente, e o número do vôo, atualizando o número de lugares disponíveis. Caso contrário, avisar ao cliente a inexistência de lugares.

191) Faça um programa capaz de corrigir provas de múltipla escolha. Cada prova terá 10 questões, valendo 1 ponto cada uma. O programa deve inicialmente ler o gabarito da prova. Ele será constituído de um conjunto de dez caracteres representando a resposta para cada uma das questões. Cada questão pode ter uma das seguintes respostas: 'a', 'b', 'c', 'd' ou 'e'.

Após a leitura do gabarito, o programa devera ler as respostas dos alunos. Para cada aluno devera ser lido o seu numero de matricula e suas respostas. O programa devera calcular a nota do aluno e imprimi-la de acordo com o gabarito. A leitura termina com um código de matrícula negativo. Ao final, o programa devera imprimir a percentagem de aprovação, sabendo-se que a nota mínima e 5.

192) Faça um programa que leia uma seqüência de 100 números da entrada de forma desordenada. O programa deve imprimir, após a leitura, os números que se repetiram na seqüência, informando quantos vezes cada um deles apareceu.

193) Faça um programa que leia uma seqüência de 10 números reais da entrada e imprima somente uma vez cada número lido. Mesmo que o número seja digitado mais de uma vez na entrada, deve aparecer somente uma única vez na saída.

194) Uma locadora de vídeo possui 200 títulos de filmes, codificados de 0 a 199. A locadora possui uma ou mais fitas para cada título. Faça um programa em C que leia as seguintes informações sobre cada fita:

Código do filme (0 a 199) e Número de empréstimos (inteiro)

e informe, na saída, para código de filme, quantas fitas devem ser substituídas.

Como um título (filme) pode possuir mais de uma fita, na entrada de dados devem aparecer códigos de filmes replicados, representando as diversas fitas daquele título.

A locadora considera uma fita inutilizável e, portanto, passível de substituição, aquelas que foram emprestadas mais de 500 vezes. A leitura termina quando for lido um código de filme inválido.

195) Faça um programa que permita armazenar os códigos (inteiro) e o preço (real) de um conjunto de peças. O programa deve ser capaz de armazenar os dados de até 100 peças, com códigos variando de 0 a 999. Construa o seu programa, de modo que o usuário seja capaz de **incluir, excluir, consultar e alterar** os dados de uma peça. Faça também uma opção permitindo que o usuário veja todas as peças já cadastradas.

196) Faça um programa para gerenciar um sistema de reservas de mesas em uma casa de espetáculo. A casa possui 30 mesas de 5 lugares cada. O programa deverá permitir que o usuário escolha o código de uma mesa (100 a 129) e forneça a quantidade de lugares desejados. O programa deverá informar se

Linguagem de Programação C

foi possível realizar a reserva e atualizar a reserva. Se não for possível, o programa deverá emitir uma mensagem. O programa deve terminar quando o usuário digitar o código 0 (zero) para uma mesa ou quando todos os 150 lugares estiverem ocupados.

197) (Prova 2 - 96/1) Faça um programa que leia 1000 valores reais da entrada e identifique o valor mais próximo da média.

Exemplo supondo quatro valores:

2.5 7.5 10.0 4.0

média = 6.0 (não deve ser impresso)

valor mais próximo da média = 7.5 (resultado a ser impresso)

198) (Prova 2 - 96/2) Faça um programa que leia dois vetores ordenados, de 100 posições cada, e gere um terceiro vetor, também ordenado, com todos os elementos dos vetores anteriores.

Exemplo para vetores de 5 posições:

entrada: v1 = {1, 4, 5, 9, 111}

v2 = {3, 3, 5, 8, 300}

saída: v3 = {1, 3, 3, 4, 5, 5, 8, 9, 111, 300}

199) (Prova 2 - 96/2) Faça um programa que leia uma seqüência de no máximo 8572 números reais, terminada por 0 (zero). O programa deve imprimir uma ocorrência de cada número.

Exemplo:

entrada: 1 1 4 9.5 1 800 4 -3.2 9.5 801 0

saída: 1 4 9.5 800 -3.2 801

200) (Prova 4 - 96/1) Observe a função main() abaixo:

```
void main (void)
{
    int el[100],qtd;
    puts ("Entre com o número de elementos a serem lidos");
    scanf ("%d",&qtd);
    le_vet (el,qtd);
    computa (el,qtd);
    imprime (el,qtd);
    descomputa (el,qtd);
    imprime (el,qtd);
}
```

Faça as seguintes funções:

- a) le_vet(): esta função lê **n** elementos de um vetor, onde o número de posições e o vetor são recebidos como parâmetros.
- b) imprime(): esta função imprime **n** elementos de um vetor, onde o número de posições e o vetor são recebidos como parâmetros.
- c) computa(): esta função soma a cada elemento do vetor todos os seus subseqüentes, onde o número de posições e o vetor são recebidos como parâmetros.
- d) descomputa(): esta função restaura os valores originais do vetor, onde o número de posições e o vetor são recebidos como parâmetros.

