**b. Cho người dùng nhập vào một vị trí cột/hàng cần tính tổng. Hãy tiến hành tính tổng các giá trị lưu trữ trong cột/hàng mà người dùng yêu cầu. Lưu ý: vị trí cột hay hàng phải hợp lệ.**

**c. Cho người dùng nhập vào một giá trị x bất kỳ. Hãy tìm xem x có tồn tại trong mảng hay không? Nếu có thì tồn tại bao nhiêu lần? Tại các vị trí nào?**

const int row = 5, col = 6;

void Input(int a[row][col], int r, int c);

void Output(int a[row][col], int r, int c);

int SumOf(int a[row][col], int r, int c, int n, int type = 0);

void FindAll(int a[row][col], int r, int c, int x);

int main()

{

int a[row][col], r, c, n;

do

{

cout << "Nhap so dong, so cot: ";

cin >> r >> c;

} while (r <= 0 || r > row || c <= 0 || c > col);

Input(a, r, c);

Output(a, r, c);

cout << "Nhap hang can tinh tong: ";

cin >> n;

cout << "Tong cac phan tu hang " << n << " la: " << SumOf(a, r, c, n) << endl;

cout << "Nhap cot can tinh tong: ";

cin >> n;

cout << "Tong cac phan tu cot " << n << " la: " << SumOf(a, r, c, n, 1) << endl;

cout << "Nhap gia tri can tim: ";

cin >> n;

FindAll(a, r, c, n);

system("pause");

return 0;

}

void Input(int a[row][col], int r, int c)

void Output(int a[row][col], int r, int c)

int SumOf(int a[row][col], int r, int c, int n, int type)

{

int sum = 0;

if (type == 0)

{

for (int i = 0; i < c; i++)

sum += a[n - 1][i];

}

else

{

for (int i = 0; i < r; i++)

sum += a[i][n - 1];

}

return sum;

}

void FindAll(int a[row][col], int r, int c, int x)

{

int pos[row\*col][2];

int count = 0;

for (int i = 0; i < r; i++)

for (int j = 0; j < c; j++)

if (a[i][j] == x)

{

pos[count][0] = i;

pos[count][1] = j;

count++;

}

if (count)

{

cout << "Tim thay x " << count << " lan o cac vi tri: ";

for (int i = 0; i < count; i++)

cout << "(" << pos[i][0] << ", " << pos[i][1] << ")\t ";

}

else

cout << "Khong tim thay" << endl;

}

**Viết chương trình cho nhập vào một ma trận vuông cấp n (chứa các số nguyên). Sau đó tính tổng các giá trị trên đường chéo chính và đường chéo phụ của ma trận vừa nhập.**

int Tongchinh(int arr[MAX][MAX], int n) //Cong tat ca phan tu tren duong cheo chinh

{

int sum = 0;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (i == j)

sum = sum + arr[i][j];

}

}

return sum;

}

int Tongphu(int arr[MAX][MAX], int n) //Cong tat ca phan tu tren duong cheo phu

{

int sum = 0;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

{

if (i+ j==n-1)

sum = sum + arr[i][j];

}

}

return sum;

}

**Viết chương trình cho nhập vào hai ma trận (chứa các số nguyên). Sau đó tính kết quả tổng và tích hai ma trận đó.**

#include <iostream>

using namespace std;

const int MAX = 100;

int menu();

void Input(int arr[MAX][MAX], int r, int c);

void Output(int arr[MAX][MAX], int r, int c);

void Tich(int A[MAX][MAX], int B[MAX][MAX], int row\_a, int col\_a, int col\_b);

void Tong(int A[MAX][MAX], int B[MAX][MAX], int r, int c);

int main()

{

int A[MAX][MAX];

int B[MAX][MAX];

int r, c, p;

int select;

do {

system("pause");

system("cls");

select = menu();

switch (select)

{

case 1:

cout << "Nhap hang va cot 2 ma tran: ";

cin >> r >> c;

Input(A, r, c);

cout << "ma tran 1" << endl;

Output(A, r, c);

Input(B, r, c);

cout << "ma tran 2" << endl;

Output(B, r, c);

Tong(A, B, r, c);

break;

case 2:

cout << "Nhap vao hang va cot ma tran 1: ";

cin >> r >> c;

Input(A, r, c);

cout << "Nhap vao cot ma tran 2: ";

cin >> p;

Input(B, c, p);

cout << "Ma tran 1 la: " << endl;

