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.

```c
#include<stdio.h>

int idade(int ano, int meses, int dias) {
  int ano_dias = ano * 365;
  int meses_dias = meses * 30;

  return (ano_dias + meses_dias + dias);

}

int main() {

  int ano, meses, dias, idade_dias;

  printf("Digite sua idade em ano, meses e dias \n");
  printf("Ano: ");
  scanf("%d", &ano);
  printf("Meses: ");
  scanf("%d", &meses);
  printf("Dias: ");
  scanf("%d", &dias);


  idade_dias = idade(ano, meses, dias);

  printf("Idade em dias: %d \n", idade_dias);

  return 0;
}
```

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 |

```c
#include <stdio.h>
#include <stdlib.h>

char conceito (float nota){

    if(nota >= 0 && nota <= 49)
       return 'D';
    else if (nota > 49 && nota <= 69)
       return 'C';
    else if (nota > 69 && nota <= 89)
       return 'B';
    else if (nota > 89 && nota <= 100)
       return 'A';
```

```
    else
        printf("\nNota invalida.\n");
}

int main()
{

    float nota;

    printf("Digite a nota: ");
    scanf("%f",&nota);

    printf("O conceito e: %c\n\n",conceito(nota));

}
```

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

```
#include<stdio.h>
#include<math.h>

float volume(float  raio) {
  float v = 4/(3*3.14*pow(raio,3));
  return v;
}

int main() {

  float raio, v;

  printf("Digite o raio da circunferencia: ");
  scanf("%f", &raio);
  v = volume(raio);
  printf("O volume da circuferencia com raio %.2f eh: %.2f \n", raio, v);

  return 0;

}
```

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) */

```
#include <stdio.h>
#include <stdlib.h>

int contaimpar (int n1, int n2){

    int aux, i;
    int impares = 0;

    if(n2 < n1){
       aux = n1;
       n1 = n2;
       n2 = aux;
    }

    for(i=n1; i<=n2; i++){
        if(i%2==1)
           impares++;
    }

    return impares;
```

```c
}

int main()
{

    int n1, n2;
    printf("Digite o valor de n1: ");
    scanf("%d",&n1);
    printf("Digite o valor de n2: ");
    scanf("%d",&n2);

    printf("O numero de valores impares entre %d e %d e' %d\n\n",n1,n2,contaimpar(n1,n2))
        ;

}
```

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. */

```c
#include<stdio.h>

void estacao(int dia, int mes) {

  int num_estacao = -1; // 1-Verao, 2-Outono, 3-Invero, 4-Primavera

  if(mes <= 3) {
    num_estacao = 1;
    if(mes ==3 && dia >= 21) {
      num_estacao = 2;
    }
  }
  else if(mes > 3 && mes <= 6) {
    num_estacao = 2;
    if(mes == 6 && dia >= 21) {
      num_estacao = 3;
    }
  }
  else if(mes > 6 && mes <= 9) {
    num_estacao = 3;
    if(mes == 9 && dia >= 23) {
      num_estacao = 4;
    }
  }
  else if(mes > 9 && mes <= 12) {
    num_estacao = 4;
    if(mes == 12 && dia >= 21) {
      num_estacao = 1;
    }
  }

  if(num_estacao == 1)
    printf("Verao\n");
  else if(num_estacao == 2)
    printf("Outono\n");
  else if(num_estacao == 3)
    printf("Inverno\n");
  else if(num_estacao == 4)
    printf("Primavera\n");
  else
    printf("Data invalida\n");


}

int main() {
```

```
47    int dia, mes;
48    printf("Digite o dia e o mes: ");
49    scanf("%d %d", &dia, &mes);
50    estacao(dia, mes);
51
52    return 0;
53
54 }
```

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 parmêtro 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 */

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  int divisao (int dividendo, int divisor, int *resto){
6
7     int resultado;
8
9     resultado = floor(dividendo/divisor);
10
11    *resto = dividendo - resultado*divisor;
12
13    return resultado;
14 }
15
16 int main()
17 {
18
19     int dividendo, divisor, resto;
20     printf("Digite o valor do dividendo: ");
21     scanf("%d",&dividendo);
22     printf("Digite o valor do divisor: ");
23     scanf("%d",&divisor);
24
25     printf("\nA divisao inteira de %d por %d e' %d",dividendo,divisor,divisao(dividendo,
           divisor,&resto));
26     printf(" com resto %d\n\n",resto);
27
28 }
```

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) */

```
1  #include<stdio.h>
2
3  int somaintervalo(int n1, int n2) {
4
5    int menor, maior, i, soma = 0;
6
7    if(n1 < n2) {
8      menor = n1;
9      maior = n2;
10   }
```

```
  else {
    menor = n2;
    maior = n1;
  }

  for(i = menor; i <= maior; i++) {
    soma = soma + i;
  }

  return soma;