Exemplo para 4 elementos lidos:

Vetor original:

6 3 2 9

Vetor após a execução da função computa():

20 14 11 9

Vetor após a execução da função descomputa():

6 3 2 9

201) (Prova 3 - 96/2) Faça um programa que leia uma seqüência de números reais terminada por 0 (zero) (com, **no máximo**, 1000 números **diferentes** - o usuário é *bonzinho*) e imprima o número de maior ocorrência. Note que pode-se ter mais do que 1000 números na entrada de dados.

202) (Prova 2 - 97/1) Faça um programa que codifique um texto lido da entrada. Inicialmente o programa deverá ler duas tabelas de codificação, uma para codificar as letras minúsculas e outra para as maiúsculas. As tabelas associam a cada letra um outro caracter qualquer. A seguir o programa lerá um texto terminado por uma linha vazia, codificando todas as letras que aparecem no texto.

203) (Prova 2 - 97/1) Faça uma função que receba como parâmetros dois vetores de caracteres e as quantidades de elementos presentes **em cada vetor** (dois valores). A função deve retirar do primeiro vetor todos os caracteres que **NÃO** aparecerem no segundo vetor (interseção) e retornar o número de caracteres restantes no primeiro vetor. Exemplo:

parâmetros:

Lembre-se sempre

Página 22 de

32

“A Sorte Ajuda aos Competentes”

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

```
v1 = {'w', 'b', 'q', 'o', 'a', '-', 't', 'p', 'x', 'r',  
'u', 'o', 'y', 'v', 'a'}
```

```
v2 = {'b', '+', 'e', 'o', '-', 'a', 'a', 'l', 'g', 'o',  
'r', 'v', 'p'}
```

```
n1 = 15      n2 = 13
```

após a execução da função:

```
v1 = {'b', 'o', 'a', '-', 'p', 'r', 'o', 'v', 'a'}
```

```
v2 = {'b', '+', 'e', 'o', '-', 'a', 'a', 'l', 'g', 'o',  
'r', 'v', 'p'}
```

valor retornado pela função: 9

Obrigações do aluno: **não declarar um terceiro vetor na função.**

204) (Prova 2 - 97/1)

- a) Faça uma função chamada `busca_bin()` que recebe um vetor de inteiros, a quantidade de elementos e um valor a ser localizado no vetor (nesta ordem). A função deve retornar a posição do valor no vetor ou -1, caso não o encontre.
- b) Faça uma função chamada `le_dados()` que recebe dois vetores, um de código (int) e outro de salários (float), o tamanho dos vetores (iguais). A função preenche os vetores através da entrada de dados padrão (teclado).
- c) Uma empresa possui 100 vendedores. Cada vendedor recebe um salário fixo e um abono de 1,5% sobre o valor de suas vendas. Faça um programa que calcule o salário final a ser pago a cada vendedor, de acordo com o volume de suas vendas. O programa deve ler inicialmente o código de cada funcionário (inteiro, entre 1533 e 2000) e seu salário base, para todos os vendedores. Considere que estes dados serão fornecidos em ordem crescente por código. A seguir, o programa deverá processar as vendas, lendo o código do funcionário que realizou a venda e o valor da venda. Isto deverá ser feito para cada venda realizada. O programa termina a leitura de vendas quando for digitado um valor de código inválido (menor que 0). O programa deve calcular o abono a ser pago a cada vendedor, imprimindo seu salário final.

Observação: no item c) é obrigatória a utilização das funções dos itens anteriores, mesmo que você não as tenha feito.

205) (Prova 2 - 97/1) Faça um programa que gerencie as informações dos empregados de uma firma de calçados. Para cada empregado deverá ser lido o seu número de cadastro na empresa (inteiro) e o número de meses desde sua contratação. A firma possui 150 empregados. O programa deve imprimir o número de cadastro dos 30 empregados contratados mais recentemente. A firma contrata, no máximo, um empregado por mês.

206) Você foi encarregado de desenvolver um programa em C para controlar o estoque e as vendas das Organizações Tabajara. O programa em questão irá ler inicialmente o código, a quantidade disponível em estoque e o preço dos produtos que estarão disponíveis para a venda. Os dados lidos do teclado serão armazenados em vetores; um **código de produto igual a zero indica o término do cadastramento de produtos**. Sabe-se que as Organizações Tabajara trabalham com no **máximo 50 produtos** diferentes.

A segunda fase do programa é a venda em si. Você deverá ler do teclado o código do produto a ser vendido e a quantidade requerida. Se o código do produto estiver cadastrado a venda poderá ser realizada, caso contrário a mensagem **Produto não Cadastrado** deverá ser exibida no monitor. Caso o produto esteja disponível, a venda só poderá ser realizada se a quantidade disponível no estoque for suficiente para atender ao pedido. Neste caso você deverá abater do estoque a quantidade vendida. Se o estoque não for suficiente para atender ao pedido a mensagem **Estoque Insuficiente** deverá ser exibida no monitor. O final das vendas será detectado quando o **código do produto for igual a zero**.

