

Bài Tập Đệ Quy C++

Hãy luyện tập các bài tập thường gặp có thể giải bằng đệ quy.

NỘI DUNG :

- [Số Fibonacci, Tổ Hợp](#)
 - [Chuyển Đổi Cơ Số](#)
 - [Các Bài Toán Liên Quan Tới Chữ Số](#)
 - [Các Bài Toán Liên Quan Tới Tổng Dãy Số](#)
 - [Các Bài Toán Liên Quan Tới Mảng](#)
-

1. Số Fibonacci, Tổ Hợp

Số Fibonacci

Số Fibonacci có thể tính bằng hàm đệ quy dựa trên bài toán cơ sở và công thức truy hồi :

Bài toán cơ sở : $F_0 = 0, F_1 = 1$

Công thức truy hồi : $F_n = F_{n-1} + F_{n-2}, n > 1$

Code :

```
#include <iostream>

using namespace std;

int F(int n){
    if(n == 0 || n == 1){
        return n;
    }
    else{
        return F(n - 1) + F(n - 2);
    }
}

int main(){
    cout << F(12) << endl;
    return 0;
}
```

Output :

Tổ hợp chập K của N

Tổ hợp chập K của N ($C(n, k)$) được tính đệ quy dựa vào bài toán cơ sở và công thức truy hồi sau :

Bài toán cơ sở : $C(n, 0) = 1$ và $C(n, n) = 1$

Công thức truy hồi : $C(n, k) = C(n - 1, k - 1) + C(n - 1, k)$

Code :

```
#include <iostream>

using namespace std;

int C(int n, int k){
    if(n == k || k == 0){
        return 1;
    }
    else{
        return C(n - 1, k - 1) + C(n - 1, k);
    }
}

int main(){
    cout << C(12, 2) << endl;
    return 0;
}
```

Output :

2.Chuyển Đổi Cơ Số

Hệ nhị phân

Hệ nhị phân biểu diễn số dưới 2 bit là 0 và 1, khi chuyển từ số thập phân N sang số nhị phân bạn thực hiện quá trình chia N cho 2 cho tới khi $N = 0$, viết ngược lại các số dư của N trong quá trình chia cho 2 đó sẽ được biểu diễn dưới dạng nhị phân.

Ví dụ $N = 37$ thì biểu diễn nhị phân của N sẽ là 100101

N	N / 2	N % 2
37	18	1
18	9	0
9	4	1
4	2	0
2	1	0
1	0	1

Code :

```
#include <iostream>

using namespace std;

void dec_to_bin(long long n){
    if(n < 2){
        printf("%d", n);
    }
    else{
        dec_to_bin(n / 2);
        cout << n % 2;
    }
}

int main(){
    cout << "37 = "; dec_to_bin(37);
    cout << endl;
    cout << "282828282828 = "; dec_to_bin(282828282828);
    return 0;
}
```

Output :

```
37 = 100101
282828282828 = 100000111011001111000010001101111001100
```

Hệ thập lục phân

Hệ thập lục phân hay hệ 16 biểu diễn số thông qua 16 ký tự gồm các chữ số từ 0 tới 9, các số từ 10 tới 15 được thay thế thành các chữ cái từ A tới F.

Tương tự như chuyển từ hệ thập phân sang hệ 16 thì ta thực hiện chia cho 16 và lưu lại số dư trong quá trình chia, viết ngược lại số dư trong quá trình chia ta được biểu diễn dưới hệ số 16

Ví dụ $N = 762$ thì biểu diễn hệ 16 là 2FA

N	N / 16	N % 16
762	47	10 (A)
47	2	15 (F)
2	0	2

Code :

```
#include <iostream>

using namespace std;

void dec_to_hex(long long n){
    if(n < 16){
        if(n < 10){
            cout << n;
        }
        else{
            cout << (char)(n + 55);
        }
    }
    else{
        dec_to_hex(n / 16);
        int r = n % 16;
        if(r < 10){
            cout << r;
        }
        else{
            cout << (char)(r + 55);
        }
    }
}
```

```
int main(){
    cout << "762 = "; dec_to_hex(762);
    cout << endl;
    cout << "282828282828 = "; dec_to_hex(282828282828);
    return 0;
}
```