Output(A, r, c);

cout << "Ma tran 2 la: " << endl;

Output(B, c, p);

Tich(A, B, r, c, p);

break;

default:

system("pause");

return 0;

}

cout << "Nhan 1 phim de tiep tuc" << endl;

} while (true);

system("pause");

return 0;

}

int menu()

{

cout << "1. Tong hai ma tran" << endl;

cout << "2. Tich hai ma tran" << endl;

cout << "0. Ket thuc" << endl;

int select;

cout << "Chon: ";

cin >> select;

return select;

}

void Input(int arr[MAX][MAX], int r, int c)

void Output(int arr[MAX][MAX], int r, int c)

void Tich(int A[MAX][MAX], int B[MAX][MAX], int row\_a, int col\_a, int col\_b)

{

int C[MAX][MAX] = { {0} };

for (int i = 0; i<row\_a; i++)

for (int j = 0; j<col\_b; j++)

for (int k = 0; k<col\_a; k++)

C[i][j] = C[i][j] + A[i][k] \* B[k][j];

cout << "Ma tran tich: " << endl;

Output(C, row\_a, col\_b);

}

void Tong(int A[MAX][MAX], int B[MAX][MAX], int r, int c)

{

int C[MAX][MAX];

for (int i = 0; i < r; i++)

for (int j = 0; j < c; j++)

C[i][j] = A[i][j] + B[i][j];

cout << "Ma tran tong: " << endl;

Output(C, r, c);

}

b. Xây dựng hàm trả về trung bình cộng của các phần tử trong mảng số nguyên r hàng và c cột.

double Trungbinhcong(int a[MAX][MAX], int r, int c)

{

int sum = 0;

for (int i = 0; i < r; i++)

{

for (int j = 0; j < c; j++)

{

sum += a[i][j];

}

}

double avg = sum / (r \* c);

return avg;

}

**c. Xây dựng hàm nhận vào mảng số nguyên r hàng và c cột; giá trị số nguyên x. Kiểm tra xem x có tồn tại trong mảng hay không và trả về vị trí của x.**

void findNumber(int a[MAX][MAX], int r, int c, int x)

{

bool found = false;

int row, col;

// Duyệt qua từng phần tử trong mảng và so sánh với giá trị x

for (int i = 0; i < r; i++)

{

for (int j = 0; j < c; j++)

{

if (a[i][j] == x)

{

found = true;

row = i+1;

col = j+1;

break;

}

}

if (found)

{

break;

}

}

// Nếu tìm thấy giá trị x, in ra vị trí của nó trong mảng

if (found) {

cout << "Gia tri " << x << " o dong " << row << " va cot " << col << endl;

}

// Nếu không tìm thấy giá trị x, in ra thông báo không tìm thấy

else {

cout << "Gia tri " << x << " khong co trong mang" << endl;

}

}

**cout << "Nhap vao gia tri can tim: ";**

**cin >> x;**

**findNumber(a, r, c, x);**

**Xây dựng hàm nhận vào mảng số nguyên r hàng và c cột (giả sử các giá trị nhập không trùng nhau) và hai giá trị số nguyên x và y (x và y phải đều tồn tại trong mảng). Tiến hành hoán đổi vị trí của x và y**

void swapNumbers(int a[MAX][MAX], int r, int c, int x, int y)

{

// Tìm vị trí của x và y trong mảng

int x\_row, x\_col, y\_row, y\_col;

for (int i = 0; i < r; i++)

{

for (int j = 0; j < c; j++)

{

if (a[i][j] == x)

{

x\_row = i;

x\_col = j;

}

if (a[i][j] == y)

{

y\_row = i;

y\_col = j;

}

}

}

// Hoán đổi giá trị của x và y

int temp = a[x\_row][x\_col];

a[x\_row][x\_col] = a[y\_row][y\_col];

a[y\_row][y\_col] = temp;

}

**Viết chương trình đặt chỗ ngồi cho 1 máy bay. Máy bay có 13 hàng ghế, mỗi hàng có 6 ghế. Hàng 1 và 2 là hạng thương gia, hàng 3 đến 7 là hạng phổ thông, hàng 8 trở đi là hạng tiết kiệm. Chương trình yêu cầu người dùng nhập vào loại vé (thương gia, phổ thông hay tiết kiệm) và chỗ ngồi muốn đặt. Chương trình kiểm tra và in thông báo thích hợp, đồng thời có menu cho phép người dùng lựa chọn.**

