

BÀI TẬP THỰC HÀNH 06

LẬP TRÌNH ĐỆ QUY

A. MỤC TIÊU

Trang bị cho sinh viên kỹ năng lập trình cơ bản trong C++:

- + Hàm đệ quy
- + Viết chương trình sử dụng hàm đệ quy
- + Phát hiện và sửa lỗi

B. KẾT QUẢ SAU KHI HOÀN THÀNH

Sinh viên thành thạo cách khai báo và định nghĩa hàm đệ quy, viết chương trình sử dụng hàm đệ quy áp dụng giải các bài tập từ đơn giản đến phức tạp.

C. NỘI DUNG

- Sử dụng cách xây dựng hàm đệ quy để kiểm tra để giải quyết lớp các bài toán có tính chất đệ quy.
- Viết chương trình theo đúng cấu trúc cơ bản

D. YÊU CẦU PHẦN CỨNG, PHẦN MỀM

Máy tính cài hệ điều hành Windows, RAM tối thiểu 256MB. Phần mềm C FREE 5.0.

E. TÓM TẮT LÝ THUYẾT

1. Khái niệm hàm đệ quy

- Các hàm tự gọi chính mình
- Chỉ giải quyết một trường hợp cơ bản (base case)
- Nếu không phải trường hợp cơ bản
 - Chia bài toán thành các bài toán nhỏ hơn
 - Gọi bản sao mới của hàm để giải quyết vấn đề nhỏ hơn (gọi đệ quy (recursive call) hoặc bước đệ quy (recursive step)
 - + Hội tụ dần dần về trường hợp cơ bản
 - + Hàm gọi chính nó tại lệnh return
 - Cuối cùng, trường hợp cơ bản được giải quyết
 - Câu trả lời đi ngược lên, giải quyết toàn bộ bài toán

2. Cấu trúc chung của hàm đệ qui

Dạng thức chung của một chương trình đệ qui như sau:

```
if (trường hợp suy biến)
{
    trình bày cách giải // giả định đã có cách giải
}
```

```

else    // trường hợp tổng quát
{
    gọi lại hàm với tham số "bé" hơn
}

```

F. HƯỚNG DẪN CHI TIẾT

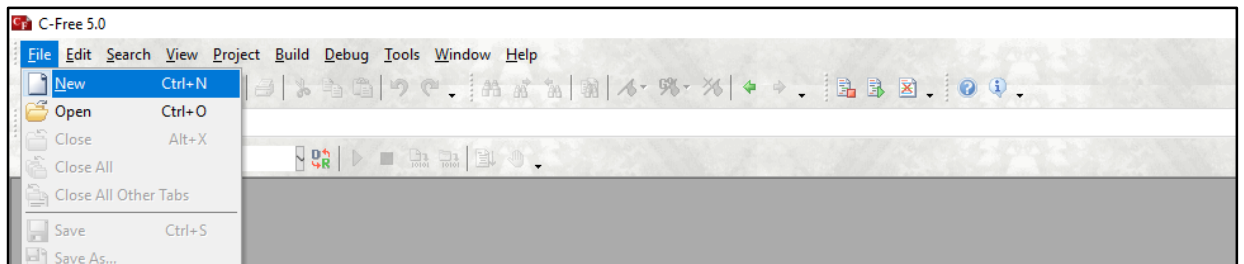
1. Hướng dẫn ban đầu

Bài 1: Viết chương trình sử dụng hàm đệ quy tính số thứ n của dãy Fibonacci

1, 1, 2, 3, 5, 8, 13, 21,

Hướng dẫn:

Bước 1: Tạo mới một file*.cpp thực hiện thao tác File\New



File mới xuất hiện, sinh viên chuyển sang bước 2 thực hiện gõ các câu lệnh theo các bước hướng dẫn.

Bước 2: Khai báo thư viện cần dùng

```
#include<iostream.h>
```

Bước 3: Định nghĩa hàm đệ quy tính số thứ n của dãy Fibonacci

```

long  Fib(int x)
{
    if (x<=2)

        return 1;

    else

        return  Fib(x-1)+Fib(x-2);

}

