

1 - Escrever um algoritmo que leia 10 notas em um vetor, exibir a menor nota digitada.

```
#include <stdio.h>

int main() {

    float vector1[10], f;

    int i;

    f = vector1[0];

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

        printf("escreva uma nota \n");

        scanf("%f", &vector1[i]);

    }

    for (i=1; i<10; i++){

        if(vector1[i] < f){

            f = vector1[i];

        }

    }

    printf("o maior valor é %.2f", f);

    return 0;

}
```

2 - Escreva um algoritmo que leia dois vetores inteiros de 10 posições e faça o produto dos elementos de mesmo índice, colocando o resultado em um terceiro vetor. Mostre o vetor resultante.

```
#include <stdio.h>

int main() {

    int vector1[10], vector2[10];

    float vector3[100];

    int i, p, f;

    p=0;
```

```

for(i=0; i<10; i++){
    printf("escreva um valor para o primeiro vetor \n");
    scanf("%d", &vector1[i]);
}

for(i=0; i<10; i++){
    printf("escreva um valor para o segundo vetor \n");
    scanf("%d", &vector2[i]);
}

for(i=0; i<10; i++){
    for(p=0; p<10; p++){
        if(vector1[i] == vector2[p]){
            vector3[f]= vector1[i] * vector2[p];

            printf("o valor do produto dos indices de mesmo valor do primeiro vetor na posição %d e do
segundo vetor na posição %d é igual = %.2f \n", i, p, vector3[f]);
        }
    }
}

return 0;
}

```

3 - Criar um algoritmo que leia uma matriz 4x4 de números inteiros. Em seguida, exiba o produto dos elementos de cada uma das linhas.

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

```

int main()
{
    int matriz [4][4];
    int lin, col;

```

```

int multiplicacao=1;
for (lin=0;lin<4;lin++){
for (col=0; col<4; col++){
printf("Digite um número inteiro \n");
scanf("%d", &matriz[lin][col]);
multiplicacao = multiplicacao * matriz[lin][col];
}
printf ("O multiplicação da linha %d = %d \n", lin, multiplicacao );
multiplicacao = 1;
}

return 0;
}

```

4 - Crie um algoritmo que leia uma matriz 4x4 de números inteiros e calcule o produto dos valores das colunas da matriz.

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

```

int main()
{
int matriz [4][4];

int lin, col;

int multiplicacao=1;
for (lin=0;lin<4;lin++){
for (col=0; col<4; col++){
printf("Digite um número inteiro \n");
scanf("%d", &matriz[lin][col]);
}
}
}

```

```

for(col=0; col<4; col++){
    for(lin=0; lin<4; lin++){
        multiplicacao = multiplicacao * matriz[lin][col];
    }
    printf ("O multiplicação da coluna %d = %d \n", col, multiplicacao );
    multiplicacao = 1;
}

return 0;
}

```

5 - Crie um algoritmo que leia duas matrizes 2x5 e crie uma terceira matriz também 2x5 com o valor da soma dos elementos de mesmo índice.

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

```

int main()
{
    int matriz1[2][5], matriz2[2][5], matriz3[2][5];
    int i, p;
    for(i=0; i<2; i++){
        for(p=0; p<5; p++){
            printf("escreva um valor para a primeira matriz \n");
            scanf("%d", &matriz1[i][p]);
        }
    }
    for(i=0; i<2; i++){
        for(p=0; p<5; p++){
            printf("escreva um valor para a segunda matriz \n");
            scanf("%d", &matriz2[i][p]);
        }
    }
}

```

```

    }
}
for(i=0; i<2; i++){
    for(p=0; p<5; p++){
        matriz3[i][p]= matriz1[i][p] + matriz2[i][p];
    }
}
printf("os valores da soma de numeros de mesmo indice estão presentes na matriz \n ");
for(i=0; i<2; i++){
    for(p=0; p<5; p++){
        printf("%d", matriz3[i][p]);
    }
    printf("\n");
}

return 0;
}

```

6 - Faça um algoritmo para ler uma matriz 3X4 e depois gerar e imprimir sua transposta (matriz 4X3 equivalente).

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

```

int main()
{
    int matriz[3][4];
    int lin, col;
    for(lin=0; lin<3; lin++){
        for(col=0; col<4; col++){
            printf("escreva um valor para a matriz \n");

```

```

        scanf("%d", &matriz[lin][col]);
    }
}
printf("essa é a matriz escrita \n");
for(lin=0; lin<3; lin++){
    for(col=0; col<4; col++){
        printf("%d", matriz[lin][col]);
    }
    printf("\n");
}
printf("essa é a matriz transposta \n");
for(col=0; col<4; col++){
    for(lin=0; lin<3; lin++){
        printf("%d", matriz[lin][col]);
    }
    printf("\n");
}

return 0;
}

```

7 - Crie um algoritmo que calcule o produto dos valores da diagonal principal de uma matriz de inteiros 6x6.

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

```

int main()
{
    int matriz[6][6];
    int lin, col;

```

```

int multiplicacao;
multiplicacao=1;
for(lin=0; lin<6; lin++){
    for(col=0; col<6; col++){
        printf("escreva um valor para a matriz \n");
        scanf("%d", &matriz[lin][col]);
    }
}

for(lin=0; lin<6; lin++){
    multiplicacao = multiplicacao * matriz[lin][lin];
}

printf("o produto da diagonal principal da matriz escrita é igual a = %d \n", multiplicacao);

return 0;
}

```

8 - Crie um algoritmo que leia uma matriz 3x3 de inteiros e calcule o somatório de todos os valores da matriz.

