

```
/*
```

Bài 01. Xây dựng các hàm đệ qui tính các biểu thức như sau:

a. $S1 = 1 + 2 + 3 + \dots + n$ (n là số nguyên dương)

b. $S2 = 1^2 + 2^2 + 3^2 + \dots + n^2$ (n là số nguyên dương)

c. $S3 = 1 + 1/3 + 1/5 + \dots + 1/n$ (n là số nguyên dương và là số lẻ)

d. $S4 = 1! + 2! + 3! + \dots + n!$ (n là số nguyên dương)

```
*/
```

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

```
int tinhTongSoNguyenDuongS1(int n);
int tinhTongBinhPhuongS2(int n);
double tinhTongSoNguyenDuongVaLaSoLeS3(int n);
int tinhGiaiThua(int n);
int tinhTongGiaiThuaS4(int n);
```

```
int main()
{
```

```
    int n, m;
```

```
    cout << "S1 = 1 + 2 + 3 + ... + n\n";
    cout << "S2 = 1^2 + 2^2 + 3^2 + ... + n^2\n";
    cout << "S3 = 1 + 1/3 + 1/5 + ... + 1/n\n";
    cout << "S4 = 1! + 2! + 3! + ... + n!\n";
    cout << endl;
```

```
    do {
```

```
        cout << "S1, S2, S4 - Nhập vào một số nguyên (n > 0): ";
        cin >> n;
        cout << endl;
```

```
        if (n <= 0)
```

```
            cout << "Nhập sai, vui lòng nhập lại một số nguyên (n > 0)!\n" << endl;
```

```
    } while (n <= 0);
```

```
    do {
```

```
        cout << "S3 - Nhập vào một số nguyên lẻ (n > 0): ";
        cin >> m;
        cout << endl;
```

```
        if (m <= 0 || m % 2 == 0)
```

```
            cout << "Nhập sai, vui lòng nhập lại một số nguyên lẻ (n > 0)!\n" << endl;
```

```
    } while (m <= 0 || m % 2 == 0);
```

```
// a. S1 = 1 + 2 + 3 + ... + n (n là số nguyên dương)
```

```
cout << "Tổng số nguyên dương S1: " << tinhTongSoNguyenDuongS1(n) << endl;
cout << endl;
```

```
// b. S2 = 1^2 + 2^2 + 3^2 + ... + n^2 (n là số nguyên dương)
```

```
cout << "Tổng lũy thừa S2: " << tinhTongBinhPhuongS2(n) << endl;
cout << endl;
```

```
// c. S3 = 1 + 1/3 + 1/5 + ... + 1/n (n là số nguyên dương và là số lẻ)
```

```
cout << "Tổng số nguyên dương và là số lẻ S3: " << tinhTongSoNguyenDuongVaLaSoLeS3(m) << endl;
cout << endl;
```

```
// d. S4 = 1! + 2! + 3! + ... + n! (n là số nguyên dương)
```

```
cout << "Tổng giai thừa S4: " << tinhTongGiaiThuaS4(n) << endl;
cout << endl;
```

```

    return 0;
}

// a.  $S1 = 1 + 2 + 3 + \dots + n$  (n là số nguyên dương)
int tinhTongSoNguyenDuongS1(int n)
{
    // Trường hợp cơ bản - có thể trả về kết quả trực tiếp
    if (n == 1)
        return 1; // return n;

    // Trường hợp tổng quát - có câu lệnh để gọi lại chính hàm đệ quy
    return n + tinhTongSoNguyenDuongS1(n - 1);
}

// b.  $S2 = 1^2 + 2^2 + 3^2 + \dots + n^2$  (n là số nguyên dương)
int tinhTongBinhPhuongS2(int n)
{
    if (n == 1)
        return 1;
    return n * n + tinhTongBinhPhuongS2(n - 1);
}

// c.  $S3 = 1 + 1/3 + 1/5 + \dots + 1/n$  (n là số nguyên dương và là số lẻ)
double tinhTongSoNguyenDuongVaLaSoLeS3(int n)
{
    if (n == 1)
        return 1;
    return 1.0 / n + tinhTongSoNguyenDuongVaLaSoLeS3(n - 2);
}

// d.  $S4 = 1! + 2! + 3! + \dots + n!$  (n là số nguyên dương)
int tinhTongGiaiThuaS4(int n)
{
    if (n == 1)
        return 1;
    return tinhGiaiThua(n) + tinhTongGiaiThuaS4(n - 1);
}

int tinhGiaiThua(int n)
{
    //int tich = 1;

    //if (n == 0 || n == 1)
    //    return 1;

    //for (int i = 2; i <= n; i++)
    //    tich = tich * i;

    //return tich;

    if (n == 1)
        return 1;
    return n * tinhGiaiThua(n - 1);
}

