

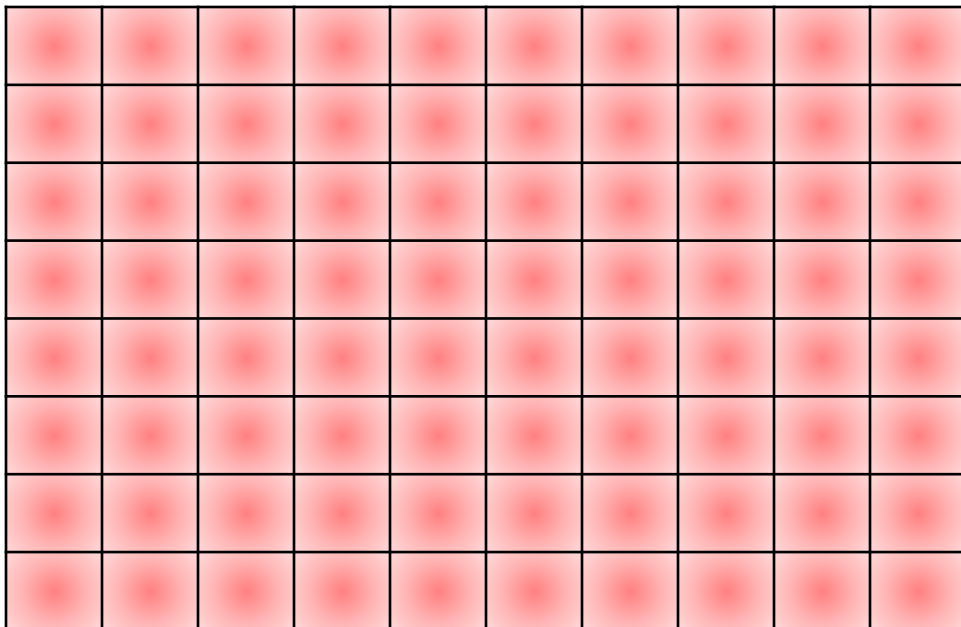
LÝ THUYẾT MA TRẬN

1. ThS. Nguyễn Hữu Lợi
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang

HÌNH ẢNH MA TRẬN

Hình ảnh ma trận

— Hình vẽ

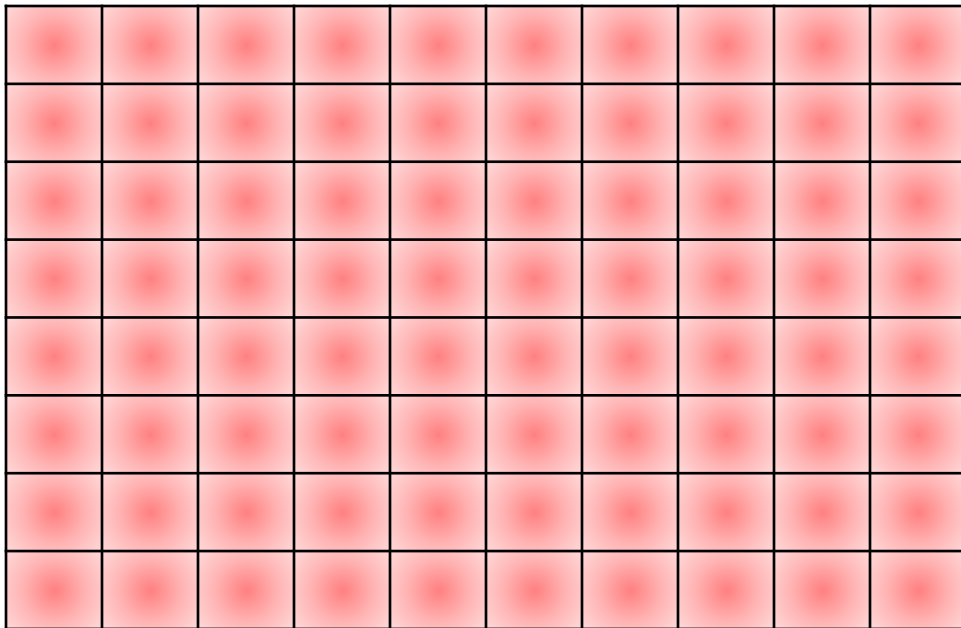


- Ma trận là một bảng gồm nhiều dòng và nhiều cột.
- Ma trận là một kiến trúc gồm nhiều dòng và nhiều cột.
- Ma trận gồm nhiều ô được xếp thành nhiều dòng và nhiều cột.

