

```
/*
```

CÂU 01. Xây dựng 1 chương trình gồm các hàm sau:

- Nhập mảng số nguyên gồm m hàng n cột (tối đa 10 hàng, 15 cột).
- Xuất mảng số nguyên gồm m hàng n cột (tối đa 10 hàng, 15 cột).
- Trả về kết quả tích các số lẻ tại 1 hàng nào đó do người dùng yêu cầu trong mảng số nguyên m hàng n cột.

Lưu ý: hàng phải hợp lệ  
(từ 1 đến số hàng tối đa mà ma trận đang có).

Nếu không có số lẻ trong hàng đó thì báo không có  
và không xuất kết quả tích.

- Hàm main để kiểm chứng toàn bộ chương trình.

```
*/
```

```
// Sử dụng mảng truyền thống
```

```
#include <iostream>
using namespace std;
```

```
const int MAX_DONG = 10, MAX_COT = 15;
```

```
void nhapMang(int a[][MAX_COT], int dong, int cot);
```

```
void inMang(int a[][MAX_COT], int dong, int cot);
```

```
int tinhTichCacSoLeTaiMotDong(int a[][MAX_COT], int cot, int dongCanTinhTich);
```

```
int main()
{
```

```
    int a[MAX_DONG][MAX_COT];
    int dongThucTe, cotThucTe;
    int dongCanTinhTich;
```

```
    // Nhập số dòng thực tế cần sử dụng
```

```
    cout << "Nhap so dong thuc te can su dung (1..10): ";
```

```
    cin >> dongThucTe;
```

```
    cout << endl;
```

```
    do {
```

```
        if (dongThucTe < 1 || dongThucTe > 10)
        {
```

```
            cout << "Vui long nhap so dong thuc te can su dung (1..10). Nhap lai: ";
```

```
            cin >> dongThucTe;
```

```
            cout << endl;
```

```
        }
```

```
    } while(dongThucTe < 1 || dongThucTe > 10);
```

```
    // Nhập số cột thực tế cần sử dụng
```

```
    cout << "Nhap so cot thuc te can su dung (1..15): ";
```

```
    cin >> cotThucTe;
```

```
    cout << endl;
```

```
    do {
```

```
        if (cotThucTe < 1 || cotThucTe > 15)
        {
```

```
            cout << "Vui long nhap lai so cot thuc te can su dung (1..10). Nhap lai: ";
```

```
            cin >> cotThucTe;
```

```
            cout << endl;
```

```
        }
```

```
    } while(cotThucTe < 1 || cotThucTe > 15);
```

```

// Gọi hàm nhập mảng
cout << "Nhap mang\n";
nhapMang(a, dongThucTe, cotThucTe);
cout << endl;

// Gọi hàm in mảng
cout << "Mang vua nhap la:\n";
inMang(a, dongThucTe, cotThucTe);
cout << endl;

// Nhập dòng cần tính tích
cout << "Nhap dong can tinh tich cac so le (1.." << dongThucTe << "): ";
cin >> dongCanTinhTich;
cout << endl;
do {
    if (dongCanTinhTich < 1 || dongCanTinhTich > dongThucTe)
    {
        cout << "Vui long nhap lai dong can tinh tich (1.."
            << dongThucTe << "). Nhap lai: ";
        cin >> dongCanTinhTich;
        cout << endl;
    }
} while (dongCanTinhTich < 1 || dongCanTinhTich > dongThucTe);

// Gọi hàm tính tích các giá trị tại một dòng bất kì

// Nếu hàm có trả về giá trị, nghĩa là có số lẻ để tính tích
if (tinhTichCacSoLeTaiMotDong(a, cotThucTe, dongCanTinhTich) != 0)
    cout << "Tich cua cac phan tu la so le tai dong [" << dongCanTinhTich << "] la: "
        << tinhTichCacSoLeTaiMotDong(a, cotThucTe, dongCanTinhTich) << endl;
else // Nếu hàm trả về giá trị = 0
    cout << "Tai dong [" << dongCanTinhTich
        << "] khong co bat ki gia tri nao la so le!" << endl;
cout << endl;

return 0;
}