Output :

```
762 = 2FA
282828282828 = 41D9E11BCC
```

3.Các Bài Toán Liên Quan Tới Chữ Số

Bài 1. Đếm số chữ số của số N

Bài toán cơ sở : $D(N) = 1$ nếu $N < 10$

Công thức truy hồi : $D(N) = 1 + D(N / 10)$ nếu $N \geq 10$

Code :

```
#include <iostream>

using namespace std;

int D(long long n){
    if(n < 10){
        return 1;
    }
    else{
        return 1 + D(n / 10);
    }
}

int main(){
    long long n = 28282828;
    cout << "So chu so cua n : " << D(n) << endl;
    return 0;
}
```

```
}
```

Output :

```
So chu so cua n : 8
```

Bài 2 : Tính tổng chữ số của số N

Bài toán cơ sở : $S(N) = N$ nếu $N < 10$

Công thức truy hồi : $S(N) = N \% 10 + S(N / 10)$ nếu $N \geq 10$

Code :

```
#include <iostream>

using namespace std;

int S(long long n){
    if(n < 10){
        return n;
    }
    else{
        return n % 10 + S(n / 10);
    }
}

int main(){
    long long n = 28282828;
    cout << "Tong chu so cua n : " << S(n) << endl;
    return 0;
}
```

Output :

```
Tong chu so cua n : 40
```

Bài 3. Tính tổng chữ số chẵn (lẻ) của N

Bài toán cơ sở : $S(N) = 0$ nếu N lẻ, N nếu N chẵn với $N < 10$

Công thức truy hồi : $S(N) = S(N / 10)$ nếu N lẻ, $N \% 10 + S(N / 10)$ nếu N chẵn với $N \geq 10$

Code :

```

#include <iostream>

using namespace std;

int S(long long n){
    if(n < 10){
        if(n % 2 == 1) return 0;
        else return n;
    }
    else{
        if(n % 2 == 1) return S(n / 10);
        else return n % 10 + S(n / 10);
    }
}

int main(){
    long long n = 12345678;
    cout << "Tong chu so chan cua n : " << S(n) << endl;
    return 0;
}

```

Output :

Tong chu so chan cua n : 20

Bài 4. Tìm chữ số lớn nhất (nhỏ nhất) của N

Bài toán cơ sở : $F(N) = N$ nếu $N < 10$

Công thức truy hồi : $F(N) = \max(N \% 10, F(N / 10))$ với $N \geq 10$

Code :

```

#include <iostream>

using namespace std;

int F(long long n){
    if(n < 10){
        return n;
    }
}

```

```

else{
    int tmp = F(n / 10);
    return n % 10 > tmp ? n % 10 : tmp;
}
}

int main(){
    long long n = 12349567;
    cout << "Chu so lon nhat cua n : " << F(n) << endl;
    return 0;
}

```

Output :

Chu so lon nhat cua n : 9

4. Các Bài Toán Liên Quan Tới Tổng Dãy Số

Bài 1. Tổng tự nhiên liên tiếp $S(n) = 1 + 2 + 3 + \dots + n$

Bài toán cơ sở : $S(n) = 1$ nếu $n = 1$

Công thức truy hồi : $S(n) = n + S(n - 1)$ với $n > 1$

Code :

```

#include <iostream>

using namespace std;

int S(int n){
    if(n == 1){
        return 1;
    }
    else{
        return n + S(n - 1);
    }
}

```



```
int main(){
    int n = 10;
    cout << "1 + 2 + 3 + ... + 10 = " << S(n) << endl;
    return 0;
}
```