KHÁI NIỆM MA TRẬN

Khái niệm ma trận

— Hình vẽ



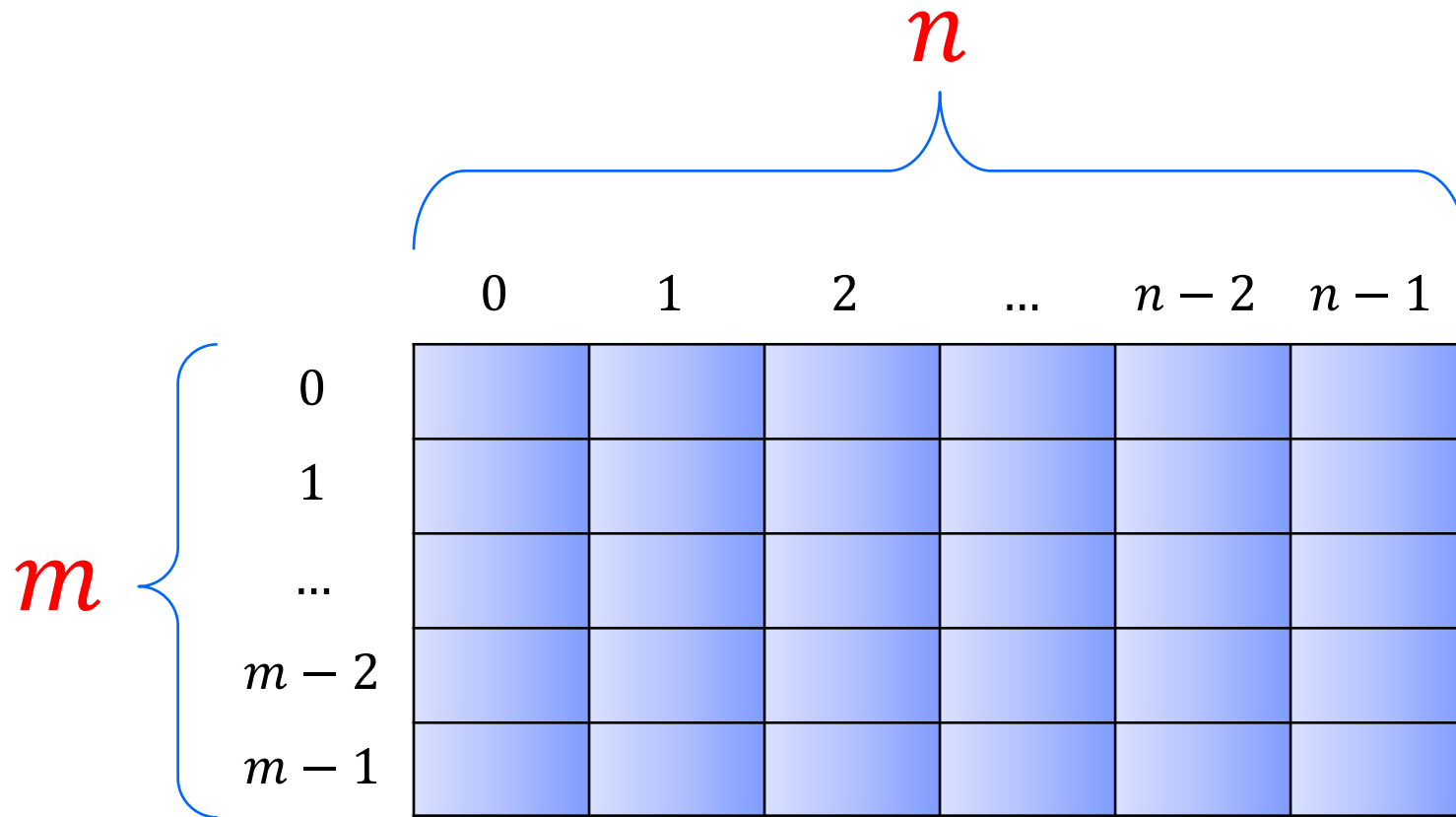
— Khái niệm: Ma trận là một tập hợp các biến có cùng kiểu dữ liệu và cùng tên.