```

/*

Bài 02. Viết hàm đệ quy tính giai thừa của một số nguyên không âm.
Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.

*/

```

#include <iostream>
using namespace std;

int tinhGiaiThua(int n);

int main()
{
    int n;

    cout << "Tính giai thừa của một số nguyên không âm\n";
    cout << endl;

    do {
        cout << "Nhập vào một số nguyên (n > 0): ";
        cin >> n;
        cout << endl;

        if (n <= 0)
            cout << "Nhập sai, vui lòng nhập lại một số nguyên (n > 0)!\n" << endl;
    } while (n <= 0);

    cout << n << "!: " << tinhGiaiThua(n) << endl;
    cout << endl;

    return 0;
}

```

// Hàm đệ quy tính giai thừa của một số nguyên không âm

```

int tinhGiaiThua(int n)
{
    if (n == 1)
        return 1;
    return n * tinhGiaiThua(n - 1);
}

```

/*

Bài 03. Viết hàm đệ quy tính giá trị Fibonacci của một số nguyên không âm.
Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.

*/

```

#include <iostream>
using namespace std;

int tinhGiaTriFibonacci(int n);

int main()
{
    int n;

    cout << "Tính giá trị Fibonacci của một số nguyên không âm\n";
    cout << endl;

    do {
        cout << "Nhập vào một số nguyên (n > 0): ";
        cin >> n;
        cout << endl;

        if (n <= 0)

```

```

        cout << "Nhap sai, vui long nhap lai mot so nguyen (n > 0)!\n" << endl;
    } while (n <= 0);

    cout << "So Fibonacci thu " << n << " la: " << tinhGiaTriFibonacci(n) << endl;
    cout << endl;

    return 0;
}

// Hàm đệ qui tính giá trị Fibonacci của một số nguyên không âm
int tinhGiaTriFibonacci(int n)
{
    if (n == 1 || n == 2)
        return 1;
    return tinhGiaTriFibonacci(n - 1) + tinhGiaTriFibonacci(n - 2);
}

/*
    Bài 04. Viết hàm đệ qui tính giá trị của  $x^n$  ; với x, n là số nguyên.
    Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.
*/

#include <iostream>
using namespace std;

double tinhLuyThua(int x, int n);

int main()
{
    int x, n;

    cout << "Ham tinh luy thua x^n\n";
    cout << endl;

    cout << "Nhap vao co so x: ";
    cin >> x;
    cout << endl;

    cout << "Nhap vao so mu n: ";
    cin >> n;
    cout << endl;

    //do {
    //    cout << "Nhap vao so mu n (n >= 0): ";
    //    cin >> n;
    //    cout << endl;

    //    if (n < 0)
    //        cout << "Nhap sai, vui long nhap lai mot so nguyen (n >= 0)!\n" << endl;
    //} while (n < 0);

    cout << x << "^" << n << " la: " << tinhLuyThua(x, n) << endl;
    cout << endl;

    return 0;
}

// Hàm đệ qui tính giá trị của  $x^n$  ; với x, n là số nguyên.
double tinhLuyThua(int x, int n)

