

*/\* Programa 1. Em eletrônica usa-se uma técnica conhecida como divisor de tensão para se obter uma tensão elétrica (Vout) menor que outra tensão de referência (Vin). Essa técnica é ilustrada no circuito abaixo, onde temos dois resistores (R1 e R2) ligados em série: (15 pontos)*

*Assim, sabendo-se que Vin é a tensão de entrada do circuito, Vout a tensão de saída, temos que:*

*Tomando como base essa definição, escreva um programa que solicite ao usuário valores para R1, R2 e Vin*

*e, usando a fórmula acima, calcule o valor da tensão de saída Vout.*

*Nesse programa você não deve usar funções. Seu programa deve ter somente a função main.*

*\*/*

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

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

```
int main() {
```

```
    // solicita o valor de R1
```

```
    float r1;
```

```
    printf("Entre com o valor de R1: ");
```

```
    scanf("%f", &r1);
```

```
    // solicita o valor de R2
```

```
    float r2;
```

```
    printf("Entre com o valor de R2: ");
```

```
    scanf("%f", &r2);
```

```
    // solicita o valor da tensão de referência (Vin)
```

```
    float Vin;
```

```
    printf("Entre com o valor da tensao de referencia(Vin): ");
```

```
    scanf("%f", &Vin);
```

```
    // calcula o valor da tensão de saída (Vout)
```

```
    float Vout;
```

```
    Vout = (r2/(r1+r2))* Vin;
```

```
    // informa o valor da tensão de saída
```

```
    printf("O valor da tensao de saida e: %.2f", Vout);
```

```
}
```

```
/*  
Programa 2. Um número é considerado primo se for divisível apenas por 1 e por ele  
mesmo. Assim, tomando  
em consideração as seguintes regras: (15 pontos)
```

- Se o número for maior que 20, o programa deve dizer que o número é inválido.
- Se o número for divisível por 2, 3, 5, 7, 11, 13, 17 ou 19 ...E.. não for um desses números o algoritmo deve dizer que ele não é primo;

```
Com base nessas regras, escreva um programa que verifique se um número menor ou  
igual a 20 é primo.
```

```
*/
```

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

```
int main() {
```

```
    //Solicita o valor
```

```
    int n1;
```

```
    printf("Digite o valor que sera verificado: ");
```

```
    scanf("%d", &n1);
```

```
    if (n1>20) {
```

```
        printf("Erro! Nao e possivel verificar um valor maior que 20. ");
```

```
    } else if ((n1 % 2 == 0 && n1 != 2) || (n1 % 3 == 0 && n1 != 3) || (n1 % 5 == 0  
&& n1 != 5) || (n1 % 7 == 0 && n1 != 7)
```

```
        || (n1 % 11 == 0 && n1 != 11) || (n1 % 13 == 0 && n1 != 13) || (n1 % 15  
        == 0 && n1 != 15) || (n1 % 17 == 0 && n1 != 17) || (n1 % 19 == 0 && n1 != 19))
```

```
    {
```

```
        printf("O numero %d nao e primo! ", n1);
```

```
    } else {
```

```
        printf("O numero %d e primo! ", n1);
```

```
    }
```

```
    return(0);
```

```
}
```

```
/*  
Programa 3. A matriz transposta de A (de dimensão m x n) é a matriz T (de dimensão  
n x m) onde cada  
coluna de T corresponde a uma linha de A. (15 pontos)
```

Por exemplo, dada a matriz A

```
10  20  30  40  
50  60  70  80  
90 100 110 120
```

sua matriz transposta T é

```
10 50 90  
20 60 100  
30 70 110  
40 80 120
```

Tomando como base essa definição, escreva um programa que:

- ? Solicite ao usuário os dados da matriz A, de 3 linhas e 4 colunas.
- ? Construa, a partir de A, a matriz transposta T, de 4 linhas e 3 colunas.
- ? Mostre na tela as duas matrizes.

```
*/
```

```
#include <stdio.h>  
#include <conio.h>  
#define LIM 25
```

```
void transposta(int A[3][4], int T[4][3])  
{  
    int i, j;  
    for (i=0; i<=2; i++)  
        for(j=0; j<=3; j++) T[j][i] = A[i][j];  
}
```

```
int main(void) {  
    //Cria matriz de 3 linhas e 4 colunas  
  
    int i, j, matrizA[3][4], matrizT[4][3];  
  
    //Geração de matriz 3x4 - inclusão dos elementos  
  
    printf("Digite os elementos da matriz: \n");  
  
    for (i=0; i<=2; i++) //Linhas da matriz  
    {  
        for (j=0; j<=3; j++) //Colunas da matriz  
            scanf("%i", &matrizA[i][j]);  
    }  
  
    //Impressão da matriz resultante  
  
    printf("A matriz resultante e: \n");  
    for (i=0; i<=2; i++) //Linhas  
    {  
        for (j=0; j<=3; j++) //Colunas  
            printf("%i ", matrizA[i][j]);  
        printf("\n");  
    }  
}
```

```
    }

    //Impressão da matriz transposta

    transposta(matrizA, matrizT);
    printf("\n \n");
    printf("A matriz transposta e: \n");
    for (i=0; i<=3; i++)
    {
        for (j=0; j<=2; j++)
            printf("%3d ", matrizT[i][j]);
        putchar('\n');
    }

    getch();
    return(0);
}
```

```
/* Aqui vamos melhorar a solução apresentada no Programa 1, movendo o cálculo de Vout para dentro de uma função. Para isso vamos escrever uma função que recebe como entrada os valores de R1, R2 e Vin e, usando a fórmula do Programa 1, calcula e retorna o valor da tensão de saída Vout.. (15 pontos)
```

Pontos a serem observados:

? Sua função não pode fazer a leitura dos valores de R1, R2 e Vin (ela deve receber os valores)

? Sua função não pode mostrar o valor de Vout (esse valor deve ser retornado pela função)

Assim sendo:

? Seu programa deve solicitar os valores de R1, R2 e Vin e chamar a função para obter Vout.

? Uma vez feito isso, ele deve mostrar o valor de Vout.

```
*/
```

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

```
// Função que calcula e retorna o valor da tensão de saída Vout
```

```
float tensaoSaida (float r1, float r2, float Vin){
```

```
    float Vout;
```

```
    Vout = (r2/(r1+r2))* Vin;
```

```
    return Vout;
```

```
}
```

```
int main(){
```

```
    // solicita o valor de R1
```

```
    float R1;
```

```
    printf("Entre com o valor de R1: ");
```

```
    scanf("%f", &R1);
```

```
    // solicita o valor de R2
```

```
    float R2;
```

```
    printf("Entre com o valor de R2: ");
```

```
    scanf("%f", &R2);
```

```
    // solicita o valor da tensão de referência (Vin)
```

```
    float VIN;
```

```
    printf("Entre com o valor da tensao de referencia(Vin): ");
```

```
    scanf("%f", &VIN);
```

```
    // A função é chamada
```

```
    float tensaoS = tensaoSaida (R1, R2, VIN);
```

```
    // informa o valor da tensão de saída
```

```
    printf("O valor da tensao de saida e: %.2f", tensaoS);
```

```
}
```