CHỈ SỐ

Chỉ số

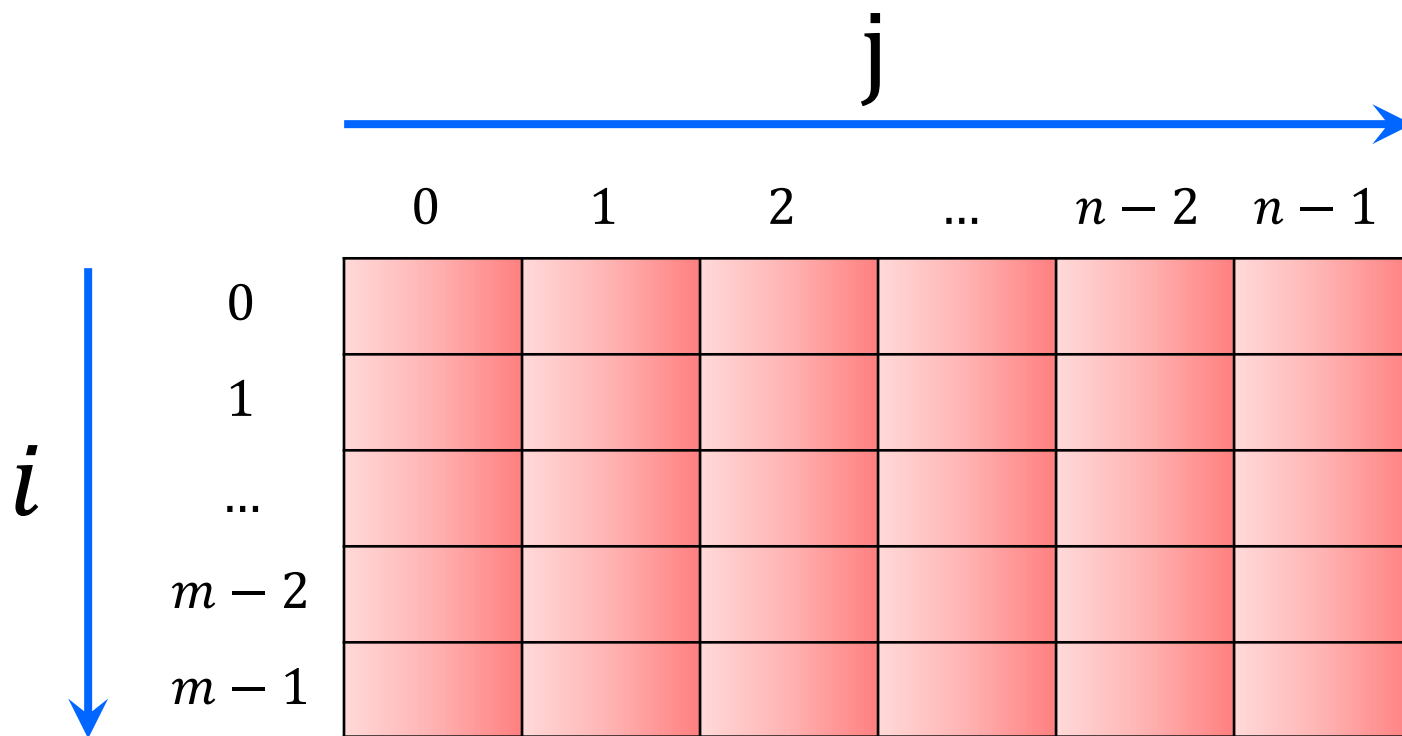
- Qui ước: các dòng và cột trong ma trận của **C/C++** được đánh chỉ số bắt buộc từ **0**.
- Trong ma trận có **m** dòng và **n** cột thì
 - + các dòng trong ma trận đánh số từ **0** cho đến **$m - 1$** ,
 - + các cột trong ma trận đánh số từ **0** cho đến **$n - 1$** .

Chỉ số



		0	1	2	...	$n - 2$	$n - 1$
0							
1							
...							
$m - 2$							
$m - 1$							

Chỉ số



The diagram illustrates a 2D array structure. A horizontal blue arrow labeled j points to the right, indicating the column index. A vertical blue arrow labeled i points downwards, indicating the row index. The array is represented as a grid of cells, each shaded in a light red color. The columns are indexed from 0 to $n-1$, and the rows are indexed from 0 to $m-1$.