#include <iostream>

#include <iomanip>

using namespace std;

#define ROW 13

#define COL 6

void Xuat(int a[ROW][COL]);

int Kiemtra(int a[ROW][COL], char loai, int r, char c);

int main()

{

int a[ROW][COL] = { 0 };

char loai;

int r;

char c;

do {

system("cls");

cout << "1. Loai ve thuong gia\n";

cout << "2. Loai ve pho thong\n";

cout << "3. Loai ve tiet kiem\n";

cout << "0. Ket thuc\n";

cout << "Chon: ";

cin >> loai;

if (loai == '0') break;

else

{

cout << "Hang ghe(1-13): ";

cin >> r;

cout << "So ghe (A-F): ";

cin >> c;

int kq = Kiemtra(a, loai, r, c);

if (kq == 0)

a[r - 1][c - 'A'] = 1;

else

{

switch (kq)

{

case 1: cout << "hang ghe ko hop le"; break;

case 2: cout << " So ghe ko hop le"; break;

case 3: cout << "Cho nay da co nguoi dat"; break;

case 4: cout << "Loai ghe ko hop le"; break;

}

cout << " chon lai\n";

}

system("pause");

}

} while (loai != '0');

Xuat(a);

system("pause");

return 0;

}

void Xuat(int a[ROW][COL])

{

cout << setw(14) << 'A';

for (int i = 'B'; i <= 'F'; i++)

cout << setw(8) << char(i);

cout << endl << endl;

for (int i = 0; i < ROW; i++)

{

cout << "Hang " << setw(3) << i + 1;

for (int j = 0; j < COL; j++)

if (a[i][j] == 0)

cout << setw(7) << "\*\t";

else

cout << setw(7) << "x\t";

cout << endl;

}

}

int Kiemtra(int a[ROW][COL], char loai, int r, char c)

{

if (r > 13 || r < 1) return 1;

if (c > 'F' || c < 'A') return 2;

if (a[r - 1][c - 'A'] != 0) return 3;

switch (loai)

{

case '1':

if (r > 2) return 1;

break;

case '2':

if (r < 3 || r>7) return 1;

break;

case '3':

if (r < 8) return 1;

break;

default: return 4;

}

return 0;

}

#include <iostream>

#include <string>

using namespace std;

#define SIZE 20

void InMang(int a[SIZE][SIZE], int r, int c);

void ZizacNgang(int a[SIZE][SIZE], int r, int c);

void ZizacDung(int a[SIZE][SIZE], int r, int c);

void XoanOc(int a[SIZE][SIZE], int r, int c);

int main()

{

int a[SIZE][SIZE], r, c;

cout << " Nhap so hang va cot: ";

cin >> r >> c;

ZizacNgang(a, r, c);

cout << "Mang zizacngang: " << endl;

InMang(a, r, c);

ZizacDung(a, r, c);

cout << "Mang zizacdung: " << endl;

InMang(a, r, c);

XoanOc(a, r, c);

cout << "Mang xoanoc: " << endl;

InMang(a, r, c);

return 0;

}

void ZizacNgang(int a[SIZE][SIZE], int r, int c)

{

int k = 1, i = 0, j;

while (k <= r\*c)

{

//qua phai

for (j = 0; j < c; j++)

a[i][j] = k++;

i++;

//qua trai

for (j = c - 1; j >= 0; j--)

a[i][j] = k++;

i++;

}

}

void ZizacDung(int a[SIZE][SIZE], int r, int c)

{

int k = 1, i, j = 0;

while (k <= r\*c)

{

//xuong

for (i= 0; i < r; i++)

a[i][j] = k++;

j++;

//len

for (i = r - 1; i >= 0; i--)

a[i][j] = k++;

j++;

}

}

void XoanOc(int a[SIZE][SIZE], int r, int c)

{

int max = r\*c;

int k = 1, i, j;

int minrow = 0, mincol = 0;

while (k <= max)

{

// qua phai

i = minrow;

for (j = mincol; j < c; j++)

a[i][j] = k++;

minrow++;

//di xuong

j = c - 1;

for (i = minrow; i < r; i++)

a[i][j] = k++;

c--;

// qua trai

i = r - 1;

for (j = c - 1; j >= mincol; j--)

a[i][j] = k++;

r--;

//di len

j =mincol;

for (i = r-1; i>=minrow; i--)

a[i][j] = k++;

mincol++;

}

}

void InMang(int a[SIZE][SIZE], int r, int c)