}

int main() {
  int numero1, numero2;
  printf("Digite dois numeros: ");
  scanf("%d %d", &numero1, &numero2);
  int soma = somaintervalo(numero1, numero2);
  printf("Soma entre o intervalo %d e %d: %d \n", numero1, numero2, soma);

  return 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.

```
#include <stdio.h>
#include <stdlib.h>

float S (int n){

   float resultado = 0;
   int i;

   for(i=1; i<=n; i++)
      resultado = resultado + 1.0/i;

   return resultado;
}

int main()
{

    int n;
    printf("Digite o valor de n: ");
    scanf("%d",&n);

    printf("\nS = %f\n\n",S(n));

}
```

**Parte III - 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.

```
#include<stdio.h>

int negativos(int vetor[]) {

  int i, count_negativos = 0;

  for(i = 0; i < 10; i++) {
    if(vetor[i] < 0)
```

```
        count_negativos++;
    }

    return count_negativos;

}

int main() {

    int v[10] = {1, 2, -2, -3, 4, 5, -1, 2, 9, 0};
    int num_negativos = negativos(v);

    printf("Quantidade de numeros negativos: %d \n", num_negativos);

    return 0;

}
```

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

```
#include <stdio.h>
#include <stdlib.h>


int maior_elemento(int vetor[]){

    int maior, i;

    maior = vetor[0];

    for(i=1; i<10; i++){
        if(vetor[i] > maior)
            maior = vetor[i];
    }

    return maior;
}


int main()
{

    int vetor[10], maior, menor;
    int i;

     for(i=0; i<10; i++){
        printf("Digite o %do elemento do vetor: ",i+1);
        scanf("%d",&vetor[i]);
    }

    printf("\nO maior elemento e' %d\n\n",maior_elemento(vetor));

}
```

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

```
#include<stdio.h>

int menor(int vetor[]) {
  int i, count_menor = 0;
  int menor = vetor[0];
  for(i = 1; i < 10; i++) {
    if(vetor[i] < menor)
      menor = vetor[i];
  }

  return menor;
```

```
12
13
14 }
15
16 int main() {
17
18   int v[10] = {1, 2, -2, -3, 4, 5, -1, 2, 9, 0};
19   int num_menor = menor(v);
20
21   printf("O menor numero eh: %d \n", num_menor);
22
23   return 0;
24
25 }
```

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

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  // Função que encontra o maior elemento do vetor com tamanho 'tamanho' e guarda o índice
       deste elemento na variável 'índice'
5  int maior_elemento(int vetor[], int tamanho, int *indice){
6
7      int maior, i;
8
9      maior = vetor[0];
10     *indice = 0;
11
12     for(i=1; i<tamanho; i++){
13         if(vetor[i] > maior){
14             maior = vetor[i];
15             *indice = i;
16         }
17     }
18
19     return maior;
20 }
21
22
23 void ordena(int vetor[]){
24
25     int aux[10]; // Vetor auxiliar que armazena os elementos que ainda não foram
           ordenados.
26     int exclui;  // Variável auxiliar para retirada do maior elemento do vetor.
27     int tamanho; // Tamanho do vetor resultante após a retirada do maior elemento.
28     int indice;  // Indice do maior elemento do vetor
29     int ordenado[10];  // Vetor final ordenado
30     int i;
31
32     for(i=0; i<10; i++)
33         aux[i] = vetor[i];
34
35     tamanho = 10;
36     for(i=9; i>=0; i--){
37         ordenado[i] = maior_elemento(aux,tamanho,&indice);
38
39         // Nessa parte queremos retirar o maior elemento encontrado no vetor aux. Para
               isso troca-se a posição deste
40         // elemento com o último elemento do vetor aux. Dessa forma o elemento que
               queremos excluir estara na última
41         // posição, bastando portanto subtraírmos 1 do tamanho do vetor para que ele seja
                excluído.
42         exclui = aux[tamanho-1];
43         aux[tamanho-1] = vetor[indice];
44         aux[indice] = exclui;
45         tamanho--;
46     }
47
```

```
    for(i=0; i<10; i++)
        vetor[i] = ordenado[i];

}

int main()
{

    int vetor[10];
    int i,j;

    for(i=0; i<10; i++){
        printf("Digite o %do elemento do vetor: ",i+1);
        scanf("%d",&vetor[i]);
    }

    ordena(vetor);

    printf("\n");
    for(i=0; i<10; i++)
        printf("%d ",vetor[i]);
    printf("\n\n");
}
```

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 */

```
#include<stdio.h>

int somavet(int vetor[], int tamanho) {
  int i, soma = 0;
  for(i = 0; i < tamanho; i++) {
    soma = soma + vetor[i];
  }

  return soma;
}

int main() {

  int v[4] = {100, 20, 10, 5};
  int num_soma = somavet(v, 4);

  printf("A soma do vetor eh: %d \n", num_soma);

  return 0;

}
```

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

```
#include <stdio.h>
#include <stdlib.h>

int busca(int vetor[], int valor){

    int i;

    for(i=0; i<10; i++){
        if(vetor[i] == valor)
            return 1;
    }
```