```

// 01. Hàm nhập mảng

```

void nhapMang(int a[][MAX_COT], int dong, int cot)
{
    for (int i = 0; i < dong; i++)
    {
        cout << "Nhap mot loat " << cot
            << " phan tu cho dong thu [" << i + 1 << "]: ";

        for (int j = 0; j < cot; j++)
            cin >> a[i][j];
    }
}

```

// 02. Hàm in mảng

```

void inMang(int a[][MAX_COT], int dong, int cot)
{
    for (int i = 0; i < dong; i++)
    {
        for (int j = 0; j < cot; j++)
            cout << a[i][j] << " ";

        cout << endl;
    }
}

```

```

// 03. Hàm tính tích (nhân) các giá số lẻ
//      tại một hàng do người dùng yêu cầu
int tinhTichCacSoLeTaiMotDong(int a[][MAX_COT], int cot, int dongCanTinhTich)
{
    int tichCacSoLe = 0;
    dongCanTinhTich--;

    for (int j = 0; j < cot; j++)
        if (a[dongCanTinhTich][j] % 2 != 0)
        {
            if (tichCacSoLe == 0)
                tichCacSoLe = 1;
            tichCacSoLe = tichCacSoLe * a[dongCanTinhTich][j];
        }

    return tichCacSoLe;
}

```

// Sử dụng con trỏ

```

#include <iostream>
using namespace std;

```

```

void nhapMang(int **a, int dong, int cot);
void inMang(int **a, int dong, int cot);
int tinhTichCacSoLeTaiMotDong(int **a, int cot, int dongCanTinhTich);

```

```

int main()
{
    int **a;
    int dongThucTe, cotThucTe;
    int dongCanTinhTich;

    // Nhập số dòng thực tế cần sử dụng
    cout << "Nhap so dong thuc te can su dung (1..10): ";
    cin >> dongThucTe;
    cout << endl;
    do {
        if (dongThucTe < 1 || dongThucTe > 10)
        {
            cout << "Vui long nhap so dong thuc te can su dung (1..10). Nhap lai: ";
            cin >> dongThucTe;
            cout << endl;
        }
    } while(dongThucTe < 1 || dongThucTe > 10);

    // Nhập số cột thực tế cần sử dụng
    cout << "Nhap so cot thuc te can su dung (1..15): ";
    cin >> cotThucTe;
    cout << endl;
    do {
        if (cotThucTe < 1 || cotThucTe > 15)
        {
            cout << "Vui long nhap lai so cot thuc te can su dung (1..10). Nhap lai: ";
            cin >> cotThucTe;
            cout << endl;
        }
    } while(cotThucTe < 1 || cotThucTe > 15);

    // Cấp phát mảng các con trỏ (số dòng)

```

```

a = new int* [dongThucTe];
// Cấp phát vùng nhớ cho mỗi con trỏ (số cột)
for (int i = 0; i < dongThucTe; i++)
    a[i] = new int[cotThucTe];

// Gọi hàm nhập mảng
cout << "Nhap mang\n";
nhapMang(a, dongThucTe, cotThucTe);
cout << endl;

// Gọi hàm in mảng
cout << "Mang vua nhap la:\n";
inMang(a, dongThucTe, cotThucTe);
cout << endl;

// Nhập dòng cần tính tích
cout << "Nhap dong can tinh tich cac so le (1.." << dongThucTe << "): ";
cin >> dongCanTinhTich;
cout << endl;
do {
    if (dongCanTinhTich < 1 || dongCanTinhTich > dongThucTe)
    {
        cout << "Vui long nhap lai dong can tinh tich (1.."
            << dongThucTe << "). Nhap lai: ";
        cin >> dongCanTinhTich;
        cout << endl;
    }
} while (dongCanTinhTich < 1 || dongCanTinhTich > dongThucTe);

// Gọi hàm tính tích các giá trị tại một dòng bất kì

// Nếu hàm có trả về giá trị, nghĩa là có số lẻ để tính tích
if (tinhTichCacSoLeTaiMotDong(a, cotThucTe, dongCanTinhTich) != 0)
    cout << "Tich cua cac phan tu la so le tai dong [" << dongCanTinhTich << "] la: "
        << tinhTichCacSoLeTaiMotDong(a, cotThucTe, dongCanTinhTich) << endl;
else // Nếu hàm trả về giá trị = 0
    cout << "Tai dong [" << dongCanTinhTich
        << "] khong co bat ki gia tri nao la so le!" << endl;
cout << endl;

// Huỷ vùng nhớ của con trỏ
for (int i = 0; i < dongThucTe; i++)
    delete[] a[i];
delete[] a;
a = nullptr;

return 0;
}

