



Universidade Federal do Rio Grande do Norte – UFRN
Escola Agrícola de Jundiaí – EAJ
Tecnologia em Análise e Desenvolvimento de Sistemas – TADS
Algoritmos e Programação – TAD0102
Profa. Alessandra Mendes

LISTA 7 – 17/07/2024

Conteúdos: Funções, Variáveis locais e globais, passagem de parâmetros por valor: conceito de modularização, tipos de retorno, passagens de parâmetro por valor, escopo das variáveis, chamadas a funções e ponteiros.

Elabore os seguintes algoritmos:

1. Numa disciplina são dadas duas provas e dois trabalhos, mas a média é calculada considerando-se apenas a maior nota de prova e a maior nota de trabalho. Elabore um algoritmo que, dadas as quatro notas de um aluno, informe a sua média. Para isso, utilize duas funções: uma que determine o maior valor entre as notas (de provas ou de trabalhos) e outra que calcule a média aritmética das duas maiores notas.
2. Faça uma função que recebe um número inteiro por parâmetro e retorna verdadeiro se ele for par e falso se for ímpar.
3. Escreva uma função que recebe 2 números inteiros $n1$ e $n2$ como entrada e retorna a soma de todos os números inteiros contidos no intervalo $[n1, n2]$. Use esta função em um programa que lê $n1$ e $n2$ do usuário e imprime a soma.
4. Escreva um programa que lê um valor inteiro (maior do que 1 e menor ou igual a 10) e exibe a tabuada (até 10) de multiplicação do número lido. Você deverá escrever as seguintes funções:
- `int LeNumero(int n1, int n2)`
Lê um número inteiro no intervalo especificado ($n1$, $n2$) e o retorna. Cada vez que for digitado um número inválido (fora do intervalo especificado) a função deve exibir a mensagem "Número inválido. Digite novamente!"
- `void Tabuada(int n)`
Recebe como parâmetro um número inteiro e exibe na tela a tabuada de multiplicação até 10 do número lido. Exemplo: número lido 5
5 x 1 = 5 5 x 2 = 10 5 x 10 = 50
5. Elaborar o algoritmo da calculadora utilizando funções. Implementar as seguintes opções: soma, subtração, multiplicação e divisão.

6. Elaborar um algoritmo que leia um vetor de 10 posições de inteiros e chame uma função MinMax. Esta função deve receber o vetor lido e, por referência, duas variáveis inteiras, min e max. O objetivo da função é buscar os valores do menor e maior elementos do vetor e atribuir às variáveis min e max respectivamente.

7. Escreva uma função chamada Troca que troque os valores dos parâmetros recebidos. Essa função não deve ter retorno.

8. Escreva uma função que determine a média e a situação de um aluno em uma disciplina. A função recebe como parâmetros as três notas de um aluno (p1, p2, e p3), seu número de faltas (faltas), o número total de aulas da disciplina (aulas) e o ponteiro para uma variável (media), conforme a seguinte assinatura:

char situacao(float p1, float p2, float p3, int faltas, int aulas, float *media); Na variável indicada pelo ponteiro media, a função deve armazenar a média do aluno, calculada como a média aritmética das três provas. Além disso, a função deve retornar um caractere indicando a situação do aluno no curso, definido de acordo com o seguinte critério:

Número de Faltas	Média	Situação	Retorno
Menor ou igual a 25% do total de aulas	Maior ou igual a 6,0	Aprovado	A
	Menor que 6,0	Reprovado	R
Maior que 25% do total de aulas	Qualquer	Reprovado por faltas	F

Em seguida, escreva a função principal de um programa que utiliza a função anterior para determinar a situação de um aluno. O programa deve:

- Ler do teclado três números reais e dois números inteiros, representando as notas da p1, p2 e p3, o número de faltas e o número de aulas, respectivamente;
- Chamar a função desenvolvida na primeira questão para determinar a média e a situação do aluno na disciplina;
- Exibir a média (com apenas uma casa decimal) e a situação do aluno, isto é, "APROVADO", "REPROVADO" ou "REPROVADO POR FALTAS", dependendo do caractere retornado pela função, conforme a tabela acima.

9. Crie um algoritmo para manipular vetores. O seu programa deve implementar uma função chamada inverteVetor, que recebe como parâmetro dois vetores V1 e V2, ambos de tamanho N. A função deve copiar os elementos de V1 para V2 na ordem inversa. Ou seja, se a função receber V1 = {1,2,3,4,5}, a função deve copiar os elementos para V2 na seguinte ordem: V2 = {5,4,3,2,1}. Além disso, a função também deve retornar o maior valor encontrado em V1. A função deve possuir a seguinte assinatura: int inverteVetor(int *v1, int *v2, int n); Elabore um teste da função no main.

10. Em seguida, utilizando o algoritmo anterior implemente outra função chamada `multiplicaEscalar`, que recebe como parâmetro dois vetores V1 e V2 (ambos de tamanho N), e um número inteiro X. A função deve multiplicar cada um dos elementos de V1 por X e armazenar os resultados em V2. A função deve possuir a seguinte assinatura: `void multiplica_escalar(int *v1, int *v2, int x, int n);`

No main, utilize as funções `inverteVetor` e `multiplicaEscalar` para inverter um vetor de tamanho 10 fornecido pelo usuário e em seguida multiplicar esse vetor por um escalar também fornecido pelo usuário. Por último, o programa deverá exibir o vetor resultante.

11. Elabore um algoritmo que apresente o menu abaixo e, de acordo com a opção escolhida pelo usuário, realize a respectiva operação conforme detalhamento.

