Sumário

1 PRG19

1.1 Matrizes

Matrizes são arranjos homogêneos de 2 dimensões. (linhas e colunas). Usa a mesma definição já estudada para vetores (que eram em 1 dimensão) agora extendido a duas. Veja o exemplo

```
int x[10][30]; // x tem 10 linhas e 30 colunas
 x[0][0]=100; // o elemento da prim. linha e prim. coluna vale 100
```

Veja a seguir como uma matriz é definida e impressa

```
#include<iostream>
#include<iomanip>
using namespace std;
int x[3][4]={1,2,3,4,5,6,7};
main(){
   int i,j;
   for (i=0;i<3;i++){
      for(j=0;j<4;j++){
         cout<<setw(4)<<x[i][j];
      }
      cout<<endl;
   } }
```

Note o uso do $<$ *iomanip* $>$ e o consequente *setw* sinalizando o tamanho de cada campo. Sem ele, a matriz fica descaracterizada. Acompanhe

```
    int i,j;
    for (i=0;i<2;i++){
        for(j=0;j<4;j++){
            cout<<v[i][j]<<" ";
        }
        cout<<endl;   }
RESULTADO----------------------
1 200 1360 2
13000 33 2300 101

    for (i=0;i<2;i++){
        for(j=0;j<4;j++){
            cout<<setw(6)<<v[i][j];
        }
        cout<<endl;   }
RESULTADO----------------------
     1   200  1360     2
 13000    33  2300   101
```

1.2 Vendas da banquinha

Exercício suponha a seguinte matriz de vendas de uma banca de jornais. As linhas referem-se a sorvete, guloseimas, lanches e bebidas. As colunas referem-se aos dados de segunda à sexta feira. Então

```
303  787  726  813  530
511  385  826  572  294
168  890  305  623  929
942  592  304  922  923
```

Determine:

- O valor total vendido na semana naquela banca
- Os 5 valores totais de cada dia
- A média diária de sorvete, guloseima, lanche e bebida
- qual foi o melhor dia para o sorvete ?

```
#include<iostream>
#include<iomanip>
using namespace std;
int v[4][5]={303,787,726,813,530, \
             511,385,826,572,294, \
             168,890,305,623,929, \
             942,592,304,922,923};
main(){
    int vs[5]={0};
    float vp[4]={0};
    int ms=-9999999;
    int i,j,dia;
    int tot=0;
    for (i=0;i<4;i++){
        for(j=0;j<5;j++){
            vp[i]=vp[i]+v[i][j];
            vs[j]=vs[j]+v[i][j];
        }
        tot=tot+vp[i];
    }
    cout<<"Total da semana: "<<tot<<endl;
    cout<<"Totais de cada dia: ";
    for (j=0;j<5;j++){cout<<vs[j]<<" ";}
    cout<<endl<<"Medias de produto: ";
    for (j=0;j<4;j++){cout<<vp[j]/5.0<<" ";}
    for (j=0;j<5;j++){
          if (v[0][j]>ms){ms=v[0][j]; dia=j+1;}
    }
    cout<<endl<<"Melhor dia sorvete foi "<<dia;    }
```

1.3 USP7.4 - Existem repetidos na matriz ?

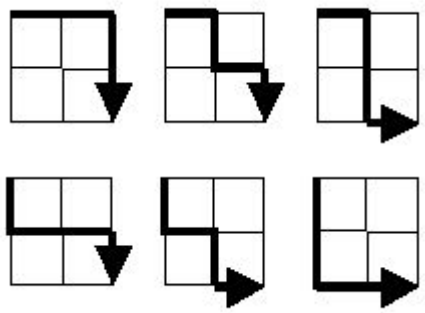
USP7.4 - Dados dois inteiros positivos m e n (ambos menores que 100) e uma matriz real $A_{m \times n}$ verificar se existem elementos repetidos em A.

```
#include<iostream>
using namespace std;
int main(){
int m,n,i,j,repe,ii,jj,ocara;
float M[100][100];
cin>>m>>n;
for (i=0;i<m;i++){
    for (j=0;j<n;j++){
        cin>>M[i][j];
    }
}
repe=0;
for (i=0;i<m;i++){
    for (j=0;j<n;j++){
        ocara=M[i][j];
```

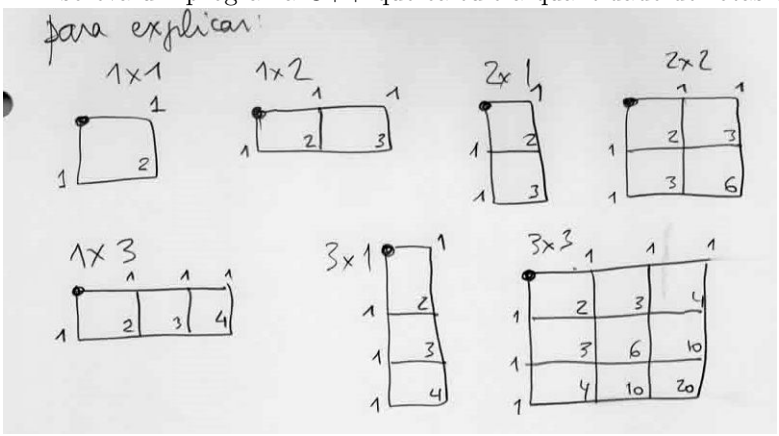
```
        for (ii=0;ii<m;ii++){
            for (jj=0;jj<n;jj++){
                if (((ii!=i)&&(jj!=j)) && (M[i][j]==M[ii][jj])){repe=1;}
            }
        }
    }
}
if (repe==1){cout<<"Existem repetidos"<<endl;}
else {cout<<"Nao existem repetidos"<<endl;}  }
```

