

**UNIVERSIDADE ESTADUAL DO MATO GROSSO DO SUL
CURSO DE SISTEMAS DE INFORMAÇÃO**

LISTA DE EXERCÍCIOS DE FUNÇÕES E PROCEDIMENTOS

1. Escreva uma função que recebe por parâmetro um valor inteiro e positivo N e retorna o valor de S.

$$S = 1 + 1/1! + 1/2! + 1/3! + 1/N!$$

2. Crie um procedimento que receba um caractere como parâmetro e informe (escreva) se tal caractere é uma consoante ou uma vogal.

```
// Funcao busca: Recebe x, n >= 0 e v e devolve um índice k em 0..n-1 tal que  
// x == v[k].
```

```
// Se tal k não existe, devolve -1.
```

```
int busca (int x, int n, int v[]) {
```

```
    int k;
```

```
    k = n-1;
```

```
    while (k >= 0 && v[k] != x)
```

```
        k --;
```

```
    return k;
```

```
}
```

```
// Funcao remove: Recebe um vetor v[0..n-1] e um índice k tal que 0 <= k < n.
```

```
// Devolve v[k] e remove esse elemento do vetor.
```

```
int remove (int k, int n, int v[]) {
```

```
    int i, x = v[k];
```

```
    for ( i = k+1; i < n; i++)
```

```
        v[i-1] = v[i];
```

```
    return x;
```

```
}
```

```
//Funcao insere: Insere x entre as posições k-1 e k do vetor v[0..n-1] supondo
```

```
// que 0 <= k <= n.
```

```
void insere (int k, int x, int n, int v[]) {
```

```
    int i;
```

```
    for ( i = n; i > k; i--)
```

```
        v[i] = v[i-1];
```

```
    v[k] = x;
```

```
}
```

3. Faça um programa que leia um vetor de n<=10 elementos e remova em n passos cada um dos elementos do vetor. Imprima o antes e o depois das remoções.

4. Faça um programa que leia um vetor V1 de $n \leq 10$ elementos e um vetor V2 de $m \leq 10$ elementos. O programa deverá buscar cada elemento de V2 no vetor V1 e se não o encontrar deverá inseri-lo no vetor V1.
5. Modifique a função insere para que incremente o valor de n depois da inserção.
6. Modifique a função remove para que decremente o valor de n depois da remoção.
7. Modifique a função busca para que ela percorra o vetor do início ao fim.
8. O quê acontece na função remove se trocarmos $v[i-1]=v[i]$ por $v[i]=v[i+1]$.
9. Escreva uma função que calcule o comprimento do mais longo segmento de zeros (ou carreira de zeros) em um vetor de números inteiros.
10. Faça uma função que elimine todos os elementos nulos de $v[0..n-1]$.
11. A função insere2, insere um elemento no vetor v de tal forma que v, já estando ordenado, se mantenha ordenado.

```
void insere2 (int v[], int n, int x) {
    int k;
    for ( k=n-1; k>=0 && v[k] > x; k--)
        v[k+1] = v[k];
    v[k+1] = x;
}
```

Fale da diferença entre a função insere e a função insere2.

12. Faça um programa que leia vários alunos e seus dados (segundo estrutura abaixo) e os mantenha ordenados em um vetor. O programa deve usar uma função que permite a inserção de um aluno em um vetor já ordenado de alunos. A chave de ordenação é a nota. Use a struct abaixo:

```
typedef struct{
    int rgm;
    int idade;
    int nota;
}aluno;
```

```
// Funcao min: Recebe um vetor v[0..n-1] e devolve o elemento mínimo deste
// conjunto de n elementos
int min (int n, int v[]) {
    int j, min = v[0];
    for (j = 1; j < n; ++j)
        // neste ponto, min é um
```

```

        // elemento mínimo de v[0..j-1]
        if (v[j] < min)
            min = v[j];
    return min;
}

int ind_min (int k, int n, int v[]) {
    int j, ind=k;
    for (j = k+1; j < n; ++j)
        // neste ponto, min é um
        // elemento mínimo de v[k..j-1]
        if (v[j] < v[ind])
            ind= j;
    return ind;
}

```

13. Fale da diferença entre a função min e a função ind_min (acima).

```

int buscaBinaria(int vet[], int n, int chave) {
    int posIni = 0, posFim = n - 1, posMeio;
    while (posIni <= posFim) {
        posMeio = (posIni + posFim) / 2;
        if (vet[posMeio] == chave)
            return posMeio;
        else if (vet[posMeio] > chave)
            posFim = posMeio - 1;
        else
            posIni = posMeio + 1;
    }
    return -1;
}

```

14. Explique a diferença entre a função busca e buscaBinaria.

15. Escreva um programa que faça o busca binária de um cpf em um vetor de dados de contribuintes da receita federal (crie um vetor struct).

16. Dada uma sequência de caracteres informada pelo usuário, verificar se a sequência trata-se de um palíndromo ou não. Desenvolva um módulo para verificar isto.

17. Faça uma função que inverta os elementos de um vetor.

ex: vetor antes de inverter

5	3	2	7
---	---	---	---

vetor depois de inverter

7	2	3	5
---	---	---	---

Obs: os parâmetros de Entrada da função devem ser o vetor e número de elementos armazenados nesse vetor. A função deve retornar o vetor invertido.

18. Faça um programa que manipule um conjunto de registro que será digitado pelo usuário com os seguintes dados:

NOME:

ENDEREÇO:

IDADE:

CEP:

TRABALHA (SIM ou NÃO):

ESTUDA(SIM ou NÃO):

SEXO:

**** passos a serem realizados ****

1° - uma função que faça a inserção de dados nos registros.

2° - uma função que retorne a posição de uma busca no registro (com base no nome a procurar).

3° - uma função que apague um item da lista de registros, para isto o algoritmo terá que primeiro fazer uma busca (claro que vai verificar se existe o nome procurado) e retornar a posição.

4° - uma função que mostre a porcentagem de pessoas que tenham 18 anos, que estudam e trabalham e que possuem o sexo masculino.

5° - uma função que devolva o CEP onde possuem a maior incidência de pessoas.

19. Faça uma função para inverter uma matriz.

ex:

```
A  a11  a12  a13  a14
    a21  a22  a23  a24
    a31  a32  a33  a34
    a41  a42  a43  a44
```

Inverso de A

```
  a11  a21  a31  a41
  a12  a22  a32  a42
  a13  a23  a33  a43
  a14  a24  a34  a44
```

Obs: os parâmetros de Entrada da função devem ser a matriz, número de linhas e número de colunas dessa matriz. A função deve retornar a matriz invertida.

20. Escreva uma função simétrica (`double a[][maxcol], int n`) que verifica se a matriz a $n \times n$ é uma matriz simétrica, isto é, se $a[i][j]$ é igual a $a[j][i]$.
21. Escreva uma função `multmatvet(double a[][mc], double b[], double c[], int n, int m)` que recebe a matriz a de $n \times m$ elementos, o vetor b de m e devolve o vetor c de n elementos que é a multiplicação de a por b .
22. Escreva uma função `ind_min_col(double a[][maxc], int n, int col)` que receba a matriz a de $n \times n$ elementos e devolva como resultado o índice do menor elemento da coluna col desta matriz.

23. Faça uma função que desenhe uma árvore de asterisco na tela.

```
  *
 * *
* * *
* * * *
```

OBS: não usar matriz

24. Faça uma função que desenhe uma árvore de asterisco na tela:

```
  *
 * *
* * *
* * * *
* * * *
 * * *
  * *
   *
```

OBS: não usar matriz.