No final do programa você deverá exibir no monitor o valor vendido no dia e a relação de todos os produtos do estoque, com suas respectivas quantidades, em **ordem decrescente de quantidade disponível**.

a) Você deverá seguir os seguintes passos:

b) Faça uma função que leia do teclado os dados do estoque e armazene-os em vetores (passados

Lembre-se sempre

Página 23 de

Linguagem de Programação C

como parâmetro);

- c) Após chamar a função do **item a**, comece o processo de venda. Você irá ler do teclado os pedidos e atualizará o estoque. Faça uma função para fazer uma busca no vetor de código de produtos para verificar se o produto existe;
- d) No final você deverá exibir um relatório. Faça uma função para ordenar o vetor em ordem decrescente.

Matrizes

Exercícios de Fixação

207) Faça um programa que leia uma matriz de números inteiros de dimensões 4x4 e imprima a soma dos valores de cada linha e a soma de todos os elementos da matriz.

208) Uma estação meteorológica registra a temperatura ocorrida em uma cidade a cada hora. Faça um programa que leia as temperaturas de uma semana no seguinte formato: dia, hora, temperatura. A leitura termina com um dia inválido. O programa deve informar para cada hora consultada, a maior temperatura ocorrida na semana. As consultas terminam com um valor de hora inválido.

209) Faça um programa que leia uma matriz de 4 x 5 elementos, calcule a soma de cada linha e a soma de todos os elementos.

210) Faça um programa que analise um cartão de loteria esportiva e diga qual dos 13 jogos possui um triplo, ou se não existir, um duplo. O programa deve ler da entrada a configuração do cartão e em seguida informar o jogo mais marcado. Dica: fixar a linha e variar a coluna.

211) Refaça o exercício anterior de modo a descobrir qual das colunas do cartão possui mais marcações, se a coluna um, a coluna dois ou a do meio. Dica: fixar a coluna e variar a linha.

212) Construa um programa que efetue a leitura, a soma e a impressão do resultado, entre duas matrizes inteiras que comportem 25 elementos.

213) Construa uma **função booleana** em C que receba como parâmetros uma matriz quadrada e a variável **n**, dimensão da matriz. A função deverá devolver **true(1)**, se a matriz for a matriz identidade de dimensão **n**, ou **false(0)**, caso contrário.

214) Construa uma **função booleana** em C que receba como parâmetros uma matriz quadrada e a variável **n**, dimensão da matriz. A função deverá devolver **true(1)**, se a matriz for simétrica, ou **false(0)**, caso contrário.

215) Construa uma **função booleana** em C que receba como parâmetros duas matrizes quadradas e a variável **n**, dimensão das matrizes. A função deverá imprimir a soma das duas matrizes linha a linha.

216) (Prova 3 - 96/1) Faça um programa que leia uma matriz de inteiros **A** quadrada de ordem 100 e calcule a sua transposta **A^T**. Exemplo para matriz de ordem 4 (**apenas um outro exemplo!!!**):

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{bmatrix}$$

Exercícios Propostos

217) A tabela dada a seguir contém vários itens que estão estocados em vários armazéns de uma companhia. É fornecido, também, o custo de cada um dos produtos armazenados.

	PRODUTO	PRODUTO2	PRODUTO3
	1		
Armazém 1	1200	3700	3737
Armazém 2	1400	4210	4224
Armazém 3	2000	2240	2444
Custo	260	420	330

Fazer um programa que:

Lembre-se sempre

32

“A Sorte Ajuda aos Competentes”

Página 24 de

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

1. leia o estoque inicial;
2. determine e imprime quantos itens estão armazenados em cada armazém;
3. qual o armazém que possui a maior quantidade de produto 2 armazenado;
4. o custo total de:
 - cada produto em cada armazém;
 - estoque em cada armazém;
 - cada produto em todos os armazéns.

218) O tempo que um determinado avião dispensa para percorrer o trecho entre duas localidades distintas está disponível através da seguinte tabela:

	A	B	C	D	E	F	G
A	00	02	11	06	15	11	01
B	02	00	07	12	04	02	15
C	11	07	00	11	08	03	13
D	06	12	11	00	10	02	01
E	15	04	08	10	00	05	13
F	11	02	03	02	05	00	14
G	01	15	13	01	13	14	00

- a) Construa uma função que leia a tabela acima e armazene as informações em uma matriz global;
- b) Construa uma função que leia da entrada o código de duas cidades ('A', 'B', ..., 'F' e 'G') e informe ao usuário o tempo necessário para percorrer a distância entre as duas cidades por ele fornecidas. A leitura termina quando o usuário fornece duas cidades iguais (fonte e destino).
- c) Construa uma função que imprima a tabela sem repetições (apenas o triângulo superior ou o triângulo inferior).
- d) Construa uma função que leia da entrada os códigos de um conjunto de cidades e imprima o tempo total para cumprir todo o percurso entre as cidades fornecidas. A leitura termina quando o usuário fornece um código de cidade inválido.
- e) Faça uma função main () para testar as suas funções.

