

Arrays em C++

Escalar Dado adimensional

```
int A;
```

Vetor Dados homogêneos dispostos em 1 dimensão

```
int A[6];
```

Matriz Dados homogêneos em 2 dimensões

```
int A[7][12];
```

Array Dados homogêneos em n dimensões

```
int A[5][44][12];
```

Em APL $n \leq 256$

Em C++, tamanho prévio

Em C++ o tamanho precisa ser fixo e previamente determinado

```
float X[200]  
...ou  
#define TAMA 200  
float X[TAMA]
```

```
... mas não  
cin>>tama  
float X[tama]
```

A matriz pode ser inicializada

```
int A[10] = {10,20,30,40,50,60,70,80,90,100};
```

...ou

```
int A[] = {10,20,30,40,50,60,70,80,90,100};
```

...ou ainda

```
int A[10];
```

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

```
    A[i]=10+i*10;
```

```
}
```

... mas não

```
int A[];
```

```
A={1,2,3,4,5,6,7,8,9,10};
```

...Também não:

```
int x = 11;
```

```
int A[10]={1,2,3,x,4,5,6,7,8,8};
```

Quando arrays são passados

a funções, eles sempre são passados por REFERÊNCIA. Ou seja

```
int bolha(int v[], int tam){           //e não int & v[], int tam
    int i,j;
    i=tam-1;
    while (i>=1){
        j=1;
        while(j<=i){
            if (v[j-1]>v[j]){
                swap(v[j-1],v[j]);
            }
            j++;
        }
        i--;
    }
}
```

```
int main(){
    int ve[]={7,3,1,9,11,23,6,6,19,10,33,44,55,23,24,25,26};
    int i,tam;
    tam = sizeof(ve)/sizeof(ve[0]);
    bolha(ve,tam);
    cout<<"bolha ";
    for (i=0;i<tam;i++){
        cout<<ve[i]<<' ';
    }
}
```

Em www.pkantek.com.br/atividades/ci208, postei um documento com 7 funções de ordenação (bolha, bolha*, seleção, inserção, shell, heap e quick) programados em c++

Impressão de matriz

```
int x[3][4]={10,20,30,31},{11,12,13,14},{100,200,300,301}};
```

... assim não

```
cout<<x;
```

```
#include<iomanip>
```

```
...
```

```
int i,j;
```

```
for (i=0;i<3;i++){
```

```
    for (j=0;j<4;j++){
```

```
        cout<<setw(4)<<x[i][j];
```

```
    }
```

```
    cout<<endl;
```

```
}
```

Resultado

```
10 20 30 31
11 12 13 14
100 200 300 301

Process returned 0 (0x0)   e
Press any key to continue.
_
```

Exercício: inicializar...

50	55	60	65	70
81	86	91	96	101
112	117	122	127	132

Resposta

```
int A[3][5];  
int i,j,v;  
v=50  
for (i=0;i<3;i++){  
    for(j=0;j<5;j++){  
        A[i][j]=v;  
        v=v+5;  
    }  
    v=v+6;  
}
```

Seja a multiplicação matricial

$$C[i][m] = \sum_{x=1}^j A[i][x] \times B[x][m]$$

```
int A[10][20];
int B[20][50];
int C[10][50];    //... obtém A e B...
for (i=0;i<10;i++){
    for (m=0;m<50;m++){
        som=0;
        for (x=0;x<20;x++){
            som=som+A[i][x]*B[x][m];
        }
        C[i][m]=som;
    }
}    // ...C está calculado
```

Definir e usar

Na definição de uma função com matriz de parâmetro, a primeira dimensão pode ser omitida:

```
int deffun(int A[] [100], ...)
```

Mas na chamada, não

```
...  
x = deffun(c[10] [100] ...)
```

Exercícios

- 1 Escreva uma função que devolva uma matriz identidade de ordem 11
- 2 Escreva uma função que receba uma matriz de ordem 9 e devolva 1 se a matriz é simétrica e 0 senão
- 3 Suponha o sistema:
$$5x + 8y + 3z = 55$$
$$2x + 9y + z = 32$$
$$3x + 2y + 5z = 53$$

Escreva um programa C++ que ache os valores de x , y e z usando determinantes.
- 4 Escreva um programa C++ que imprima o triângulo de pascal de ordem 12 (setw 6 é suficiente).
- 5 Escreva um programa C++ que ache os caminhos possíveis em uma treliça 9×9 . (setw 8 é suficiente).
- 6 Escreva uma função que receba a matriz A (7×7) e calcule a sua transposta A^t .