	0	1	2	...	$n-2$	$n-1$
0						
1						
...						
$m-2$						
$m-1$						

KHAI BÁO MA TRẬN

Khai báo ma trận

— Cú pháp:

```
KDL <TenBien>[SoDongToiDa][SoCotToiDa];
```

— Ví dụ 1:

```
int a[20][20];
```

Trong ví dụ trên ta nói *a* là ma trận có tối đa 20 dòng và tối đa 20 cột. Mỗi phần tử trong ma trận có kiểu là kiểu số nguyên.

Khai báo ma trận

— Cú pháp

```
KDL <TenBien>[SoDongToiDa][SoCotToiDa];
```

— Ví dụ 2:

```
float b[10][30];
```

Trong ví dụ trên ta nói *b* là ma trận có tối đa 10 dòng và tối đa 30 cột. Mỗi phần tử trong ma trận có kiểu là kiểu số thực.

Khai báo ma trận

— Ví dụ 3:

```
1. struct PhanSo
```

```
2. {
```

```
3. |   int Tu;
```

```
4. |   int Mau;
```

```
5. };
```

```
6. typedef struct PhanSo PHANSO;
```

```
7. PHANSO c[10][15];
```

— Trong ví dụ trên ta nói *c* là ma trận có tối đa 10 dòng và tối đa 15 cột. Mỗi phần tử trong ma trận có kiểu là kiểu cấu trúc **PHANSO**.

KỸ THUẬT NHẬP MA TRẬN

Kỹ thuật nhập ma trận

Vấn đề 1: Định nghĩa hàm nhập
ma trận số nguyên.

Kỹ thuật nhập ma trận

```
void Nhap(int [][][100], int&, int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các số nguyên.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham biến.

Kỹ thuật nhập ma trận

```
void Nhap(int[][100], int&, int&);
```

Khai báo hàm Nhap

```
11. void Nhap(int a[][100], int& m, int& n)
12. {
13.     cout << "Nhap m : ";
14.     cin >> m;
15.     cout << "Nhap n : ";
16.     cin >> n;
17.     for (int i = 0; i < m; i++)
18.         for (int j = 0; j < n; j++)
19.         {
20.             cout << "Nhap a[" << i << "][" << j << " : ";
21.             cin >> a[i][j];
22.         }
23. }
```

Định nghĩa hàm Nhap

Kỹ thuật nhập ma trận

Vấn đề 2: Định nghĩa hàm nhập ma trận số thực.

Kỹ thuật nhập ma trận

```
void Nhap(float[][100], int&, int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các số thực.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham biến.

Kỹ thuật nhập ma trận

```
void Nhap(float[][100], int&, int&);
```

Khai báo hàm Nhap

```
11. void Nhap(float a[][100], int& m, int& n)
```

```
12. {
```

```
13.     cout << "Nhap m : ";
```

```
14.     cin >> m;
```

```
15.     cout << "Nhap n : ";
```

```
16.     cin >> n;
```

```
17.     for (int i = 0; i < m; i++)
```

```
18.         for (int j = 0; j < n; j++)
```

```
19.         {
```

```
20.             cout << "Nhap a[" << i << "][" << j << " : ";
```

```
21.             cin >> a[i][j];
```

```
22.         }
```

```
23. }
```

Định nghĩa hàm Nhap

Kỹ thuật nhập ma trận

Vấn đề 3: Định nghĩa hàm nhập ma trận số nguyên dài.

Kỹ thuật nhập ma trận

```
void Nhap(long [][][100], int&, int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các số nguyên dài.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham biến.

Kỹ thuật nhập ma trận

```
void Nhap(long[][100], int&, int&);
```

Khai báo hàm Nhap

```
11. void Nhap(long a[][100], int& m, int& n)
```

```
12. {
```

```
13.     cout << "Nhap m : ";
```

```
14.     cin >> m;
```

```
15.     cout << "Nhap n : ";
```

```
16.     cin >> n;
```

```
17.     for (int i = 0; i < m; i++)
```

```
18.         for (int j = 0; j < n; j++)
```

```
19.         {
```

```
20.             cout << "Nhap a[" << i << "][" << j << " : ";
```

```
21.             cin >> a[i][j];
```

```
22.         }
```

```
23. }
```

Định nghĩa hàm Nhap

Kỹ thuật nhập ma trận

Vấn đề 4: Định nghĩa hàm nhập
ma trận số thực dài.

Kỹ thuật nhập ma trận

```
void Nhap(double [][][100], int&, int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các số thực dài.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham biến.