219) Faça um programa que leia o número de matrícula de 100 alunos e suas notas (5 notas para cada aluno). O programa deve armazenar todos os dados em uma matriz: a matrícula na primeira coluna e as notas nas colunas seguintes. O programa deverá

calcular, para cada aluno, a média de suas notas, que deve ser armazenada na sétima coluna da matriz. Após o cálculo da média, o programa deverá ordenar a matriz pela média de cada aluno. Utilize a função main() abaixo.

```
#include <stdio.h>
#define C 7
#define L 100

void le_matriz (int [][][C]);
void media_alunos (int [][][C]);
void ordena_matriz (int [][][C]);
void troca_linha (int [][][C], int, int);
void troca (int *, int *);
void imprime (int [][][C]);

void main (void)
{
    int i, j;
    int m[L][C];

    le_matriz (m);
    media_alunos (m);
    ordena_matriz (m);
    imprime (m);
}
```

Teste seu programa com os seguintes dados:

Matrícula	N1	N2	N3	N4	N5
134	8	5	7	9	10
99	8	6	5	2	3
120	10	9	10	10	8

A matriz resultante será:

Matrícula	N1	N2	N3	N4	N5	Média
99	8	6	5	2	3	3
134	8	5	7	9	10	7
120	10	9	10	10	8	9

220) A Escola Pagou Passou possui 10 turmas, com **exatamente** 60 alunos em cada uma, e a sua diretoria resolveu fazer o acompanhamento do rendimento dos alunos matriculados em cada uma das turmas.

Linguagem de Programação C

Para atender as necessidades da diretoria você deverá construir um programa em C que faça o seguinte:

- a) defina duas matrizes para armazenar as matrículas e as notas dos alunos das 10 turmas, cada qual com **exatamente** 60 alunos;
- b) faça uma **função** que leia do teclado a matrícula e a nota dos alunos e armazene-as nas respectivas matrizes, turma a turma. Lembre-se de que a escola tem **exatamente** 10 turmas, cada uma das quais com **exatamente** 60 alunos. A **função** deverá receber as duas matrizes como parâmetros;
- c) faça uma **função** que calcule e exiba no vídeo a média de cada turma, armazenando-as em um vetor de médias com 10 elementos. As matrizes e o vetor deverão ser passados como parâmetros;
- d) faça uma **função** que percorra a matriz de notas, turma a turma, exibindo no vídeo, uma a uma, as matrículas dos alunos cujas as notas ficaram abaixo da média das respectivas turmas.
- e) a **função main** deverá chamar cada uma das **funções** acima de maneira que o programa cumpra o seu objetivo corretamente.

Strings

Exercícios de Fixação

221) Faça uma função chamada *strcpy* que recebe duas *strings* como parâmetros e copie todos os elementos da segunda *string* para a primeira, apagando-a. Faça um programa para testar sua função.

222) Faça uma função chamada *strlen* que recebe uma *string* como parâmetro e retorne o número de caracteres da *string*. O caractere '\0' não deve ser considerado na contagem. Faça um programa para testar sua função.

223) Faça uma função chamada *strcat* que recebe duas *strings* como parâmetros e copie todos os

caracteres da segunda *string* para o final da primeira *string*. Faça um programa para testar sua função.

224) Faça uma função chamada *strcmp* que recebe duas *strings* (*s1* e *s2*) como parâmetros e compara os seus conteúdos. A função retorna um número < 0 se *s1* for menor que *s2*, um número > 0 se a *s1* for maior que *s2* e 0 (zero) se as *strings* forem iguais. Faça um programa para testar sua função.

225) Faça uma função chamada *puts* que recebe uma *string* como parâmetro e imprime seu conteúdo na tela. Após imprimir a *string*, a função deve “pular uma linha”. Faça um programa para testar sua função.

226) Faça uma função chamada *gets* que recebe uma *string* como parâmetro e preenche esta *string* com os caracteres lidos através do teclado. A leitura termina quando o usuário digita <Enter>. Faça um programa para testar sua função.

227) Faça uma função que receba uma *string* como parâmetro e imprima a *string* invertida.

228) Faça uma função que receba uma *string* como parâmetro e converta todos as letras minúsculas encontradas em letras maiúsculas. Os demais caracteres permanecem inalterados.

229) Faça uma função que receba uma *string* como parâmetro e retorne o número de palavras encontradas. Cada palavra pode estar separada por um ou mais caracteres de espaço. Não considere os caracteres de pontuação como separadores de palavras.

230) Refaça a questão anterior considerando os caracteres '.', ',', ';' como separadores de palavras, além do espaço em branco.

231) Faça uma função que receba uma *string* como parâmetro e um caractere. A função deve retornar a primeira posição onde o caractere foi encontrado na *string* ou -1 caso o caractere não apareça na *string*.

232) Faça uma função que receba uma *string* como parâmetro e um caractere. A função deve retornar a

Lembre-se sempre

Página 26 de

Linguagem de Programação C

última posição onde o caracter foi encontrado na *string* ou -1 caso o caracter não apareça na *string*.

233) Faça uma função que receba uma *string* como parâmetro e um caracter. A função deve retirar todas as ocorrências deste caracter da *string*.

234) Faça uma função que receba uma *string* como parâmetro e dois caracteres. A função deve trocar toda a ocorrência do primeiro caracter na *string* pelo segundo caracter.

