

RESOLUÇÃO DA LISTA 03

1- Leia 10 números inteiros e armazene em um vetor. Em seguida escreva os elementos que são primos e suas respectivas posições no vetor.

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

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

```
int main(){
```

```
    printf("\tLISTA 3 - Questao 1 \n\n");
```

```
    int v[10] ;
```

```
    int d,i,limite;
```

```
    int primo=1;
```

```
    for(i=0; i<10 ;i++) {
```

```
        printf("\nEntre com os numeros inteiros: ");
```

```
        scanf("%d",&v[i]);
```

```
        if (v[i] > 1) {
```

```
            d = 2;
```

```
            primo = 1;
```

```
            limite = sqrt(v[i]);
```

```
            while(primo && d <= limite)
```

```
            {
```

```
                if (v[i] % d == 0){
```

```
                    primo = 0;
```

```
                }
```

```
            d++;
```

```

    }

    if (primo) // é o mesmo que verifica == 1
        printf("\nVetor[%d]:%d eh primo.\n",i, v[i]);
    else printf("\nVetor[%d]:%d nao eh primo.\n", v[i],i);
}
}

return 0;
}

```

2- Faça um programa que receba do usuário dois vetores, A e B, com 10 números inteiros cada. Crie um novo vetor denominado C calculando $C=A-B$. Mostre na tela os dados do vetor C.

```

#include<stdio.h>

#include<stdlib.h>

int main() {
printf("\tLISTA 3 - Questao 2 \n\n");

int a[10],b[10],c[10];

int i,j;

for(i=0;i<10;i++){

    printf("\n\nEntre com os dados no vetor A [%d]: ",i);

    scanf("%d",&a[i]);

    printf("\n\nEntre com os dados no vetor B [%d]: ",i);

    scanf("%d",&b[i]);

    c[i]=a[i]-b[i];

}

for(j=0;j<10;j++)

    printf("\nVetor C[%d]:%d",j,c[j]);

return 0;

}

```

3- Leia duas matrizes 4x4 e escreva uma terceira com os maiores valores de cada posição das matrizes lidas.

```

#include<stdio.h>

#include<stdlib.h>

```

```

int main() {

printf("\tLISTA 3 - Questao 3 \n\n");

int A[4][4];

int B[4][4];

int C[4][4];

int i,j;

int maior=-1;


printf("\n*** Entrada da Matriz A ***\n\n");

for(i=0;i<4;i++){

for(j=0;j<4;j++){

printf("Matriz A(%d,%d): ",i,j);

scanf("%d",&A[i][j]);

}

}

printf("\n*** Entrada da Matriz B ***\n\n");

for(i=0;i<4;i++){

for(j=0;j<4;j++){

printf("Matriz B(%d,%d): ",i,j);

scanf("%d",&B[i][j]);

}

}

printf("\n*** Os maiores valores de cada posicao sao: ***\n\n");

for(i=0;i<4;i++){

for(j=0;j<4;j++){

if (A[i][j]>B[i][j])

C[i][j]=A[i][j];

else C[i][j]=B[i][j];


printf("Matriz C(%d,%d): %d\t",i,j,C[i][j]);

} printf("\n\n");

}

return 0;

```

```
}
```

4- Faça um programa que leia uma matriz 3x6 com valores reais.

(a) Imprima a soma de todos os elementos das colunas ímpares;

(b) Imprima a média aritmética dos elementos da segunda e quarta colunas;

(c) Substitua os valores da sexta coluna pela soma dos valores das colunas 1 e 2;

(d) Imprima a matriz modificada.

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

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

```
int main() {
```

```
printf("\tLISTA 3 - Questao 4 \n\n");
```

```
float m[3][6];
```

```
int i,j;
```

```
float m1,m2,s1,s2,s3;
```

```
printf("\n*** Entrada da Matriz ***\n\n");
```

```
for(i=0;i<3;i++){
```

```
    for(j=0;j<6;j++){
```

```
        printf("Matriz(%d,%d): ",i,j);
```

```
        scanf("%f",&m[i][j]);
```

```
    }
```

```
}
```

```
for(i=0;i<3;i++){
```

```
    for(j=0;j<6;j++){
```

```
        s1+=m[i][0];
```

```
        s2+=m[i][2];
```

```
        s3+=m[i][4];
```

```
        m1+=m[i][1];
```

```
        m2+=m[i][3];
```

```
        m[i][5]=m[i][0]+m[i][1];
```

```
    }
```

```
}
```

```
printf("\n\nSoma da coluna 1: %.f | Soma da coluna 3: %.f | Soma da coluna 5: %.f",s1,s2,s3);
```

```

printf("\n\nA media aritmetica da coluna 2: %.f | A media aritmetica da coluna 4: %.f",m1,m2);

printf("\n\nColuna Modificada:\n\n");

    for(i=0;i<3;i++){

        for(j=0;j<6;j++){

            printf("%.f\t",m[i][j]);

        } printf("\n\n");

    }

    return 0;

}

```

5- Escrever um algoritmo e um programa em C que produza na tela um triângulo de Pascal de grau “n” usando uma

matriz. Abaixo temos um triângulo de Pascal de grau 6 (isto é, com seis linhas):