Ex 1

```
void mi11(int A[11][11]){
    int i,j;
    for (i=0;i<11;i++){
        for (j=0;j<11;j++){
            if (i==j){
                A[i][j]=1;
            }
            else {
                A[i][j]=0;
            }
        }
    }
}

int main(){
    int x[11][11];
    int i,j;
    mi11(x);
    for (i=0;i<11;i++){
        for (j=0;j<11;j++){
            cout<<setw(2)<<x[i][j];
        }
    }
    cout<<endl;
}
```

Ex 2

```
int msim(int a[9][9]){
    int e=1;
    int i,j;
    for (i=0;i<9;i++){
        for (j=0;j<9;j++){
            if (a[i][j]!=a[j][i]) {
                e=0;
            }
        }
    }
    return (e);
}

int main() {
    int x[9][9];
    int mi,mj;
    for (mi=0;mi<9;mi++){
        for (mj=0;mj<9;mj++){
            x[mi][mj]=0;
        }
    }
    x[3][5]=99;
    cout<<msim(x)<<endl;
    x[3][5]=0;
```

Ex 3

```
int det(int x[3][3]){
    int i,j;
    i=(x[0][0]*x[1][1]*x[2][2])+(x[0][1]*x[1][2]*x[2][0])+(x[1][0]*x[2][1]*x[0][2]);
    j=(x[2][0]*x[1][1]*x[0][2])+(x[0][0]*x[1][2]*x[2][1])+(x[1][0]*x[0][1]*x[2][2]);
    return (i-j);
}

int main(){
    int A[3][3]={5,8,3},{2,9,1},{3,2,5}};
    int Dx[3][3]={55,8,3},{32,9,1},{53,2,5}};
    int Dy[3][3]={5,55,3},{2,32,1},{3,53,5}};
    int Dz[3][3]={5,8,55},{2,9,32},{3,2,53}};
    int x,y,z;
    x=det(Dx)/det(A);
    y=det(Dy)/det(A);
    z=det(Dz)/det(A);
    cout<<x<<" "<<y<<" "<<z<<endl;
}
```

Ex4

```
#define tam 13
using namespace std;
int main(){
    int tp[tam][tam]={0}; // ou {};
    int i,j;
    for (i=0;i<tam;i++){
        for (j=0;j<tam;j++){
            if (j==0){
                tp[i][j]=1;
            }
        }
    }
    for (i=1;i<tam;i++){
        for (j=1;j<tam;j++){
            tp[i][j]=tp[i-1][j-1]+tp[i-1][j];
        }
    }
    for (i=0;i<tam;i++){
        for (j=0;j<tam;j++){
            cout<<setw(6)<<tp[i][j];
        }
        cout<<endl;
    }
}
```

Ex 5

```
int main(){                                #define tam 9
    int tp[tam][tam]={0}; // ou {};
    int i,j;
    for (i=0;i<tam;i++){
        for (j=0;j<tam;j++){
            if ((j==0)||(i==0)){
                tp[i][j]=1;
            }
        }
    }
    for (i=1;i<tam;i++){
        for (j=1;j<tam;j++){
            tp[i][j]=tp[i][j-1]+tp[i-1][j];
        }
    }
    for (i=0;i<tam;i++){
        for (j=0;j<tam;j++){
            cout<<setw(8)<<tp[i][j];
        }
        cout<<endl;
    }
}
```

Ex 6

```
int transp (int A[7][7], int T[7][7]){
    int i,j;
    for (i=0;i<7;i++){
        for (j=0;j<7;j++){
            T[i][j]=A[j][i];
        }
    }
}

int main(){
    int x[7][7];
    int y[7][7];
    int i,j;
    for (i=0;i<7;i++){
        for (j=0;j<7;j++){
            x[i][j]=(i*7)+j;
        }
    }
    for (i=0;i<7;i++){
        for (j=0;j<7;j++){
            cout<<setw(8)<<x[i][j];
        }
        cout<<endl;
    }
}
```