```

Bước 4: Khai báo hàm main() là hàm chính của chương trình

```
int  main()
```

Bước 5: Khai báo biến số nguyên

```
int  n;
```

Bước 6: Nhập dữ liệu đầu vào

```
cout<<"Nhập n "; cin>>n;
```

Bước 7: Gọi hàm Fib ở trên để tính số thứ n của dãy Fibonacci, in kết quả ra màn hình:

```

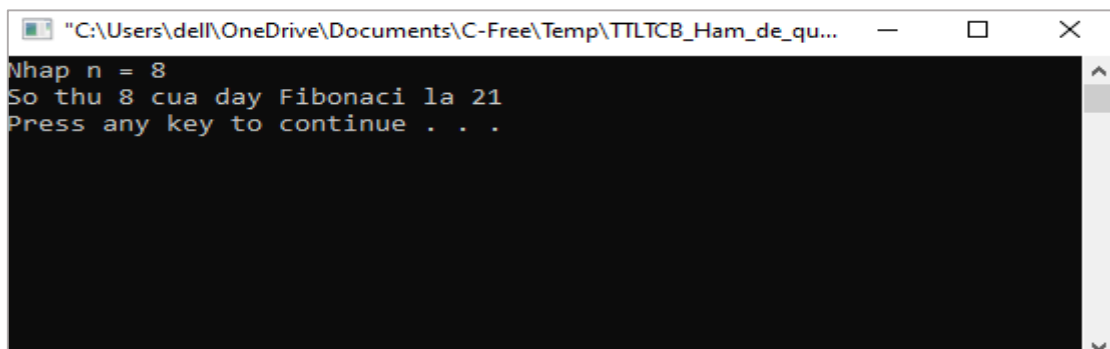
cout << "So thu " <<n<<" của day Fibonacci la "<<Fib(n)<<
endl;

```

Hoàn thiện chương trình như sau:

```
#include <iostream.h>
long    Fib(int x)
{    if (x<=2)
        return 1;
    else
        return Fib(x-1)+Fib(x-2) ;
}
int  main()
{    int n;
    cout<<"n = ";cin>>n;
    cout<<"So thu " <<n<<" cua day Fibonacci la
"<<Fib(n)<< endl;
    return 0;
}
```

Kết quả chạy chương trình:



Bài 2: Viết hàm đệ quy tính

$$S = 1 + 2 + 3 + \dots + (n-1) + n.$$

Hướng dẫn:

Bước 1: Tạo mới một file*.cpp thực hiện thao tác File\New

Tạo file mới: tương tự như hướng dẫn ở bài thực hành 01

Bước 2: Khai báo thư viện cần dùng

```
#include<iostream.h>
```

Bước 3: Định nghĩa hàm đệ quy tính số thứ n của dãy Fibonacci

```
int    sum(int    x)
{    if (x==0)
        return 0;
```

```

        else
            return sum(x-1) + x;
    }

```

Bước 4: Khai báo hàm main() là hàm chính của chương trình

```
int main()
```

Bước 5: Khai báo biến số nguyên

```
int n;
```

Bước 6: Nhập dữ liệu đầu vào

```
cout<<"Nhap n = "; cin>>n;
```

Bước 7: Gọi hàm Fib ở trên để tính số thứ n của dãy Fibonacci, in kết quả ra màn hình:

```
cout << "Tong la S = "<< sum(n)<< endl;
```

Hoàn thiện chương trình như sau:

```

#include <iostream.h>
int sum(int x)
{
    if (x==0)
        return 0;
    else
        return sum(x-1) + x;
}

int main()
{
    int n;
    cout<<"n = ";cin>>n;
    cout << "Tong la S = "<< sum(n)<< endl;
    return 0;
}

```

2. Hướng dẫn thường xuyên

Bài 3: Viết chương trình sử dụng hàm đệ qui để tính tổng $S = 1 + 1/2 + 1/3 + \dots + 1/n$.

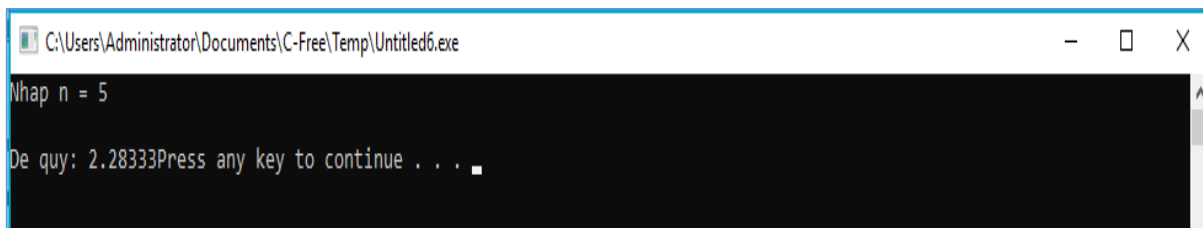
Hoàn thiện chương trình như sau:

```

#include <iostream.h>
float Tong(int x)
{
    if (x==1)
        return 1;
    else
        return Tong(x-1)+ (float)1/x;
}
int main()
{
    int n;
    cout<<"Nhap n = "; cin>>n;
    cout<<"Tong la S = "<<Tong(n)<<endl;
    return 0;
}