```

```

{
    if (n == 0)
        return 1;

    if (n == -1)
        return 1.0 / x;

    else
        if (n < 0)
            return 1.0 / x * tinhLuyThua(x, n + 1);
        else
            return x * tinhLuyThua(x, n - 1);
}

/*
    Bài 05. Viết hàm đệ qui đếm số chữ số của một số nguyên không âm.
    Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.
*/

#include <iostream>
using namespace std;

int demChuSoNguyenDuong(int n);
int tinhTongCacChuSo(int n);

int main()
{
    int n;

    cout << "Dem so chu so\n";
    cout << endl;

    do {
        cout << "Nhap vao mot so nguyen (n > 0): ";
        cin >> n;
        cout << endl;

        if (n <= 0)
            cout << "Nhap sai, vui long nhap lai mot so nguyen (n > 0)!\n" << endl;
    } while (n <= 0);

    cout << "So nguyen " << n << " co tong so chu so la: "
         << demChuSoNguyenDuong(n) << endl;

    cout << "So nguyen " << n << " co tong cac chu so la: "
         << tinhTongCacChuSo(n) << endl;
    cout << endl;

    return 0;
}

// 01. Hàm đệ qui đếm số chữ số của một số nguyên không âm
int demChuSoNguyenDuong(int n)
{
    // Chỉ có một chữ số
    if (n >= 0 && n <= 9) // n < 10
        return 1;
    else
        return 1 + demChuSoNguyenDuong(n / 10);
}

```

```
}
```

```
// 02. Hàm tính tổng các chữ số
```

```
int tinhTongCacChuSo(int n)
```

```
{
```

```
    if (n >= 0 && n <= 9) // n < 10
```

```
        return n;
```

```
    else
```

```
        return n % 10 + tinhTongCacChuSo(n / 10);
```

```
}
```

```
/*
```

```
    Bài 06. Viết hàm đệ qui tính số đảo ngược của một số nguyên không âm.
```

```
    Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.
```

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
int tinhSoDaoNguoc(int n, int soDaoNguoc);
```

```
int main()
```

```
{
```

```
    int n;
```

```
    int soDaoNguoc = 0;
```

```
    cout << "Tính so dao nguoc\n";
```

```
    cout << endl;
```

```
    do {
```

```
        cout << "Nhap vao mot so nguyen (n > 0): ";
```

```
        cin >> n;
```

```
        cout << endl;
```

```
        if (n <= 0)
```

```
            cout << "Nhap sai, vui long nhap lai mot so nguyen (n > 0)!\n" << endl;
```

```
    } while (n <= 0);
```

```
    cout << "So dao nguoc cua " << n << " la: " << tinhSoDaoNguoc(n, soDaoNguoc) << endl;
```

```
    cout << endl;
```

```
    return 0;
```

```
}
```

```
// Hàm đệ qui tính số đảo ngược của một số nguyên không âm
```

```
int tinhSoDaoNguoc(int n, int soDaoNguoc)
```

```
{
```

```
    if (n == 0)
```

```
        return soDaoNguoc;
```

```
    soDaoNguoc = soDaoNguoc * 10 + n % 10;
```

```
    return tinhSoDaoNguoc(n / 10, soDaoNguoc);
```

```
}
```

```
/*
```

```
    Bài 07. Viết hàm đệ qui tính tổng các chữ số chẵn/lẻ của một số nguyên không âm.
```

```
    Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.
```

```
*/
```

```

#include <iostream>
using namespace std;

int tongChuSoChan(int n);
int tongChuSoLe(int n);

int main()
{
    int n;
    int soDaoNguoc = 0;

    cout << "Tính tổng các chu số chẵn / lẻ\n";
    cout << endl;

    do {
        cout << "Nhập vào một số nguyên (n > 0): ";
        cin >> n;
        cout << endl;

        if (n <= 0)
            cout << "Nhập sai, vui lòng nhập lại một số nguyên (n > 0)!\n" << endl;
    } while (n <= 0);

    cout << "Tổng chu số chẵn của " << n << " là: " << tongChuSoChan(n) << endl;
    cout << "Tổng chu số lẻ của " << n << " là: " << tongChuSoLe(n) << endl;
    cout << endl;

    return 0;
}

```

// 01. Hàm đệ qui tính tổng các chữ số chẵn

```

int tongChuSoChan(int n)
{
    if (n == 0)
        return 0;
    if (n % 10 % 2 == 0)
        return n % 10 + tongChuSoChan(n / 10);
    return tongChuSoChan(n / 10);
}

```

// 02. Hàm đệ qui tính tổng các chữ số lẻ