235) Faça uma função que receba uma *string* como parâmetro e um caracter. A função deve retornar o número de vezes que o caracter aparece na *string*.

236) Faça uma função que receba uma *string* como parâmetro, um caracter e um valor numérico. A função deve preencher a *string* com tantos caracteres quanto for o valor numérico passado com parâmetro.

237) Faça uma função chamada *strncpy* que receba duas *strings* (s1 e s2) e um valor numérico como parâmetros. A função deve copiar tantos caracteres quanto o valor numérico, da *string* s2 para a *string* s1.

238) Faça um função chamada *stricmp* que compara duas *strings* (s1 e s2) passadas como parâmetro, não diferenciando maiúsculas de minúsculas. A função deve retornar: 0 se as *strings* forem iguais; um valor menor que 0 se s1 for menor que s2 e um valor maior que 0 se s1 for maior que s2.

Exercícios Propostos

239) (Prova 3 - 94/1) Faça uma função que receba como parâmetro uma *string* de, no máximo, 80 caracteres e converta todas as letras para minúsculas, exceto a primeira letra de cada palavra, que deve ser maiúscula. Considere que a *string* só possui letras e espaços e entre as palavras existe um único espaço.

Exemplo:

Frase recebida:

EsTa fRase e apENAS uM EXEmPLO

Frase gerada:

Esta Frase E Apenas Um Exemplo

240) Faça uma função que receba uma *string* como parâmetro e imprima esta *string* em modo justificado, ou seja, preenchendo os intervalos entre as palavras com espaços em branco de maneira proporcional. Considere este procedimento apenas para *strings* com menos de 80 caracteres. As *strings* com mais de 80 caracteres devem ser impressas sem alteração.

241) Faça uma função que receba uma *string* como parâmetro e a imprima. Se a *string* tiver mais de 40 caracteres, ela deve ser particionada, quantas vezes for necessário, de modo a imprimir no máximo 40 caracteres por linha, sendo que nenhuma palavra deve ser impressa fragmentada.

242) Modifique a função do item anterior de modo a imprimir a *string* em modo justificado, além de realizar o seu particionamento.

243) Faça uma função que receba duas *strings* como parâmetro (s1 e s2) e retorne a posição da *string* s2 dentro da *string* s1, caso a *string* s1 contenha a *string* s2, ou -1 em caso contrário.

Exemplo:

s1	'c'	'a'	'm'	'a'	'l'	'e'	'ã'	'o'	'\0'
s2	'l'	'e'	'ã'	'o'	'\0'				

A função deve retornar o valor 4, pois é o valor da posição onde começa a *string* “leão” na *string* “camaleão”.

244) Faça uma função que receba uma *string* como parâmetro e verifique se é um palíndromo, isto é, da esquerda para direita ou da direita para a esquerda têm-se a mesma sequência de caracteres. Exemplos: ama, ovo, asa.

245) (Prova 3 - 94/1) Considere a seguinte lista de países e respectivas capitais:

Canadá	Ottawa
Inglaterra	Londres
França	Paris
Portugal	Lisboa
Japão	Tóquio

Lembre-se sempre

32

“A Sorte Ajuda aos Competentes”

Página 27 de

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

Escreva um programa interativo que aceite um nome de um país como entrada e imprima o nome de sua capital. O programa deve executar repetidamente até que seja digitada a palavra FIM.

246) (Prova 3 - 96/1) Faça uma função que receba duas strings e um inteiro como parâmetros. Esta função inclui a segunda string a primeira a partir da posição indicada pelo inteiro. Exemplo (**apenas um exemplo!!!**):

1ª string: “isto é exemplo”

resultado após execução da função:

2ª string: “apenas um ”

1ª string: “isto é apenas um exemplo”

inteiro: 7

2ª string: “apenas um ”

247) (Prova 4 - 96/1) Faça uma função chamada **strmix** que receba 3 strings (s1, s2 e s3) e misture s1 e s2 do seguinte modo:

1º caracter de s3: 1º caracter de s1

2º caracter de s3: 1º caracter de s2

3º caracter de s3: 2º caracter de s1 e assim sucessivamente

Observações: a string resultante deverá ser armazenada em s3. As strings s1 e s2 podem ter tamanhos diferentes.

Exemplo:

s1: “abcde”

s2: “012”

string resultante (s3): “a0b1c2de”

248) (Prova 3 - 96/2) Faça uma função que receba duas strings e retire da primeira todo caracter que fizer parte da segunda string. Exemplo:

1ª string: “EFBACDEIEBBLBBD”

2ª string: “EBD”

Resultado

1ª string: “FACIL”

2ª string: “EBD”

249) (Prova 3 - 97/1) Faça uma função chamada **compara()** que receba duas strings e dois inteiros **pos** e **qtd**, onde **pos** é a posição nas strings a partir da qual vamos comparar **qtd** caracteres. A comparação **NÃO** deve levar em conta se os

caracteres são maiúsculos ou minúsculos. A função retorna 0 (zero) se as porções de strings forem iguais, senão, ela retorna -1. Dica: use uma função que converta uma letra para maiúsculo. Considere que os valores de **pos** e **qtd** são sempre válidos.