Output :

```
1 + 2 + 3 + ... + 10 = 55
```

Bài 2. Tổng bình phương liên tiếp $S(n) = 1^2 + 2^2 + 3^2 + \dots + n^2$

Bài toán cơ sở : $S(n) = 1$ nếu $n = 1$

Công thức truy hồi : $S(n) = n^2 + S(n - 1)$ với $n > 1$

Code :

```
#include <iostream>

using namespace std;

int S(int n){
    if(n == 1){
        return 1;
    }
    else{
        return n * n + S(n - 1);
    }
}

int main(){
    int n = 10;
    cout << "1^2 + 2^2 + 3^2 + ... + 10^2 = " << S(n) << endl;
    return 0;
}
```

Output :

```
1^2 + 2^2 + 3^2 + ... + 10^2 = 385
```

Bài 3. Tổng bình phương liên tiếp $S(n) = 1/1 + 1/2 + 1/3 + \dots + 1/n$

Bài toán cơ sở : $S(n) = 1$ nếu $n = 1$

Công thức truy hồi : $S(n) = 1/n + S(n - 1)$ với $n > 1$

Code :

```
#include <iostream>
#include <iomanip>

using namespace std;

double S(int n){
    if(n == 1){
        return 1;
    }
    else{
        return (double)1 / n + S(n - 1);
    }
}

int main(){
    int n = 10;
    cout << "1/1 + 1/2 + 1/3 + ... + 1/10 = " << fixed << setprecision(2)
    << S(n) << endl;
    return 0;
}
```

Output :

```
1/1 + 1/2 + 1/3 + ... + 1/10 = 2.93
```

5. Các Bài Toán Liên Quan Tới Mảng

Nếu bạn chưa học lý thuyết về mảng thì có thể học phần mảng trước khi làm các bài tập mục này.

Bài 1. Tính tổng các số chẵn trong mảng

Code :

```
#include <iostream>
```

```

#include <iomanip>

using namespace std;

int even_sum(int a[], int n){
    if(n == 0){
        return 0;
    }
    else{
        if(a[n - 1] % 2 == 0){
            return a[n - 1] + even_sum(a, n - 1);
        }
        else{
            return even_sum(a, n - 1);
        }
    }
}

int main(){
    int n = 6;
    int a[6] = {1, 2, 3, 4, 5, 6};
    cout << "Tong cac so chan trong mang : " << even_sum(a, n) << endl;
    return 0;
}

```

Output :

Tong cac so chan trong mang : 12

Bài 2 . Kiểm tra mảng đối xứng

Code :

```

#include <iostream>
#include <iomanip>

using namespace std;

bool doixung(int a[], int left, int right){
    if(left > right){

```

```

        return true;
    }
    else{
        if(a[left] != a[right]){
            return false;
        }
        else{
            return doixung(a, left + 1, right - 1);
        }
    }
}

int main(){
    int n = 6;
    int a[6] = {1, 2, 3, 3, 2, 1};
    cout << "Mang a doi xung : " << boolalpha << doixung(a, 0, n - 1) << endl;
    return 0;
}

```

Output :

```
Mang a doi xung : true
```

Bài 3. In ra mảng từ trái qua phải

Code :

```

#include <iostream>
#include <iomanip>

using namespace std;

void left_to_right(int a[], int n){
    if(n > 0){
        left_to_right(a, n - 1);
        cout << a[n - 1] << " ";
    }
}

```

```
int main(){
    int n = 6;
    int a[6] = {1, 2, 3, 4, 5, 6};
    left_to_right(a, 6);
    return 0;
}
```

Output :

```
1 2 3 4 5 6
```

Bài 4. In ra mảng từ phải qua trái

Code :

```
#include <iostream>
#include <iomanip>

using namespace std;

void left_to_right(int a[], int n){
    if(n > 0){
        cout << a[n - 1] << " ";
        left_to_right(a, n - 1);
    }
}

int main(){
    int n = 6;
    int a[6] = {1, 2, 3, 4, 5, 6};
    left_to_right(a, 6);
    return 0;
}
```

Output :

```
6 5 4 3 2 1
```