```

Kết quả chạy chương trình:



Bài 4:

Viết chương trình sử dụng hàm đệ quy để tính tích sau:

$$S_n = 1^1 \times 2^2 \times 3^3 \times 4^4 \times \dots \times n^n$$

Hướng dẫn:

Bước 1: Tạo file mới: tương tự như hướng dẫn ở bài thực hành 01

Bước 2: Khai báo thư viện cần dùng

```
#include<iostream.h>
```

Bước 3: Định nghĩa hàm đệ quy tính cơ số

```

double luythua(int x)
{
    if (x==1)
        return 1;
    else
        return (float)pow(x,x) * luythua(x-1);
}

```

Bước 4: Khai báo hàm main() là hàm chính của chương trình

```
int main()
```

Bước 5: Khai báo biến

```
int n;
```

Bước 6: Nhập dữ liệu

```
cout<<"Nhap n = ";  
cin>>n;
```

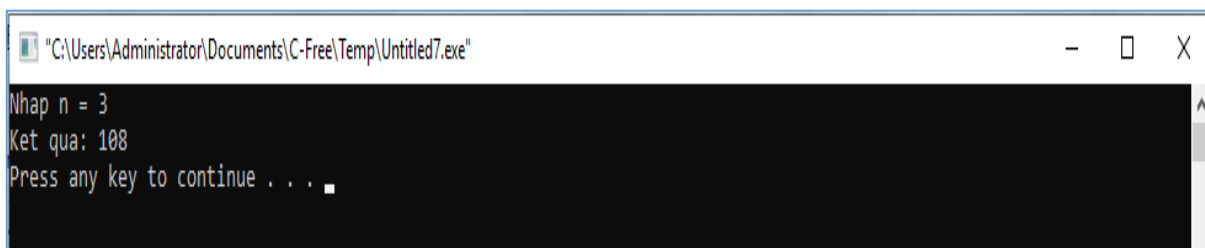
Bước 7: Gọi hàm *luythua* ở trên để tính biểu thức S, in kết quả ra màn hình:

```
cout<<"Ket qua: "<<luythua(n)<<endl;
```

Hoàn thiện chương trình như sau:

```
#include <iostream.h>  
#include <math.h>  
double luythua(int x)  
{  
    if(x==1)  
        return 1;  
    else  
        return (float)pow(x,x) * luythua(x-1);  
}  
  
int main()  
{  
    int n;  
    cout<<"Nhap n = "; cin>>n;  
    cout<<"Ket qua: "<<luythua(n)<<endl;  
    return 0;  
}
```

Kết quả chạy chương trình:



3. Bài tập tự giải

Bài 5: Viết hàm đệ quy tính tổng

$$S = 1 + 4 + 9 + \dots + n^2$$

Bài 6: Viết hàm đệ quy tính tích

$$P = 1 * 3 * 5 * \dots * (2n-1)$$

Bài 7: Viết hàm đệ quy tính giai thừa. Áp dụng tính tổ hợp chập k của n.

$$C_n^k = \frac{n!}{k! * (n-k)!}$$

4. Bài tập về nhà

Bài 8: Viết hàm đệ quy tìm UCLN của 2 số nguyên dương m, n . Áp dụng tìm UCLN của 3 số nguyên dương a, b, c .

Bài 9: Viết hàm đệ quy tính $x(n)$, biết:

$$x(n) = \begin{cases} 1 & \text{với } n=0 \\ n^2x(0)+(n-1)^2x(1)+\dots+2^2x(n-2)+1^2x(n-1) & \text{với } n>0 \end{cases}$$

Bài 10: Có ba cột A, B, C. Cột A hiện đang gán n đĩa có kích thước khác nhau, đĩa nhỏ ở trên đĩa lớn hơn ở dưới. Cần chuyển n đĩa từ cột A sang cột C (dùng cột B làm trung gian) với điều kiện mỗi lần chỉ được chuyển một đĩa, đĩa đặt trên bao giờ cũng nhỏ hơn đĩa đặt dưới.

Bài 11: Sử dụng đệ quy tìm chữ số đầu tiên của số nguyên dương n .

Bài 12: Sử dụng hàm đệ quy tìm chữ số có giá trị lớn nhất của số nguyên dương n .