

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

```
#define LIMITE 25
```

- 1) Analise o código abaixo e complete os procedimentos que estão destacados
(EM NEGRITO). VALOR (2,0).

```
//Esse é um procedimento utilizado para ler uma matriz.
```

```
// Fique a vontade para acrescentar parâmetros, criar variáveis.
```

```
void lerMatriz(      )
```

```
{
```

```
}
```

```
//Esse é um procedimento utilizado para somar duas matrizes.
```

```
// Fique a vontade para acrescentar parâmetros, criar variáveis.
```

```
void somarMatrizes(      )
```

```
{
```

```
}
```

```
void escreverMatriz(int mat[LIMITE][LIMITE], int m, int n)
```

```
{
```

```
    int i, j;
```

```
    for (i = 0; i < m; i = i + 1)
```

```
    {
```

```
        for (j = 0; j < n; j = j + 1) printf("%3d", mat[i][j]);
```

```
        printf("\n");
```

```
    }
```

```
}
```

```
int main(void)
```

```
{
```

```
    int m, n, m1[LIMITE][LIMITE], m2[LIMITE][LIMITE], m3[LIMITE][LIMITE];
```

```
    printf("Informe o tamanho da primeira dimensao das matrizes\n");
```

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

```
    printf("Informe o tamanho da segunda dimensao das matrizes\n");
```

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

```
    if (m < LIMITE && n < LIMITE)
```

```
    {
```

```

printf("Informe os elementos da primeira matriz\n");
lerMatriz(m1, m, n);

printf("Informe os elementos da segunda matriz\n");
lerMatriz(m2, m, n);
printf("\n");
somarMatrizes(m1, m2, m3, m, n);
printf("Vejam como vai ficar a soma das matrizes\n");
escreverMatriz(m3, m, n);
}
else printf("Matriz excessivamente grande!\n");
system("PAUSE");
return 0;
}

```

- 2) Analise o código abaixo. Comente o mesmo e informe o que será impresso na tela para o usuário. **(VALOR 1,0)**

```

#include <stdio.h>
#include <stdlib.h>
struct aluno {
    char nome[40], float P1, float P2, float T, int faltas;
};
int main(int argc, char *argv[])
{
    struct aluno joao, maria;
    joao.P1 = 9.5;
    joao.P2 = 8.5;
    joao.T = 9.0;
    joao.faltas = 4;
    system("PAUSE");
    return 0;
}

```

- 3) Analise o código abaixo, complete as informações que estão em negrito. Depois crie uma função, procedimento ou dentro da classe MAIN mesmo um código para mostrar os alunos de MAIOR IDADE e MENOR IDADE. **(VALOR 3,0)**.

```

struct aluno
{

};
struct aluno vetoraluno[3];
int x;
int main(int argc, char *argv[])

```

```

{
    while (x<4)
    //for (x=0; x<3;x++)
    {
        printf("Digite a idade do aluno numero %d\n", x+1);
        scanf("%d", &vetoraluno[x].idade);
        printf("Digite o nome do aluno numero %d\n", x+1);
        gets(vetoraluno[x].nome);
        printf("Digite o cpf do aluno numero\n");
        gets(vetoraluno[x].cpf);
        printf("Digite o telefone do aluno numero %d\n", x+1);
        gets(vetoraluno[x].tel);
        printf("Digite o sexo do aluno numero %d\n", x+1);
        gets(vetoraluno[x].sexo);

        x++;
    }

    system("PAUSE");
    return 0;
}

```

- 4) Analise o vetor abaixo. Crie uma função chamada BUSCA VETOR, onde essa função vai receber um valor qualquer vai retornar -1 caso não seja encontrado esse valor no vetor ou retorna a posição em que esse valor está no vetor. EXEMPLO, valor 555 retornaria 1 e valor 12 retornaria -1. **(VALOR 3,0).**

222	555	111	333	444	666	555	888	777
-----	-----	-----	-----	-----	-----	-----	-----	-----

- 5) Analise o código abaixo e informe o que será impresso. **(VALOR 1,0)**

```

main ( )
{
    int mat[LIN][COL],i,j;
    for ( i=1; i<3; i++ )
        for ( j=1; j<3; j++ )
        {
            printf ("Entre com o elemento[%d][%d]", i, j);
            scanf ("%d", &mat[ i ][ j ]);
        }
    for ( i=1; i<3; i++ )
        for ( j=1; j<3; j++ )
            if ( i == j )
                printf ("n", mat[ i ][ j ]);
}

```