Exemplo:

```
string 1: Banana
string 2: ANNA MARIA
pos = 2
qtd = 2
retorna 0
```

```
string 1: Banana
string 2: ANNA MARIA
pos = 2
qtd = 3
retorna -1
```

250) (Prova 4 - 96/2) Faça um programa que leia um texto, em japonês, terminando com uma linha vazia e imprima o número de ocorrências de cada palavra no texto. O programa deverá imprimir uma lista das palavras que apareceram no texto e o número de ocorrências de cada uma delas.

Sabe-se que em japonês, existem 2548 palavras, a maior palavra possui 78 letras e entre cada palavra existe apenas um único espaço em branco.

a) faça uma função que leia uma palavra,

b) faça uma função para procurar uma palavra em uma lista de palavras,

c) faça o programa utilizando as funções acima.

251) (Prova 4 - 96/2) Faça uma função chamada **xxx()** que receba duas strings como parâmetros e retorne a posição da 2ª string de trás para frente dentro da 1ª string ou um número negativo caso não encontre.

Faça um programa para testar se esta função que você fez está funcionando.

Obs.: note que as strings podem ter qualquer tamanho!!!

Exemplos:

	Exemplo 1	Exemplo 2
1ª string:	Esta é uma linha	Esta é uma linha
2ª string:	amu	uma
resultado:	7	-1542

Lembre-se sempre

32

“A Sorte Ajuda aos Competentes”

Página 28 de

Prof.: Oswaldo Borges Peres
profoswaldo@yahoo.com.br

Linguagem de Programação C

252) (Prova 4 - 97/1) Faça uma função chamada **cria_string()** que recebe como parâmetros uma string **s1** (de tamanho suficiente), um formato **fmt** (também uma string) e um número inteiro positivo **num** (nesta ordem). Esta função irá preencher a string **s1** de acordo com o formato passado como parâmetro e o valor do inteiro **num**. O formato irá conter uma sequência de caracteres que devem ser integralmente copiados para a string **s1**, com exceção da sequência **%d**. No seu lugar deve ser copiado o valor do inteiro **num** convertido para caracteres.

Dica: lembre-se que o maior número inteiro que pode ser representado no TC é 32767 (5 dígitos).

Exemplos:

Exemplo 1

Parâmetros passados:

s1: (irrelevante, mas de tamanho suficiente)

fmt: "O valor %d é a resposta"

num: 15

Resultado após a execução da função

s1: "O valor 15 é a resposta"

fmt: (irrelevante)

num: (irrelevante)

Exemplo 2

Parâmetros passados:

s1: (irrelevante, mas de tamanho suficiente)

fmt: "Converta o %d para caracteres"

num: 17430

Resultado após a execução da função

s1: "Converta o 17430 para caracteres"

fmt: (irrelevante)

num: (irrelevante)

Arquivos

Exercícios de Fixação

253) Faça um programa que leia um arquivo. O arquivo contém linhas de texto de até 80 caracteres. O nome do arquivo deve ser perguntado (sem extensão) e a extensão é txt. O programa deve

converter todas as letras minúsculas encontradas para letras maiúsculas. Os dados convertidos devem ser escritos em outro arquivo. O nome do arquivo de saída deve ser o mesmo do que foi perguntado com extensão out.

254) Faça um programa que leia do arquivo matriz.txt os valores de uma matriz 10 x 10 de números inteiros e imprima na tela a matriz e o resultado da soma de todos os seus valores. Dica: use a função *fscanf*. O resultado para a matriz que está no disquete é 2085.

255) Faça um programa que leia do arquivo matriz.txt os valores de uma matriz 10 x 10 de números inteiros e gere um arquivo chamado transp.txt, contendo a matriz transposta da que foi lida.

256) Faça um programa que leia da entrada uma sequência de até 100 nomes. O usuário pode terminar a leitura com um nome vazio, ou seja, digitando <Enter> como primeiro caractere de um novo nome. Cada nome lido deverá ter no máximo 40 caracteres. Armazene esta sequência de nomes em um arquivo chamado nomes.txt.

257) Faça um programa que leia o arquivo nomes.txt, gerado por você, ordene alfabeticamente a sequência de nomes e os imprima na tela.

258) Faça um programa que leia da entrada o nome (40 caracteres) e a nota final (*float*) dos alunos de uma turma de LTP1. Cada turma tem no máximo 50 alunos. O programa deve armazenar os dados em um arquivo binário chamado "ltp1".

259) Faça um programa que leia o arquivo "ltp1", criado por você, e imprima a média dos alunos da turma e quantos alunos tiveram notas acima da média.

260) Faça um programa que leia um arquivo contendo linhas de texto e imprima o número de caracteres, o número de linhas e o número de palavras contidas no texto. Considere que cada linha do texto não contém mais que 80 caracteres (adicione mais um para o '\n' e outro para o '\0').

Linguagem de Programação C

261) Faça um programa que leia um arquivo contendo linhas de texto e imprima as linhas em ordem alfabética. Considere que cada linha do texto não contém mais que 80 caracteres.