Kỹ thuật nhập ma trận

```
void Nhap(double[][100], int&, int&);
```

Khai báo hàm Nhap

```
11. void Nhap(double a[][100],int& m,int& n)
12. {
13.     cout << "Nhap m : ";
14.     cin >> m;
15.     cout << "Nhap n : ";
16.     cin >> n;
17.     for (int i = 0; i < m; i++)
18.         for (int j = 0; j < n; j++)
19.         {
20.             cout << "Nhap a[" << i << "][" << j << " : ";
21.             cin >> a[i][j];
22.         }
23. }
```

Định nghĩa hàm Nhap

Kỹ thuật nhập ma trận

Vấn đề 5: Định nghĩa hàm nhập ma trận các phân số.

Kỹ thuật nhập ma trận

```
void Nhap(PHANSO [][][100], int&, int&);
```

1. Khai báo hàm **Nhap**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Nhap** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các phân số.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham biến.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham biến.

Kỹ thuật nhập ma trận

```
void Nhap(PHANSO[][100], int&, int&);
```

Khai báo hàm Nhap

Kỹ thuật nhập ma trận

```
11.struct PhanSo
12.{
13.    int Tu;
14.    int Mau;
15.};
16.typedef struct PhanSo PHANSO;
```

Khai báo kiểu dữ liệu cấu trúc PHANSO

Kỹ thuật nhập ma trận

```
11. void Nhap(PHANSO& x)
12. {
13.     cout << "Nhap tu : ";
14.     cin >> x.Tu;
15.     cout << "Nhap mau : ";
16.     cin >> x.Mau;
17. }
```

Định nghĩa hàm Nhap PHANSO

```
11. void Nhap(PHANSO a[][100],int& m,int& n)
12. {
13.     cout << "Nhap m : ";
14.     cin >> m;
15.     cout << "Nhap n : ";
16.     cin >> n;
17.     for (int i = 0; i < m; i++)
18.         for (int j = 0; j < n; j++)
19.         {
20.             cout << "Nhap a[" << i << "][" << j << " : ";
21.             Nhap(a[i][j]);
22.         }
23. }
```

Định nghĩa hàm Nhap

KỸ THUẬT XUẤT MA TRẬN

Kỹ thuật xuất ma trận

Vấn đề 1: Định nghĩa hàm xuất ma trận số nguyên.

Kỹ thuật xuất ma trận

```
void Xuat(int [][][100], int, int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các số nguyên.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham trị.

Kỹ thuật xuất ma trận

```
void Xuat(int[][100], int, int);
```

Khai báo hàm Xuat

Kỹ thuật xuất ma trận

```
11. void Xuat(int a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             cout << fixed << setw(10) << a[i][j];
17.         cout << endl;
18.     }
19. }
```

Định nghĩa hàm Xuat

Kỹ thuật xuất ma trận

Vấn đề 2: Định nghĩa hàm xuất ma trận số thực.

Kỹ thuật xuất ma trận

```
void Xuat(float [][][100], int, int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các số thực.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham trị.

Kỹ thuật xuất ma trận

```
void Xuat(float[][100], int, int);
```

Khai báo hàm Xuat

Kỹ thuật xuất ma trận

```
11. void Xuat(float a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             cout << setw(10) << setprecision(3) << a[i][j];
17.         cout << endl;
18.     }
19. }
```

Định nghĩa hàm Xuat

Kỹ thuật xuất ma trận

Vấn đề 3: Định nghĩa hàm xuất ma trận số nguyên dài.

Kỹ thuật xuất ma trận

```
void Xuat(long [][][100], int, int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các số nguyên dài.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham trị.

Kỹ thuật xuất ma trận

```
void Xuat(long [][][100], int, int);
```

Khai báo hàm Xuat

Kỹ thuật xuất ma trận

```
11. void Xuat(long a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             cout << fixed << setw(10) << a[i][j];
17.         cout << endl;
18.     }
19. }
```

Định nghĩa hàm Xuat

Kỹ thuật xuất ma trận

Vấn đề 4: Định nghĩa hàm xuất ma trận số thực dài.

Kỹ thuật xuất ma trận

```
void Xuat(double [][][100], int, int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các số thực dài.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham trị.