MENU PRINCIPAL

- 1 - ORDEM
 - 2 - DATA
 - 3 - SALÁRIO
 - 4 - MULTA
 - 5 - CALCULAR
 - 6 - SAIR
-

Opção 1: leia três valores inteiros e um char ('c' – crescente, 'd' – decrescente), passe-os para uma função que deverá ordená-los de acordo com a ordem escolhida e retorná-los para o chamador, que deverá escrever o resultado da ordenação.

Opção 2: Declare o registro Data contendo 3 campos inteiros, dia, mês e ano, e leia 5 datas em um vetor de datas. Passe o vetor lido para uma função que deverá verificar e retornar um vetor booleano contendo true em cada posição cuja data for válida e false em cada posição que a data não for válida. Considerar meses de 30 e 31, desconsiderar anos bissextos. O chamador, após receber o resultado da função, deverá escrever uma mensagem de "Data Válida" ou "Data Inválida" para cada resultado recebido no vetor.

Exemplo: Vetor de Datas: 21/1/2000, 31/12/2012, 45/3/2000, 1/40/2000, 31/02/1999

Saída da função: true, true, false, false, false

Opção 3: leia o salário de uma pessoa e a sua quantidade de filhos e mande-os para uma função. A função deverá retornar quanto do salário da pessoa será dedicado a cada um dos seus filhos (considerando 10% para cada filho, desde que não exceda o 50% do valor total do salário. Se exceder, o valor para cada filho deverá ser 50% do salário dividido pelo total de filhos. O chamador deverá informar o valor do salário por filho ou, se a pessoa não tiver filhos, a mensagem "O seu salário é 100% para o seu próprio custeio".

Opção 4: leia a velocidade máxima permitida em uma avenida e a velocidade com que o motorista estava dirigindo nela, envie-os para uma função void que deverá calcular se a pessoa vai pagar multa e, se for, qual o valor da multa. Considere que serão pagos como multa: a) 50 reais se o motorista ultrapassar em até 10km/h a velocidade permitida (ex.: velocidade máxima: 50km/h; motorista a 60km/h ou a 56km/h); b) 100 reais, se o motorista ultrapassar de 11 a 30 km/h a velocidade permitida. c) 200 reais, se estiver acima de 31km/h da velocidade permitida. A mensagem informando o valor da multa deverá ser escrita pelo chamador após a execução da função.

Opção 5: leia o gabarito correto de uma prova e o gabarito de um aluno em uma matriz 2x10 (10 respostas do gabarito na primeira linha da matriz e 10 repostas do aluno na segunda linha da matriz) e envie para uma função. A função deverá calcular e retornar o percentual de acerto do aluno na prova (int) e a sua nota (float), considerando que cada questão vale 1,5 ponto. Escreva os resultados retornados pela função.

Opção 6: Escreva a mensagem “Fim de Programa” e encerre o algoritmo.

12. Escreva uma função *hm* que converta minutos em horas-e-minutos. A função recebe um inteiro *mnts* e os endereços de duas variáveis inteiras, *h* e *m*, e atribui valores a essas variáveis de modo que *m* seja menor que 60 e que $60 * h + m$ seja igual a *mnts*. Elabore um algoritmo que teste a função *hm*.

13. Escreva uma função *mm* que receba um vetor inteiro $v[0..n-1]$ e os endereços de duas variáveis inteiras, *min* e *max*, e deposite nessas variáveis o valor do menor elemento (mínimo) e do maior elemento (máximo) do vetor. Elabore um algoritmo que teste a função *mm*.

14. Escreva uma função que receba uma *string* e uma letra e retorne um vetor de inteiros contendo as posições (índices no vetor da *string*) onde a letra foi encontrada e um inteiro contendo o tamanho do vetor criado (total de letras iguais encontradas). Utilize o retorno de um vetor para retornar os índices e um ponteiro para guardar o tamanho do vetor. Elabore um algoritmo que teste a função supracitada.

15. Considerando uma lista encadeada de valores inteiros definida pela struct abaixo:

```
struct noh{
    int valor;
    lista *próximo;
};
Noh meuNoh;
```

Elabore um algoritmo que apresente o menu abaixo e execute as operações de acordo com a escolha do usuário.

MENU PRINCIPAL

-
- 1 – ADICIONAR ELEMENTO
- 2 – REMOVER ELEMENTO
- 3 – MOSTRAR TODOS OS ELEMENTOS
- 4 – SAIR
-

Opção 1: o algoritmo deverá ler um elemento inteiro, inseri-lo em um novo nó e adicioná-lo no fim da lista utilizando a função *adicionar* que deve ser elaborada. Opção

2: o algoritmo deverá ler um elemento inteiro, busca-lo na lista e, caso seja encontrado, removê-lo utilizando a função *remover* que deve ser elaborada.

Opção 3: o algoritmo deverá exibir todos os elementos da lista utilizando a função *relatorio* que deve ser elaborada.

16. Escreva uma função que receba como parâmetros dois vetores de inteiros: x_1 e x_2 e as suas respectivas quantidades de elementos: n_1 e n_2 . A função deverá retornar um ponteiro para um terceiro vetor, x_3 , alocado dinamicamente, contendo a união de x_1 e x_2 e usar o ponteiro qtd para retornar o tamanho de x_3 . Sendo $x_1 = \{1, 3, 5, 6, 7\}$ e $x_2 = \{1, 3, 4, 6, 8\}$, x_3 irá conter $\{1, 3, 4, 5, 6, 7, 8\}$. Assinatura da função:

```
int* uniao(int *x1, int *x2, int n1, int n2, int* qtd);
```

17. Escreva um programa que leia números 10 inteiros, armazene-os em um vetor e os escreva na ordem contrária à de leitura. Obs: todos os acessos ao vetor devem ser feitos usando somente ponteiros, sem usar os índices.