1.4 (treliça-euler15)

Começando no canto esquerdo de uma treliça 2×2 e permitindo-se apenas movimentos para a direita e para baixo existem exatamente 6 rotas até o canto inferior da treliça, como se pode ver em



Escreva um programa C++ que calcule a quantidade de rotas em uma treliça 20×20




```
#include<iostream>
using namespace std;
int main(){
   long long int x[21][21];
   int i,j;
   for (i=0;i<21;i++){
      x[0][i]=x[i][0]=1;}
   for (i=1;i<21;i++){
      for (j=1;j<21;j++){
         x[i][j]=x[i][j-1]+x[i-1][j];
      }
   }
   cout<<x[20][20];
}
}
```

```
 a matriz... (só as primeiras 7 linhas e colunas)
     1      1      1      1      1      1      1
     1      2      3      4      5      6      7
     1      3      6     10     15     21     28
     1      4     10     20     35     56     84
     1      5     15     35     70    126    210
     1      6     21     56    126    252    462
```

```
1     7    28    84   210   462   924
```

1.5 USP7.5 - matriz de permutação ?

USP7.5 - Dizemos que uma matriz inteira $A_{n\times n}$ é uma matriz de permutação se em cada linha e em cada coluna houver $n-1$ elementos nulos e um único elemento igual a 1. Por exemplo

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

é de permutação enquanto

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

não é.
Dados um inteiro positivo $n \leq 10$ e uma matriz inteira $A_{n\times n}$ verifique se A é de permutação.

```
#include<iostream>
using namespace std;
int main(){
   int x[10][10];
   int n,i,j,e,sl,sc;
   cin>>n;
   for (i=0;i<n;i++){
     for (j=0;j<n;j++){
        cin>>x[i][j];
     }
   }
   e=1;

   for (i=0;i<n;i++){
     sl=0; sc=0;
     for (j=0;j<n;j++){
        if ((x[i][j]!=0)&&(x[i][j]!=1)){e=0;}
        sl=sl+x[i][j];
        sc=sc+x[j][i];
     }
     if (sl!=1){e=0;}
     if (sc!=1){e=0;}
   }
   if (e==1){cout<<"e permutacao";}
   else {cout<<"nao e permutacao";}  }
```

1.6 (USP1.9: multiplos)

Dados n, i e j imprimir em ordem crescente os n primeiros naturais múltiplos de i, de j ou de ambos.

```
int main(){
   int i,j,n;
   int qtd=0,cand=0;
   cin>>n>>i>>j;
   while (qtd<n){
      if (((cand%i)==0) || ((cand%j)==0)){
           cout<<cand<<" ";
           qtd=qtd+1;
      }
      cand++;
   }
}
```

1.7 (USP1.8: fatorial)

Dado um inteiro não negativo n determinar $n!$.

```
int main(){
  long int fa=1;
  int n;
  cin>>n;
  while (n>1){
    fa=fa*n;
    n--;
  }
  cout<<fa;
}
```

1.8 (maxcols)

Faça um programa C++ que leia do usuário uma matriz $N \times M$ e preecha um vetor de M elementos tal que a posição i do vetor contenha o maior valor da coluna i da matriz. Ao final, imprimir o vetor

```
//---------maxcols-----------------
#include<iostream>
#include<iomanip>
using namespace std;
#define N 3
#define M 5
void leitura(int mat[N][M]){
  int i,j;
  cout<<"Informe "<<N<<" x "<<M<<" elementos "<<endl;
  for (i=0;i<N;i++){
    for (j=0;j<M;j++){
        cin>>mat[i][j];
    }
  }
}
void imprime(int mat[N][M]){
  int i,j;
  cout<<"Matriz original "<<endl;
  for (i=0;i<N;i++){
    for (j=0;j<M;j++){
        cout<<setw(5)<<mat[i][j];
    }
    cout<<endl;
  }
}
int main(){
  int mat[N][M];
  int vet[M];
  int i,j,k,maior;
  leitura(mat);
  for (i=0;i<M;i++){
    maior=-9999;
    for (j=0;j<N;j++){
        if (mat[j][i]>maior){maior=mat[j][i];}
    }
    vet[i]=maior;
  }
  imprime(mat);
  cout<<"---------------------"<<endl;
  for (i=0;i<M;i++){
    cout<<setw(5)<<vet[i];
  }
}
```