Kỹ thuật xuất ma trận

```
void Xuat(double [][][100], int, int);
```

Khai báo hàm Xuat

Kỹ thuật xuất ma trận

```
11. void Xuat(double a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             cout << setw(10) << setprecision(3) << a[i][j];
17.         cout << endl;
18.     }
19. }
```

Định nghĩa hàm Xuat

Kỹ thuật xuất ma trận

Vấn đề 5: Định nghĩa hàm xuất ma trận các phân số.

Kỹ thuật xuất ma trận

```
void Xuat(PHANSO [][][100], int, int);
```

1. Khai báo hàm **Xuat**.
2. Kiểu dữ liệu trả về không có (**void**).
3. Hàm **Xuat** có ba tham số đầu vào.
4. Tham số thứ nhất có kiểu dữ liệu là mảng hai chiều các phân số.
5. Tham số thứ nhất là tham biến.
6. Tham số thứ hai có kiểu dữ liệu là **int**.
7. Tham số thứ hai là tham trị.
8. Tham số thứ ba có kiểu dữ liệu là **int**.
9. Tham số thứ ba là tham trị.

Kỹ thuật xuất ma trận

```
void Xuat(PHANSO [][][100], int, int);
```

Khai báo hàm Xuat

Kỹ thuật xuất ma trận

```
11.struct PhanSo
12.{
13.    int Tu;
14.    int Mau;
15.};
16.typedef struct PhanSo PHANSO;
```

Khai báo kiểu dữ liệu cấu trúc PHANSO

Kỹ thuật xuất ma trận

```
11. void Xuat(PHANSO x)
12. {
13. |   cout << x.Tu << "/" << x.Mau;
14. }
```

Định nghĩa hàm Xuat PHANSO

Kỹ thuật xuất ma trận

```
11. void Xuat(PHANSO a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             Xuat(a[i][j]);
17.         cout << endl;
18.     }
19. }
```

Định nghĩa hàm Xuat



CHƯƠNG TRÌNH THỨ NHẤT

Chương trình thứ nhất

- Viết chương trình thực hiện các yêu cầu sau:
 - + Tạo ngẫu nhiên ma trận các số nguyên với yêu cầu mỗi phần tử trong ma trận là số nguyên nằm trong đoạn $[-100, 100]$.
 - + Xuất ma trận ra màn hình.



Chương trình thứ nhất

```
11.#include <iostream>
12.#include <iomanip>
13.using namespace std;

14.void Nhap(int[][100], int&, int&);
15.void Xuat(int[][100], int, int);
```

Khởi khai báo

Chương trình thứ nhất

```
11.int main()  
12.{  
13.    int b[100][100];  
14.    int k, l;  
15.    Nhap(b, k, l);  
16.    Xuat(b, k, l);  
17.    cout << "Ket thuc!!!\n\n\n";  
18.    return 0;  
19.}
```

Định nghĩa hàm main()

Chương trình thứ nhất

```
11. void Nhap(int a[][100], int& m, int& n)
12. {
13.     cout << "Nhap m : ";
14.     cin >> m;
15.     cout << "Nhap n : ";
16.     cin >> n;
17.     srand(time(NULL));
18.     for (int i = 0; i < m; i++)
19.         for (int j = 0; j < n; j++)
20.             a[i][j] = rand() % (200 + 1) - 100;
21. }
```


Chương trình thứ nhất

```
11. void Xuat(int a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             cout << fixed << setw(10) << a[i][j];
17.         cout << endl;
18.     }
19. }
```

Định nghĩa hàm Xuat



CHƯƠNG TRÌNH THỨ HAI

Chương trình thứ hai

- Viết chương trình thực hiện các yêu cầu sau:
 - + Tạo ngẫu nhiên ma trận các số thực với yêu cầu mỗi phần tử trong ma trận là số thực nằm trong đoạn $[-100, 100]$.
 - + Xuất ma trận ra màn hình.