262) Faça um programa que leia os valores de duas matrizes inteiras 3x3 da entrada e armazene em um arquivo binário chamado matriz o resultado da soma das duas matrizes.

263) Faça um programa que leia um número da entrada e a matriz contida no arquivo binário matriz e imprima o resultado da multiplicação do número pelos elementos da matriz.

264) Faça um programa que leia da entrada um texto de no máximo 50 linhas. Cada linha terá no máximo 80 caracteres. O programa deve armazenar este texto em um arquivo binário chamado texto.

265) Faça um programa que leia o arquivo binário "texto", converta todas as letras encontradas para minúsculas e armazene o resultado em um arquivo texto chamado texto.txt.

266) Faça um programa que leia da entrada um nome de arquivo. Este arquivo contém linhas de texto e deve ser impresso na tela em modo justificado. Considere que as linhas terão no máximo 80 caracteres. Obs.: Utilize a função de justificar texto.

267) Faça um programa que leia de um arquivo texto (2a.txt) duas *strings* de até 15 caracteres e leia dois números inteiros do teclado (n1 e n2) e grave um segundo arquivo texto (2b.txt) com n1 vezes a primeira *string* e n2 vezes a segunda *string*.

268) Faça um programa que leia, de um arquivo binário (3a.bin), um conjunto de, no máximo, 20 números inteiros e grave outro arquivo binário (3b.bin) com os mesmos números na ordem inversa.

269) Faça um programa que leia um arquivo texto (4a.txt) com nomes de até 30 caracteres e idades e grave um arquivo texto (4b.txt) com os dados daqueles cujos nomes comecem por 'A'.

270) Faça um programa que leia um arquivo texto (5a.txt) com nomes de até 30 caracteres e grave um outro arquivo binário (5b.bin) com o nome e a quantidade de letras maiúsculas nesse nome.

271) Faça um programa que leia dois números inteiros e uma matriz, também de inteiros, de no máximo 30x30, todos eles de um arquivo binário (6a.bin). Os números são a quantidade de linhas e de colunas, respectivamente. O programa deve gravar, em outro arquivo binário (6b.bin), a soma das linhas dessa matriz.

272) Faça um programa que leia um arquivo texto (7a.txt) com linhas de no máximo 80 caracteres e imprima em um arquivo binário (7b.bin) o número total de vogais, consoantes e caracteres não letras encontrados no texto. O texto contém apenas letras minúsculas.

273) Faça um programa que leia dois caracteres da entrada e um arquivo de texto (8a.txt) contendo linhas de no máximo 80 caracteres. O programa deve trocar todas as ocorrências do primeiro caractere no texto pelo segundo caractere e imprimir o texto resultante em outro arquivo texto (8b.txt).

274) Faça um programa que leia um arquivo texto (9a.txt) contendo 10 nomes de até 15 caracteres e grave um outro arquivo texto (9b.txt) com os nomes ordenados.

275) Faça um programa que leia uma matriz de caracteres 6x6 de um arquivo texto (10a.txt) e grave um arquivo binário (10b.bin) com o número da linha e a maior letra de cada linha.

276) Faça um programa que leia uma linha de até 80 caracteres do teclado e gere um arquivo texto (11a.txt) com os caracteres na seguinte sequência: 1º, último, 2º, penúltimo, 3º, ante-penúltimo, ...

Exercícios Propostos

277) Faça um programa que gerencie uma agenda de telefones. O programa deve ser capaz de armazenar as informações para até 100 pessoas. A agenda deve conter o nome e o telefone de cada pessoa, devendo

Lembre-se sempre

Página 30 de

Linguagem de Programação C

ser possível realizar as seguintes operações: inclusão de um novo telefone; alteração do número de um telefone já cadastrado; exclusão de um telefone; impressão dos telefones cadastrados; ordenação por nome; consulta a partir do nome de uma pessoa; armazenamento dos dados em um arquivo e recuperação dos dados do arquivo.

278) (Prova 3 - 97/1) Faça um programa, utilizando as funções abaixo, que leia de um arquivo chamado **nomenota.txt**, um conjunto de **até** 100 nomes de alunos, com **até** 50 caracteres, e suas respectivas médias finais (números reais). O programa deverá permitir alteração nos nomes e nas notas dos alunos através de um menu com as opções: a) alterar nome, b) alterar média, c) terminar. Ao final, o programa deverá guardar os dados no mesmo arquivo, no mesmo formato.