```

int tongChuSoLe(int n)
{
    if (n == 0)
        return 0;
    if (n % 10 % 2 != 0)
        return n % 10 + tongChuSoLe(n / 10);
    return tongChuSoLe(n / 10);
}

```

/*

Bài 10. Viết hàm đệ qui tìm ước số chung lớn nhất của hai số nguyên.
Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.

* Ước chung lớn nhất của 2 số nguyên a và b
có thể được tính bằng thuật giải Euclid như sau:

Bước 1: nếu b = 0 thì kết quả là a và kết thúc,

ngược lại sang bước 2.

Bước 2: $a \% b \rightarrow d = \text{số dư}$. Sang bước 3.

Bước 3: gán $a = b$, $b = d$ (số dư). Quay lại bước 1.

*/

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

```
int tinhUocChungLonNhat(int a, int b);
```

```
int main()
```

```
{
```

```
    int a, b;
```

```
    cout << "Tim uoc chung lon nhat cua hai so nguyen\n";
```

```
    cout << endl;
```

```
    // Nhập số thứ nhất
```

```
    do {
```

```
        cout << "Nhap vao mot so nguyen thu nhat (a > 0): ";
```

```
        cin >> a;
```

```
        cout << endl;
```

```
        if (a <= 0)
```

```
            cout << "Nhap sai, vui long nhap lai mot so nhi phan (a > 0)!\n" << endl;
```

```
    } while (a <= 0);
```

```
    // Nhập số thứ hai
```

```
    do {
```

```
        cout << "Nhap vao mot so nguyen thu hai (b > 0): ";
```

```
        cin >> b;
```

```
        cout << endl;
```

```
        if (b <= 0)
```

```
            cout << "Nhap sai, vui long nhap lai mot so nhi phan (b > 0)!\n" << endl;
```

```
    } while (b <= 0);
```

```
    cout << "Uoc chung lon nhat giữa hai số " << a << " và " << b
```

```
        << " là: " << tinhUocChungLonNhat(a, b) << endl;
```

```
    cout << endl;
```

```
    return 0;
```

```
}
```

```
// Hàm tìm ước số chung lớn nhất của hai số nguyên
```

```
int tinhUocChungLonNhat(int a, int b)
```

```
{
```

```
    int d;
```

```
    if (b == 0)
```

```
        return a;
```

```
    else
```

```
    {
```

```
        d = a % b;
```

```
        a = b;
```

```
        b = d;
```

```
        return tinhUocChungLonNhat(a, b);
```

```
    }
```

```
}
```



```

/*
    Bài 11. Viết hàm đệ qui tính tổng các giá trị của một mảng 1 chiều lưu trữ số nguyên.
    Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.
*/

#include <iostream>
using namespace std;

const int MAX = 20;

void nhapMang(int a[], int n);
void xuatMang(int a[], int n);
int tinhTongCacGiaTri(int a[], int n);

int main()
{
    int a[MAX];
    int n;

    cout << "Tính tổng các giá trị của một mảng\n";
    cout << endl;

    do {
        cout << "Nhập số lượng phần tử của mảng (n > 0): ";
        cin >> n;
        cout << endl;

        if (n <= 0)
            cout << "Nhập sai, vui lòng nhập lại số lượng phần tử của mảng (n > 0)!\n" <<
endl;
    } while (n <= 0);

    cout << "Nhập một loạt " << n << " phần tử của mảng: ";
    nhapMang(a, n);
    cout << endl;

    cout << "Các phần tử có trong mảng là: ";
    xuatMang(a, n);
    cout << endl;

    cout << "Tổng của các phần tử có trong mảng là: " << tinhTongCacGiaTri(a, n) << endl;
    cout << endl;

    return 0;
}

// 01. Hàm nhập mảng
void nhapMang(int a[], int n)
{
    for (int i = 0; i < n; i++)
        cin >> a[i];
}

// 02. Hàm xuất mảng
void xuatMang(int a[], int n)
{
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
}

```

```

    cout << endl;
}

// 03. Hàm tính tổng các giá trị của một mảng 1 chiều
int tinhTongCacGiaTri(int a[], int n)
{
    if (n == 1)
        return a[0];
    else
        return a[n - 1] + tinhTongCacGiaTri(a, n - 1);
}

/*
    Bài 12. Viết hàm đệ qui tìm giá trị nhỏ nhất của một mảng 1 chiều lưu trữ số nguyên.
    Sau đó xây dựng chương trình để mô phỏng và kiểm tra hàm vừa xây dựng.
*/