```
12
13      return 0;
14  }
15
16  int main()
17  {
18
19      int vetor[10], valor;
20      int i,j;
21
22      for(i=0; i<10; i++){
23          printf("Digite o %do elemento do vetor: ",i+1);
24          scanf("%d",&vetor[i]);
25      }
26
27      printf("Digite o valor a ser procurado: ");
28      scanf("%d",&valor);
29
30      if(busca(vetor,valor))
31          printf("\nValor %d pertence ao vetor\n\n");
32      else
33          printf("\nValor %d nao pertence ao vetor\n\n");
34  }
```

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

```
1  #include<stdio.h>
2
3  float mediavet(int vetor[]) {
4
5    float media;
6    int i, soma = 0;
7
8    for(i = 0; i < 10; i++) {
9      soma = soma + vetor[i];
10   }
11
12   media = soma/10.0;
13
14   return media;
15 }
16
17 int main() {
18
19   int v[10] = {1, 2, -2, -3, 4, 5, -1, 2, 9, 0};
20   float media = mediavet(v);
21
22   printf("A media do vetor eh: %.2f \n", media);
23
24   return 0;
25
26 }
```

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 */

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int so_positivo(int vetor[], int tamanho){
5
6      int i;
```

```
    int resultado = 0;

    for(i=0; i<tamanho; i++){
        if(vetor[i] < 0){
            vetor[i] = 0;
            resultado++;
        }

    }

    return resultado;
}

int main()
{

    int tamanho;
    int i,j;
    int resultado;

    printf("Digite o tamanho do vetor: ");
    scanf("%d",&tamanho);

    int vetor[tamanho];

    for(i=0; i<tamanho; i++){
        printf("Digite o %do elemento do vetor: ",i+1);
        scanf("%d",&vetor[i]);
    }

    resultado = so_positivo(vetor,tamanho);
    printf("\nNovo vetor: ");
    for(i=0; i<tamanho; i++)
        printf("%d ",vetor[i]);
    printf("\nNumero de valores substituidos: %d", resultado);

}
```

**Parte IV - 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 */

```
#include<stdio.h>

int contc(char str[], char c) {

  int i, count_ch = 0;

  for(i = 0; str[i] != '\0'; i++) {
    if(str[i] == c)
      count_ch++;
  }

  return count_ch;

}

int main() {

  char texto[] = "EXEMPLO";

```

```c
  int x = contc(texto, 'E');

  printf("O texto contem %d E. \n", x);

  return 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 */

```c
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

void stringup(char destino[], char origem[]){

    int i;

    for(i=0; origem[i] != '\0'; i++)
        destino[i] = toupper(origem[i]);
    destino[i] = '\0';
}

int main()
{

    char origem[100], destino[100];

    printf("Digite uma string: ");
    fgets(origem, 100, stdin);

    stringup(destino,origem);
    printf("\nString convertida: ");
    puts(destino);
}
```

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 */

```c
#include<stdio.h>

int ultima(char str[], char c) {
  int i, posicao = -1;

  for(i = 0; str[i] != '\0'; i++) {
    if(str[i] == c)
      posicao = i;
  }

  return posicao;
}

int main() {
  char str[] = "teste";
  int q;

  q = ultima(str, 'x');
```

```
19
20    printf("Ultima posicao do ch: %d \n", q);
21
22    return 0;
23
24 }
```

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 */

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h>
4
5  int contabrancos(char string[]){
6
7      int i;
8      int resultado = 0;
9
10     for(i=0; string[i] != '\0'; i++){
11         if(string[i] == ' ')
12             resultado++;
13     }
14
15     return resultado;
16 }
17
18 int main()
19 {
20
21     char string[100];
22     int brancos;
23
24     printf("Digite uma string: ");
25     fgets(string, 100, stdin);
26
27     brancos = contabrancos(string);
28     printf("\nA string contem %d espacos em branco.\n\n", brancos);
29
30 }
```

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"*/

```
1  #include<stdio.h>
2  #include<string.h>
3
4  void ninvert(char destino[], char origem[], int num) {
5
6    int i, controle;
7
8    if(num > strlen(origem)) {
9      num = strlen(origem);
10   }
11
12   controle = num-1;
```

```
13
14
15    for(i = 0; i < num; i++) {
16      destino[controle] = origem[i];
17      controle--;
18    }
19    destino[i] = '\0';
20
21  }
22
23  int main() {
24    char s1[80] = "ABCDE";
25    char s2[80];
26    ninvert(s2, s1, 3);
27    int i;
28    puts(s2);
29
30    return 0;
31
32  }
```

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"*/

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h>
4
5  void copiaate(char destino[], char origem[], char parar){
6
7      int i = 0;;
8
9      while(origem[i] != parar && origem[i] != '\0'){
10         destino[i] = origem[i];
11         i++;
12     }
13
14     destino[i] = '\0';
15
16 }
17
18 int main()
19 {
20
21     char origem[100];
22     char destino[100];
23     char parar;
24
25     printf("Digite uma string: ");
26     fgets(origem, 100, stdin);
27
28     printf("Digite o carcter de termino: ");
29     scanf("%c",&parar);
30     copiaate(destino,origem,parar);
31
32     printf("\nString copiada: ");
33     puts(destino);
34
35 }
```