- a) faça uma função chamada `le_tudo()` que leia as informações do arquivo. Esta função recebe um vetor de strings e um vetor de reais como parâmetros (nesta ordem). A função retorna a quantidade de nomes lidos. Se deu erro de leitura, a função retorna 0 (zero).
- b) faça uma função chamada `escreve_tudo()` que escreva tudo no arquivo. Esta função recebe um vetor de strings, um vetor de reais e o número de elementos a gravar como parâmetros (nesta ordem). A função não retorna valor algum.
- c) faça uma função chamada `busca()` que realize uma busca em um vetor de strings. Esta função recebe o vetor de strings, uma string a ser localizada e o número de elementos do vetor. A função retorna a posição da string no vetor ou -1 caso não a encontre.
- d) faça uma função chamada `altera_nome()` que realize a alteração do nome do aluno. Esta função recebe o vetor de strings, uma string contendo o nome anterior, uma string com o nome corrigido e o número de elementos do vetor. A função retorna -1 se não conseguir alterar o nome. Use a função `busca()`.
- e) faça uma função chamada `altera_media()` que realize a alteração da média de um aluno. Esta função recebe o vetor de strings, o vetor de

médias, uma string contendo o nome do aluno, a nova média e o número de elementos dos vetores (o mesmo para ambos). A função retorna -1 se não conseguir alterar a média. Use a função `busca()`.

Obs.:

- a) O programa deverá avisar ao usuário sempre que não puder realizar as alterações.
- b) O arquivo possui uma linha com nome, outra linha com nota, depois mais um nome e uma nota e assim por diante.
- c) Não se esqueça de fazer a função `main()`.

279) (Prova 4 - 97/1) Faça um programa que leia um arquivo **texto** chamado **entrada.txt** e imprima o número de caracteres, palavras e linhas encontrados no texto. Os únicos separadores de palavras são o caracter de espaço em branco e o caracter de fim de linha. As palavras serão separadas por apenas um caracter de espaço em branco ou caracter de fim de linha.

Obs.:

- a) Os tamanhos da maior palavra e da maior linha são desconhecidos, logo não adianta declarar um vetor de tamanho enorme para guardar uma palavra ou uma linha.
- b) Não se sabe, a priori, quantas linhas o arquivo possui, nem muito menos quantas palavras por linha

Exemplo para você se basear:

Arquivo `entrada.txt`:

```
Este texto é apenas para você se
basear<FIM_DE_LINHA>
O seu programa deve funcionar para outros
exemplos<FIM_DE_LINHA>
Lembre que tu não sabes quantas linhas o
arquivo possui<FIM_DE_LINHA>
<FIM_DO_ARQUIVO>
```

```
Saída do programa
número de caracteres = 144
número de palavras = 26
número de linhas = 3
```

280) (Prova 4 - 97/1) Você foi contratado para desenvolver um programa de reserva de passagens

Lembre-se sempre

Página 31 de

Linguagem de Programação C

para uma companhia aérea que faz vôos para no **máximo 10 cidades** (destinos) diferentes. Com o objetivo de racionalizar as suas operações, a companhia opera com no **máximo 5 tipos de aeronaves** diferentes. O programa a ser desenvolvido deverá fazer o seguinte:

- a) faça uma função, chamada `le_aviao()`, que receba como parâmetros um vetor de strings e um vetor de inteiros. Esta função deverá ler do arquivo texto `arq_aviao.txt` o modelo da aeronave (um string com exatamente 4 caracteres) e a quantidade de assentos da mesma (inteiro), carregando as informações lidas do arquivo nos vetores passados como parâmetros. A função deverá retornar quantos elementos foram lidos (o mesmo para ambos).
- b) faça uma função, chamada `le_destino()`, que receba como parâmetros dois vetores de strings. Esta função deverá ler do arquivo texto `arq_destino.txt` uma cidade para a qual a companhia voa regularmente (um string com exatamente 3 caracteres) e o modelo da aeronave usado para esta rota (um string com exatamente 4 caracteres), carregando as informações lidas do arquivo nos vetores passados como parâmetros. A função deverá retornar quantos elementos foram lidos (o mesmo para ambos).
- c) faça uma função, chamada `pesq_destino()`, que receba como parâmetros um vetor de strings (tabela de destinos), um inteiro (tamanho da tabela de destinos), e um string (destino). A função deverá pesquisar na tabela de destinos o destino passado com parâmetro, retornando a sua posição na tabela de destinos caso a pesquisa seja bem sucedida, ou o valor inteiro -1, caso contrário.
- d) faça uma função, chamada `pesq_aviao()`, que receba como parâmetros um vetor de strings (tabela de aeronaves), um inteiro (tamanho da tabela de aeronaves), e um string (aeronave). A função deverá pesquisar na tabela de aeronaves a aeronave passada com parâmetro, retornando a sua posição na tabela de aeronaves caso a pesquisa seja bem sucedida, ou o inteiro -1, caso contrário.

- e) faça a função `main()` que deverá ler continuamente do teclado as cidades para as quais cada um dos usuários quer voar (string de 3 caracteres. Ex: RJO); uma string vazia indica o término do programa. Se a cidade para qual o usuário quiser voar não for atendida pela companhia (isto é, não está no vetor de destino) envie uma mensagem adequada. Caso contrário tente fazer uma reserva para o usuário no vôo para a cidade solicitada. Neste caso você deverá verificar se ainda existem lugares vagos no vôo, em função da capacidade da aeronave empregada na rota desejada. Caso a reserva possa ser feita envie a mensagem Reserva Confirmada, caso contrário envie a mensagem Vôo Lotado.

Dicas:

- a) lembre-se que aviões diferentes mas de mesmo modelo podem atender a mais de um destino.
- b) crie um vetor de lugares vagos para cada destino.