```

// 01. Hàm nhập mảng

```

void nhapMang(int **a, int dong, int cot) // a[MAX_DONG][MAX_COT]
{
    for (int i = 0; i < dong; i++)
    {
        cout << "Nhap mot loat " << cot
            << " phan tu cho dong thu [" << i + 1 << "]: ";

        for (int j = 0; j < cot; j++)
            cin >> (*(a + i) + j); // a[i][j]
    }
}

```

```

// 02. Hàm in mảng
void inMang(int **a, int dong, int cot) // a[MAX_DONG][MAX_COT]
{
    for (int i = 0; i < dong; i++)
    {
        for (int j = 0; j < cot; j++)
            cout << (*(a + i) + j) << " "; // a[i][j]

        cout << endl;
    }
}

// 03. Hàm tính tích (nhân) các giá số lẻ
//      tại một hàng do người dùng yêu cầu
int tinhTichCacSoLeTaiMotDong(int **a, int cot, int dongCanTinhTich) // a[MAX_DONG][MAX_COT]
{
    int tichCacSoLe = 0;
    dongCanTinhTich--;

    for (int j = 0; j < cot; j++)
        if (*(a + dongCanTinhTich) + j) % 2 != 0 // a[dongCanTinhTich][j]
        {
            if (tichCacSoLe == 0)
                tichCacSoLe = 1;
            tichCacSoLe = tichCacSoLe * (*(a + dongCanTinhTich) + j); //
a[dongCanTinhTich][j]
        }

    return tichCacSoLe;
}

```

```

/*
    CÂU 02. Xây dựng 1 chương trình gồm các hàm sau:
        - Hàm đệ quy tính  $S = 1/2! + 1/3! + \dots + 1/n!$ 
          (n là số nguyên dương lớn hơn 1)
        - Hàm main để kiểm chứng toàn bộ chương trình.
*/

```

```

#include <iostream>
using namespace std;

int tinhGiaiThua(int n);
double tinhTongS(int n);

int main()
{
    int n;
    cout << "Nhap so nguyen duong (n > 1): ";
    cin >> n;
    cout << endl;

    do {
        if (n <= 1)
        {
            cout << "Nhap so nguyen duong lon hon 1 (n > 1). Nhap lai: ";
            cin >> n;
            cout << endl;
        }
    } while(n <= 1);
}

```

```

    cout << "Tong chuoi la: " << tinhTongS(n) << endl;

    return 0;
}

// 01. Hàm tính giai thừa
int tinhGiaiThua(int n)
{
    if (n == 2)
        return 2;
    return n * tinhGiaiThua(n - 1);
}

// 02. Hàm tính tổng
double tinhTongS(int n)
{
    if (n == 2)
        return 1.0 / tinhGiaiThua(2);
    else
        return tinhTongS(n - 1) + (1.0 / tinhGiaiThua(n));
}

```

```

/*

```

CÂU 3. Dùng con trỏ và ký pháp độ dời để thực hiện:

Viết chương trình dùng cấp phát động, thực hiện các hàm:

- Nhập 1 mảng số nguyên 1 chiều tối đa 20 phần tử.
- Xuất 1 mảng số nguyên 1 chiều tối đa 20 phần tử
- Hàm trả về địa chỉ phần tử là số chính phương cuối trong mảng.  
Nếu không có số chính phương thì trả về NULL.  
(Biết rằng số chính phương là số có căn bậc 2 là một số nguyên.)
- Hàm xóa 1 phần tử tại vị trí x ra khỏi mảng.
- Hàm main để kiểm chứng toàn bộ chương trình.

```

*/

```

```

#include <iostream>
using namespace std;

```

```

void nhap_Mang(int *mang, int phan_Tu)
{
    cout << "Nhap " << phan_Tu << " so nguyen: ";
    for (int i = 0; i < phan_Tu; i++)
        cin >> *(mang + i);
}

```

```

void xuat_Mang(int *mang, int phan_Tu)
{
    for (int i = 0; i < phan_Tu; i++)
        cout << *(mang + i) << " ";
}

