Introdução a Programação COLTEC/UFMG
Professores: João Eduardo Montandon e Virgínia Fernandes Mota
Turmas: 103 e 106
Lista preparatória para a primeira prova (Data de entrega: 02/05)

Parte I - Subrotinas

Desenvolver os respectivos programas em C para resolver os problemas abaixo:

1. Faça uma função que recebe a idade de uma pessoa em anos, meses e dias e retorna essa idade expressa em dias.

2. Faça uma função que recebe a média final de um aluno por parâmetro e retorna o seu conceito, conforme a tabela abaixo:

Nota	Conceito
De 0 a 49	D
De 50 a 69	C
De 70 a 89	B
De 90 a 100	A

3. Faca uma função que recebe por parâmetro o raio de uma esfera e calcula o seu volume ($v = 4/3.\pi.R^3$).

4. Escrever uma função int contaimpar(int n1, int n2) que retorna o número de inteiros impares que existem entre n1 e n2 (inclusive ambos, se for o caso). A função deve funcionar inclusive se o valor de n2 for menor que n1.

Ex: n = contaimpar(10,19); /* n recebe 5 (11,13,15,17,19) */
n = contaimpar(5,1); /* n recebe 3 (1,3,5) */

5. Escrever um procedimento void estacao(int dia, int mes), que exibe no vídeo qual a estação do ano da data passada por parâmetro. Lembrando que a primavera começa no dia 23 de setembro, o verão em 21 de dezembro, o outono em 21 de março e o inverno em 21 de junho.

Ex: estacao(25,10); /* 25/10 é primavera. */
estacao(29,12); /* 29/12 é verão. */

6. Escrever uma função int divisao(int dividendo, int divisor, int *resto), que retorna a divisão inteira (sem casas decimais) de dividendo por divisor e armazena no parmetro resto, passado por referência, o resto da divisão.

Ex: int r, d;
d = divisao(5, 2, &r);
printf("Resultado:%d Resto:%d", d, r); /* Resultado:2 Resto:1 */

7. Escrever uma função int somaintervalo(int n1, int n2) que retorna a soma dos números inteiros que existem no intervalo fechado entre n1 e n2. A função deve funcionar inclusive se o valor de n2 for menor que n1.

Ex: n=somaintervalo(3, 6); /* n recebe 18 (3 + 4 + 5 + 6) */
n=somaintervalo(5,5); /* n recebe 5 (5) */
n=somaintervalo(-2,3); /* n recebe 3 (-2 + -1 + 0 + 1 + 2 + 3) */
n=somaintervalo(4, 0); /* n recebe 10 (4 + 3 + 2 + 1 + 0) */

8. Escreva uma função que receba como parâmetro um valor n inteiro e positivo e que calcule a seguinte soma: S = 1 + 1/2 + 1/3 + 1/4 + ... + 1/n. A função deverá retornar o valor de S.

Parte II - Vetores

Desenvolver os respectivos programas em C para resolver os problemas abaixo:

9. Escrever uma função que receba um vetor com 10 valores e retorne quantos destes valores são negativos.

10. Implemente uma função que retorne o maior elemento de um vetor de inteiros de tamanho 10.

11. Implemente uma função que retorne o menor elemento de um vetor de inteiros de tamanho 10.

12. Implemente um procedimento que ordene um vetor de inteiros de tamanho 10.

13. Escrever uma função int somavet(int vetor[], int tamanho), que recebe por parâmetro um vetor de inteiros e o seu tamanho e retorna a soma de seus elementos.

Ex: int lista[4]={100, 20, 10, 5};
int total;
total = somavet(lista, 4); /* total recebe 135 */

14. Implemente uma funç ao que, dado um valor, retorne se esse valor pertence ou não a um vetor de inteiros de tamanho 10.

15. Implemente uma função que retorne a média dos valores armazenados em um vetor de inteiros de tamanho 10.

16. Escrever uma função int so_positivo(int vetor[], int tamanho), que substitui por zero todos os números negativos do vetor passado por parâmetro, sendo que o número de elementos do vetor é passado para a função no parâmetro tamanho. A função deve retornar o número de valores que foram substituídos.

Ex: int v[5] = {3, -5, 2, -1, 4};
tr = so_positivo(v,5); printf("%d", tr); /* 2 */

Parte III - Vetores de caracteres

Desenvolver os respectivos programas em C para resolver os problemas abaixo:

17. Escreva uma função int contc(char str[], char c) que retorna o número de vezes que o caracter c aparece na string str, ambos passados como parâmetros.

Ex: char texto[]="EXEMPLO";
x=contc(texto,'E'); /* x recebe 2 */
x=contc(texto,'L'); /* x recebe 1 */
x=contc(texto,'W'); /* x recebe 0 */

18. Escrever um procedimento void stringup(char destino[], char origem[]), que copia todos os caracteres da string origem para destino, convertendo-os para maiúscula.

Ex: char s1[20], s2[20]="aula de c";
stringup(s1, s2);
printf("%s", s1); /* AULA DE C */

19. Escrever uma função int ultima(char string[], char c) que retorna qual a última posição na string em que aparece o caracter c. Se o caracter não estiver na string, retornar -1.

Ex: char str[]="teste";
int q;
q=ultima(str, 't'); /* q recebe 3 */
q=ultima(str, 'x'); /* q recebe -1 */

20. Escrever uma função int contabranco(char string[]), que retorna o número de espaços em branco contidos na string passada como parâmetro.

Ex: n = contabrancos("a b c"); /* n recebe 3 */
n = contabrancos("abc "); /* n recebe 2 */
n = contabrancos("abc"); /* n recebe 0 */

21. Escrever um procedimento void ninvert(char destino[], char origem[], int num), que copia invertido para a string destino, os num primeiros caracteres da string origem. Se num for maior que o tamanho da string, copiar todos os caracteres.

EX: char s1[80] = "ABCDE";
char s2[80];
ninvert(s2, s1, 3); /* s2 = "CBA"*/
ninvert(s2, s1, 10); /* s2 = "EDCBA"*/

22. Escrever um procedimento void copiaate(char destino[], char origem[], char parar) que copia para a string destino os caracteres da string origem que estão antes da primeira ocorrência do caracter parar ou até o final de origem, se parar não for encontrado.

Ex: char str[80];
copiaate(str, "testando a funcao", 'a'); /* str recebe "test"*/
copiaate(str, "testando a funcao", 'n'); /* str recebe "testa"*/
copiaate(str, "testando a funcao", 'o'); /* str recebe "testand"*/