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

```

int main()
{
    int matriz[3][3];
    int lin, col;
    int soma;
    soma=0;
    for(lin=0; lin<3; lin++){
        for(col=0; col<3; col++){
            printf("escreva um valor para a matriz \n");

```

```

        scanf("%d", &matriz[lin][col]);

        soma = soma + matriz[lin][col];
    }
}

printf("a somatoria de todos os valores da matriz escrita é igual a = %d \n", soma);

return 0;
}

```

9 - Escreva uma função sem retorno (função void) que recebes 3 valores reais X, Y e Z e que verifique se esses valores podem ser os comprimentos dos lados de um triângulo e, neste caso, informar qual o tipo de triângulo formado. Para que X, Y e Z formem um triângulo é necessário que a seguinte propriedade seja satisfeita: o comprimento de cada lado de um triângulo é menor do que a soma do comprimento dos outros dois lados. O procedimento deve identificar o tipo de triângulo formado observando as seguintes definições: Triângulo Equilátero: os comprimentos dos 3 lados são iguais. Triângulo Isósceles: os comprimentos de 2 lados são iguais. Triângulo Escaleno: os comprimentos dos 3 lados são diferentes.

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

```

void teste (float x, float y, float z){
    float soma[3];
    soma[0]=x+y;
    soma[1]=y+z;
    soma[2]=z+x;
    if(x>soma[1] || y>soma[2] || z>soma[0]){
        printf("os valores X Y Z não formam um triangulo \n");
    }
    else if(x==y && x==z){
        printf("O triangulo de X Y Z é um triangulo equilátero \n");
    }
    else if(x==y || x==z || z==y){
        printf("O triangulo de X Y Z é um triangulo isósceles \n");
    }
}

```



```

}
else {
    printf("O triangulo de X Y Z é um triangulo escaleno \n");
}
}

```

```

int main(){
float x, y, z;
float equacao;
printf("escreva os valores de X Y Z \n");
scanf("%f%f%f", &x, &y, &z);
teste(x,y,z);
return 0;
}

```

10 - Faça uma função para verificar se um ano é bissexto ou não. Utilize a regra: Um ano bissexto é divisível por 4, mas não por 100, ou então se é divisível por 400. Exemplo: 1988 é bissexto, se é divisível por 4 e não é por 100; 2000 é bissexto porque é divisível por 400. A função deve retornar um valor.

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

```

int bissexto (int ano){
int a;
a=0;
int divisao[3];
divisao[0] = ano % 4;
divisao[1] = ano % 100;
divisao[2] = ano % 400;
if(divisao[0] == 0){
if(divisao[1] == 0){

```

```
a=1;
}
}
else if(divisao[2] == 0){
a=1;
}
else {
a=2;
}
return a;
}
```

```
int main()
{
int ano, resultado;
printf("escreva o ano \n");
scanf("%d", &ano);
resultado = bissexto(ano);

if(resultado == 1){
printf("o ano %d é um ano bissexto \n", ano);
}
else{
printf("o ano %d não é um ano bissexto \n", ano);
}

return 0;
}
```

11 – Crie um programa em C para verificar se um número é primo ou não.

```
#include <stdio.h>

int main(){

    int num;

    int i, a;

    int divisao;

    a=0;

    printf("escreva um numero \n");

    scanf("%d", &num);

    for(i=2; i<num; i++){

        divisao = num % i;

        if(divisao==0){

            a++;

        }

    }

    if(a==0 && a<1){

        printf("o numero é primo \n");

    }

    else{

        printf("o numero não é primo \n");

    }

    return 0;

}
```

12 – Escrever um programa que faça reserva de passagens aéreas de uma companhia. Além da leitura do número dos vôos e a quantidade de lugares disponíveis, ler vários pedidos de reserva, construídos do número da carteira de identidade do cliente e do número do voo desejado. Para cada cliente, verificar se há disponibilidade no voo desejado. Em caso afirmativo, imprimir o número de identidade do cliente, e o número do voo, atualizando o número de lugares disponíveis. Caso contrário, avisar ao cliente da

inexistência de lugares. Indicando o fim dos pedidos de reserva, existe um passageiro cujo o número da carteira de identidade é 9999. Considerar fixo e igual a 37 o número de voos da companhia.

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

```
int main(){
```

```
    int avioes[37][2];
```

```
    int linha, coluna;
```

```
    int reservas, escolha, identidade;
```

```
    printf("escreva o numero de cada avião \n");
```

```
    for(linha=0; linha<37; linha++){
```

```
        scanf("%d", &avioes[linha][1]);
```

```
    }
```

```
    for(linha=0; linha<37; linha++){
```

```
        printf("escreva a quantidade de assentos disponiveis do aviao %d \n", avioes[linha][1]);
```

```
        scanf("%d", &avioes[linha][2]);
```

```
    }
```

```
    printf("Essa é a lista de voos e os assentos disponiveis \n");
```

```
    for(linha=0; linha<37; linha++){
```

```
        printf("%d: ", linha);
```

```
        for(coluna=0; coluna<2; coluna++){
```

```
            printf("%d ", avioes[linha][coluna]);
```

```
        }
```

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

```
    }
```

```
    printf("quantas reservas deseja fazer ? \n");
```

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

```
    printf("qual voo deseja escolher ?\n");
```

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

```
    if(reservas>avioes[escolha][1]){
```

```
    printf("não a assentos disponiveis \n");  
}  
else{  
    printf("qual o número da sua carteira de identidade ? \n");  
    scanf("%d", &identidade);  
    printf("cliente %d o senhor reservou %d lugares no voo %d \n",identidade, reservas,  
avioes[escolha][1]);  
}  
    return 0;  
}
```