```

```

int* kiem_Tra_Phan_Tu_Cuoi_La_So_Chinh_Phuong(int *mang, int phan_Tu)
{
    for (int i = phan_Tu; i >= 0; i--)
    {
        int dem = 0;
        while (dem * dem <= *(mang + i))
        {
            if (dem * dem == *(mang + i))

```

```

        {
            int vi_Tri = i;
            return &mang[vi_Tri];
        }
        dem++;
    }
}
return NULL;
}

void xoa_Phan_Tu(int *mang, int &phan_Tu, int vi_Tri_Xoa)
{
    vi_Tri_Xoa--;
    for (int i = vi_Tri_Xoa; i < phan_Tu; i++)
        *(mang + i) = *(mang + i + 1);
    phan_Tu--;
}

int main()
{
    int *mang;
    int phan_Tu, vi_Tri_Xoa, *dia_Chi_Chinh_Phuong;
    cout << "Nhap so phan tu: ";
    cin >> phan_Tu;
    do {
        if (phan_Tu <= 0 || phan_Tu > 20)
        {
            cout << "Nhap phan tu tu 0 den 20 phan tu. Nhap lai: ";
            cin >> phan_Tu;
        }
    } while (phan_Tu <= 0 || phan_Tu > 20);
    mang = new int[phan_Tu];
    nhap_Mang(mang, phan_Tu);
    cout << "Mang vua nhap: ";
    xuat_Mang(mang, phan_Tu);
    cout << endl;
    cout << "Dia chi phan tu dau tien cua mang: " << int(&mang[0]) << endl;
    dia_Chi_Chinh_Phuong = kiem_Tra_Phan_Tu_Cuoi_La_So_Chinh_Phuong(mang, phan_Tu);
    if (dia_Chi_Chinh_Phuong == NULL)
        cout << "Khong co phan tu chinh phuong trong mang.\n";
    else
        cout << "Dia chi phan tu chinh phuong cuoi cung trong mang: " <<
int(dia_Chi_Chinh_Phuong) << endl;
    cout << "Nhap vi tri phan tu can xoa: ";
    cin >> vi_Tri_Xoa;
    cout << "Mang sau khi xoa la: ";
    xoa_Phan_Tu(mang, phan_Tu, vi_Tri_Xoa);
    xuat_Mang(mang, phan_Tu);
    delete[] mang;
    mang = NULL;
}

```

/\*

CÂU 04. Dùng con trỏ và ký pháp độ dời để thực hiện:  
Viết chương trình dùng cấp phát động, thực hiện các hàm:

- Khởi tạo giá trị cho tam giác Pascal ngược có chiều cao h (h > 0).  
Ví dụ: h = 4

- Hàm xuất tam giác pascal ngược có chiều cao h.
- Hàm khởi tạo giá trị zigzag cho ma trận vuông cấp n chứa các số nguyên.  
Ví dụ: n = 4
- Hàm main để kiểm chứng toàn bộ chương trình.

\*/

```
#include <iostream>
using namespace std;
int tam_Giac_Pas_Cal(int x, int y)
{
    if (y == 0 || x == y)
        return 1;
    else
        return tam_Giac_Pas_Cal(x- 1, y) + tam_Giac_Pas_Cal(x - 1, y - 1);
}

void ve_Tam_Giac_Pas_Cal(int chieu_Cao)
{
    int dem = 0;
    for (int i = 0; i < chieu_Cao; i++)
    {
        int tam = dem;
        for (int j = 0; j < chieu_Cao; j++)
            if (i + j >= chieu_Cao - 1)
            {
                cout << tam_Giac_Pas_Cal(i, tam);
                tam--;
            }
            else
                cout << " ";
        cout << endl;
        dem++;
    }
}

int main()
{
    int chieu_Cao;
    int **mang;
    cout << "Nhap chieu cao cua tam giac: ";
    cin >> chieu_Cao;
    mang = new int*[chieu_Cao];
    for (int i = 0; i < chieu_Cao; i++)
        mang[i] = new int[chieu_Cao];
    cout << "Tam giac pascal co chieu cao " << chieu_Cao << ": \n";
    ve_Tam_Giac_Pas_Cal(chieu_Cao);
    for (int i = 0; i < chieu_Cao; i++)
        delete[]mang[i];
    delete[]mang;
    mang = nullptr;
}
```