PROGRAMAS EXEMPLOS DE VETOR/MATRIZ COM SUBROTINAS

```
//PROGRAMA outroExemploDeUsoDeVetorComSubrotinas
// exercício lista vetor numero 1
#include <iostream>
#include <iomanip>
using namespace std;
#define TMAX 50
// facilita troca tamanho maximo, equivale: const sem gastar memoria

//prototipos das funcoes - VETORES SEMPRE POR REFERENCIA
int leiaNLimSup (int limSup);
void leiaInfos (int n, string nomes[], float notas[]);
float media (int n, float notas[]);
void melhoresAlunos (int n, string nomes[], float notas[]);

int main (){
    string nomes[TMAX];
    float notas[TMAX];
    int n = leiaNLimSup(TMAX); // maximo 50 alunos
    leiaInfos (n, nomes, notas);
    melhoresAlunos (n, nomes, notas);
    return 1;
}
int leiaNLimSup (int limSup){
    int n;
    do{
        cout << "Qtde de elementos (max" << limSup << ") : " ;
        cin>>n;
    }while(n<=0 or n>limSup);
    return n;
}
void leiaInfos (int n, string nomes[], float notas[]){
    for (int i=0; i<n; i++){
        cin.ignore();
        cout << "Aluno "<< i+1 <<"\nNome: ";
        getline(cin,nomes[i]);  // poderia validar string nao vazia
        do{
            cout << "Nota: ";
            cin >> notas[i];
        }while(notas[i]<0 or notas[i]>10);
    }
}
float media (int n, float notas[]){
    float soma=0;
    for (int i=0; i<n; i++)
        soma+= notas[i]; // equivale soma = soma + notas[i];
    return soma / n;
}
void melhoresAlunos (int n, string nomes[], float notas[]){
    int qtde=0;
    float mediaT = media(n, notas); // chamada da função media
    cout << "\nAlunos com nota acima da media " << mediaT;
    cout << " - c/2 casas:"<< setprecision(3) << mediaT << endl;
    for (int i=0; i<n; i++)
        if( notas[i] > mediaT ){
            cout << nomes[i] << endl;
            qtde++;
        }
    cout <<"\nTotal de alunos: " << qtde << endl << endl;
}
```

PROGRAMAS EXEMPLOS DE VETOR/MATRIZ COM SUBROTINAS

```
//PROGRAMA exemploDeUsoDeMatrizComSubrotinas
// exercício lista matriz numero 1
#include <iostream>
using namespace std;
#define MAX 10
// facilita troca tamanho maximo, equivale: const sem gastar memoria

//prototipos das funcoes - MATRIZES E VETORES SEMPRE POR REFERENCIA
int leiaNLimSup (int limSup);
void leiaMatriz(int,int,int matriz[][MAX]);
void mostraMatriz(int,int,int matriz[][MAX]);
int somaColuna(int,int,int matriz[][MAX]);
void geraVetMultLinha(int, int,int matriz[][MAX], int vet[]);
int somaTotMat(int,int,int matriz[][MAX]);
int somaDiag(int,int matriz[][MAX]);
void mostraVetor(int,int vet[]);

int main(){
    int n, A[MAX][MAX], vet[MAX];
    n = leiaNLimSup(MAX);
    leiaMatriz(n,n,A);
    cout<<"\n\nMATRIZ: "<<endl;
    mostraMatriz(n,n,A);
    cout<<"\nSoma da 2a coluna: " << somaColuna(n,1,A); // 2a coluna = indice 1
    geraVetMultLinha(n,n,A,vet);
    cout<<"\nVetor com multiplicacao das linhas: ";
    mostraVetor(n,vet);
    cout<<"\nSoma diagonal: "<< somaDiag(n,A);
    cout<<"\nSoma total: "<< somaTotMat(n,n,A);
    return 1;
}
int leiaNLimSup (int limSup){
    int n;
    do{
        cout << "Qtde de elementos (max" << limSup << ") : " ;
        cin>>n;
    }while(n<=0 or n>limSup);
    return n;
}
void leiaMatriz(int nl,int nc,int matriz[][MAX]){
    for(int i=0; i<nl; i++)
        for(int j=0; j<nc; j++){
            cout << "Elemento "<< i <<","<< j<<": ";
            cin>> matriz[i][j];
        }
}
void mostraMatriz(int nl,int nc,int matriz[][MAX]){
    for(int i=0; i<nl; i++)    {
        for(int j=0; j<nc; j++)
            cout << matriz[i][j]<<"\t";
        cout << endl;
    }
}
int somaColuna(int nl,int colX,int matriz[][MAX]){
    int soma=0;
    for(int i=0; i<nl; i++)
        soma+=matriz[i][colX];
    return soma;
}
```

```
void geraVetMultLinha(int nl, int nc,int matriz[][MAX], int vet[]){
    for(int i=0;i<nl;i++){
        vet[i]=1;
        for(int j=0; j<nc; j++)
            vet[i]*=matriz[i][j];
    }
}
int somaTotMat(int nl,int nc,int matriz[][MAX]){
    int soma=0;
    for(int i=0; i<nl; i++)
        for(int j=0; j<nc; j++)
            soma+=matriz[i][j];
    return soma;
}
int somaDiag(int nl,int matriz[][MAX]){
    int soma=0;
    for(int i=0; i<nl; i++)
        soma+=matriz[i][i];
    return soma;
}
void mostraVetor(int nl,int vet[]){
    for(int i=0; i<nl; i++)
        cout <<"["<<vet[i]<<"]";
    cout << endl;
}
```