Chương trình thứ hai

```
11.#include <iostream>
12.#include <iomanip>
13.using namespace std;

14.void Nhap(float[][100], int&, int&);
15.void Xuat(float[][100], int, int);
```

Khởi khai báo

Chương trình thứ hai

```
11.int main()  
12.{  
13.    float b[100][100];  
14.    int k, l;  
15.    Nhap(b, k, l);  
16.    Xuat(b, k, l);  
17.    cout << "Ket thuc!!!\n\n\n";  
18.    return 0;  
19.}
```

Định nghĩa hàm main()

Chương trình thứ hai

```
11. void Nhap(float a[][100], int& m, int& n)
12. {
13.     cout << "Nhap m : ";
14.     cin >> m;
15.     cout << "Nhap n : ";
16.     cin >> n;
17.     for (int i = 0; i < m; i++)
18.         for (int j = 0; j < n; j++)
19.             a[i][j] = -100.0 + rand() / (RAND_MAX / 200);
20. }
```

Định nghĩa hàm Nhap

Chương trình thứ hai

```
11. void Xuat(float a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             cout << fixed << setw(10) <<
17.                 setprecision(3) << a[i][j];
18.         cout << endl;
19.     }
20. }
```

Định nghĩa hàm Xuat



CHƯƠNG TRÌNH THỨ BA

Chương trình thứ ba

- Viết chương trình thực hiện các yêu cầu sau:
 - + Nhập ma trận các số nguyên từ tập tin.
 - + Xuất ma trận ra màn hình.



Chương trình thứ ba

```
11.#include <iostream>
12.#include <iomanip>
13.#include <fstream>
14.using namespace std;
```

```
15.void Nhap(string, int[][100], int&, int&);
16.void Xuat(int[][100], int, int);
```

Khởi khai báo

Chương trình thứ ba

```
11.int main()  
12.{  
13.    int b[100][100];  
14.    int k, l;  
15.    Nhap("intmatrandata01.inp", b, k, l);  
16.    Xuat(b, k, l);  
17.    cout << "Ket thuc!!!\n\n\n";  
18.    return 0;  
19.}
```

Định nghĩa hàm main()

Chương trình thứ ba

```
11. void Nhap(string fname, int a[][100], int& m, int& n)
12. {
13.     ifstream fi(fname);
14.     fi >> m >> n;
15.     for (int i = 0; i < m; i++)
16.         for (int j = 0; j < n; j++)
17.             fi >> a[i][j];
18. }
```

Định nghĩa hàm Nhap

Chương trình thứ ba

```
11. void Xuat(int a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             cout << fixed << setw(10) << a[i][j];
17.         cout << endl;
18.     }
19. }
```

Định nghĩa hàm Xuat



CHƯƠNG TRÌNH THỨ TƯ

Chương trình thứ tư

- Viết chương trình thực hiện các yêu cầu sau:
 - + Nhập ma trận các số thực từ tập tin.
 - + Xuất ma trận ra màn hình.



Chương trình thứ tư

```
11.#include <iostream>
```

```
12.#include <iomanip>
```

```
13.#include <fstream>
```

```
14.using namespace std;
```

```
15.void Nhap(string, float[][100], int&, int&);
```

```
16.void Xuat(float[][100], int, int);
```

Khởi khai báo

Chương trình thứ tư

```
11.int main()  
12.{  
13.    float b[100][100];  
14.    int k, l;  
15.    Nhap("floatmatrandata01.inp", b, k, l);  
16.    Xuat(b, k, l);  
17.    cout << "Ket thuc!!!\n\n\n";  
18.    return 0;  
19.}
```

Định nghĩa hàm main()

Chương trình thứ tư

```
11. void Nhap(string fname, float a[][100], int& m, int& n)
12. {
13.     ifstream fi(fname);
14.     fi >> m >> n;
15.     for (int i = 0; i < m; i++)
16.         for (int j = 0; j < n; j++)
17.             fi >> a[i][j];
18. }
```

Định nghĩa hàm Nhap

Chương trình thứ tư

```
11. void Xuat(float a[][100], int m, int n)
12. {
13.     for (int i = 0; i < m; i++)
14.     {
15.         for (int j = 0; j < n; j++)
16.             cout << fixed << setw(10) <<
17.                 setprecision(3) << a[i][j];
18.         cout << endl;
19.     }
20. }
```

Định nghĩa hàm Xuat

Cảm ơn quý vị đã lắng nghe

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐH CÔNG NGHỆ THÔNG TIN
TOÀN DIỆN – SÁNG TẠO – PHỤNG SỰ