#include <iostream>
using namespace std;

const int MAX = 20;

void nhapMang(int a[], int n);
void xuatMang(int a[], int n);
int soSanh(int a, int b);
int timGiaTriNhoNhat(int a[], int n);

int main()
{
    int a[MAX];
    int n;

    cout << "Tinh tong cac gia tri cua mot mang\n";
    cout << endl;

    do {
        cout << "Nhap so luong phan tu cua mang (n > 0): ";
        cin >> n;
        cout << endl;

        if (n <= 0)
            cout << "Nhap sai, vui long nhap lai so luong phan tu cua mang (n > 0)!\n" <<
endl;
    } while (n <= 0);

    cout << "Nhap mot loat " << n << " phan tu cua mang: ";
    nhapMang(a, n);
    cout << endl;

    cout << "Cac phan tu co trong mang la: ";
    xuatMang(a, n);
    cout << endl;

    cout << "Gia tri nho nhat trong mang la: " << timGiaTriNhoNhat(a, n) << endl;
    cout << endl;

    return 0;
}

```

// 01. Hàm nhập mảng

```
void nhapMang(int a[], int n)
{
    for (int i = 0; i < n; i++)
        cin >> a[i];
}
```

// 02. Hàm xuất mảng

```
void xuatMang(int a[], int n)
{
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    cout << endl;
}
```

// 03. Hàm so sánh hai giá trị

```
int soSanh(int a, int b)
{
    if (a > b)
        return b;
    else
        return a;
}
```

// 04. Hàm tìm giá trị nhỏ nhất của một mảng 1 chiều

```
int timGiaTriNhoNhat(int a[], int n)
{
    int min = INT_MAX;

    if (n == 1)
        return a[0];
    else
        if (min > a[n])
            return soSanh(timGiaTriNhoNhat(a, n - 1), a[n - 1]);
}
```

/*

Bài 13. Viết hàm đệ qui để xuất ra 1 tam giác Pascal.

Biết rằng 1 tam giác Pascal có dạng như sau với chiều cao là 5:

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

*/

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

```
const int MAX = 20;
```

```
int inTamGiacPascal(int a, int b);
```

```
int main()
```

```
{
    int h;

    cout << "In tam giac Pascal\n";
```

```

cout << endl;

do {
    cout << "Nhap chieu cao cua tam gia Pascal (h > 0): ";
    cin >> h;
    cout << endl;

    if (h <= 0)
        cout << "Nhap sai, vui long nhap lai chieu cao cua tam gia Pascal (h > 0)!\n" <<
endl;
} while (h <= 0);

for (int i = 0; i < h; i++)
{
    for (int j = 0; j <= i; j++)
        cout << inTamGiacPascal(i, j) << " ";
    cout << endl;
}

return 0;
}

// Hàm xuất ra 1 tam giác Pascal
int inTamGiacPascal(int a, int b)
{
    if (b == 0 || a == b)
        return 1;

    if (b == 1)
        return a;
    else
        return inTamGiacPascal(a - 1, b - 1) + inTamGiacPascal(a - 1, b);
}

// Chuyển đổi hệ số
#include <iostream>

using namespace std;

int chuyenThapSangNhi(int n)
{
    if (n == 0)
        return 0;

    else
        return n % 2 + chuyenThapSangNhi(n / 2)*10;
}

int main()
{
    int n;

    cout << "Nhap so nguyen duong n: ";

    cin >> n;

    cout << n << " co ma nhi phan la: " << chuyenThapSangNhi(n) << endl;
}

```

```
}
```

162.

```
#include <iostream>
```

```
using namespace std;
```

```
int tinhMu(int y)
```

```
{
```

```
    if (y == 0)
```

```
        return 1;
```

```
    else
```

```
        return 2 * tinhMu(y - 1);
```

```
}
```

```
int chuyenNhiSangThap(int n, int soMu)
```

```
{
```

```
    if (n == 0)
```

```
        return 0;
```

```
    else
```

```
        return n % 10 * tinhMu(soMu) + chuyenNhiSangThap(n / 10, soMu + 1);
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cout << "Nhap chuoi nhi phan: ";
```

```
    cin >> n;
```

```
    cout << n << " chuyen sang he thap phan la: " << chuyenNhiSangThap(n, 0) << endl;
```

```
    return 0;
```

```
}
```