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

```

Os elementos extremos em cada linha são iguais a 1. Os outros são obtidos somando-se os dois valores que aparecem

imediatamente acima e à esquerda na linha anterior. Exemplo: O quarto elemento da linha corresponde a soma do quarto e do terceiro elemento na linha anterior, isto é, $4 = 1 + 3$.

(Suponha: uma matriz quadrada de dimensão nqualquer, ou seja, n será fornecido pelo usuário do programa)

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

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

```
int main() {
```

```
    printf("\tLISTA 3 - Questao 5 \n\n");
```

```
    int n,i,j;
```

```
    int a[n][n];
```

```
        printf("Defina a ordem da matriz: ");
```

```
        scanf("%d",&n);
```

```

printf("\n*** Triangulo de Pascal ***\n\n");

for(i=0;i<n;i++){
for(j=0;j<=i;j++){
a[i][j]=0;
    a[i][i]=1;
    a[i][0]=1;
    if(a[i][j]==0)
        a[i][j]=a[i-1][j-1]+a[i-1][j+1];
    printf("%d\t",a[i][j]);

}
printf("\n\n");
}
return 0;
}

```

6- Faça um programa que receba uma palavra e a imprima de trás-para-frente.

```

#include <stdio.h>

#include <string.h>

int main() {
    printf("\tLISTA 3 - Questao 6 \n\n");

    char p[10];

    int i, j;

    printf("\nEntre com uma palavra: ");

    gets(p);

    j = strlen(p);

    printf("\nA palavra invertida eh:");

    for(i = j; i >= 0; i--) {
        printf("%c", p[i]);
    }

    printf("\n");

    return 0;
}

```

7- Faça um programa que receba um nome e retorne quantas letras tem esse nome.

```

#include<stdio.h>

```

```

#include<stdlib.h>

#include<string.h>

int main() {

printf("\tLISTA 3 - Questao 7 \n\n");

    char nome[10];

    int i;

    printf("Entre com o nome: ");

    gets(nome);

    i=strlen(nome);

    printf("\nO nome: %s, tem %d letras\n",nome,i);

        return 0;

}

```

8- Escreva um programa que leia a idade e o primeiro nome de até 10 pessoas. Seu programa deve terminar quando uma idade negativa for digitada. Ao terminar, seu programa deve escrever o nome e a idade da pessoa mais jovem e da mais velha.

```

#include <stdio.h>

struct Pessoa {

    char n[10];

    int id;

};

int main() {

    printf("\tLISTA 3 - Questao 8 \n\n");

    printf("\n*** Adicione as informacoes (\"IDADE NEGATIVA FINALIZA PROGRAMA\") ***\n");

    printf("\n");

    struct Pessoa p[10];

    int menor = +999, maior = -999;

    for(int i=0 ; i < 10; i++) {

        printf("\nEntre com um nome: ");

        scanf("%s", &p[i].n);

        printf("Entre com a idade: ");

        scanf("%d", &p[i].id);

        if(p[i].id < 0)

            break;
    }
}

```

```

        if(p[i].id > maior)

            maior = p[i].id;

        if(p[i].id < menor)

            menor = p[i].id;
    }

    printf("\n Pessoa mais nova: \n");

    for(int i=0 ; i < 10; i++) {

        if(p[i].id < 0)

            break;

        if(p[i].id == menor)

            printf("Nome: %s   Idade: %d anos\n", p[i].n, p[i].id);
    }

    printf("\n Pessoa mais velha: \n");

    for(int i=0 ; i < 10; i++) {

        if(p[i].id < 0) break;

        if(p[i].id == maior)

            printf("Nome: %s   Idade: %d anos\n", p[i].n, p[i].id);
    }

    return 0;

}

```

9- Fazer um programa de “criptografia” (codificação de dados visando a privacidade de acesso as informações), onde dada uma string este programa codifique os dados através de um processo de substituição de letras. Você pode definir o seu próprio método de criptografia, desde que depois seja possível reverter este processo, ou seja, um código criptografado deve poder ser convertido novamente ao valor inicial). Fazer um outro programa complementar a este que deve ser capaz de descriptografar a string, ou seja, deve pegar uma string codificada e retornar ao texto original. Dica: use uma matriz que seja inversível.

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

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

```
int main() {
```

```
printf("\tLISTA 3 - Questao 9 \n\n");
```

```
char p[10],c[10];
```

```
int i=0;
```

```
int opcao=0;
```



```

while(opcao!=3){

    printf("\n\t MENU DE OPCOES:\n");

    printf("\n1-Insercao da palavra");

    printf("\n2-Criptografia & Descriptografia da palavra");

    printf("\n3-sair");

    printf("\n\nQual sua opcao: ");

    scanf("%d",&opcao);

    getchar();

    switch(opcao){

        case 1:

            printf("\n\tENTRADA DA PALAVRA\n");

            printf("\nEntre com a palavra a ser criptografada: ");

            gets(p);

            printf("\nPALAVRA \"%s\" INSERIDA !\n\n",p);

            break;

            case 2:

                printf("\n\tCRIPTOGRAFIA\n");

                printf("\nA palavra criptografada: ");

                for(i=0;p[i]!='\0';i++)

                    printf("%c",1+p[i]);

                getchar();

                printf("\nA palavra descriptografada: ");

                for(i=0;p[i]!='\0';i++)

                    printf("%c",(1+p[i])-1);

                printf("\n\n");

                getchar();

                break;

            }

        }

    }

return 0;

}

```