

User-defined Functions

Simple functions

Ví dụ: Viết hàm để xác định số nhỏ hơn trong 2 số x và y, sau đó sử dụng hàm này để xác định số nhỏ hơn trong 3 số a, b, c.

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

int min_x_y(int x, int y)
{
    if (x < y)
        return x;
    return y;
}

int main()
{
    int a, b, c, min;
    cout << "Nhap a b c: ";
    cin >> a >> b >> c;

    min = min_x_y(a, b);
    min = min_x_y(min, c);

    cout << "Gia tri nho nhat: " << min;

    cout << endl;
    return 0;
}
```

Function prototypes

Ví dụ: Xây dựng các hàm tính giai thừa của 1 số nguyên dương và tổ hợp chập k của n. Nhập các số nguyên dương n và k, tính $C_n^k = \frac{n!}{k!(n-k)!}$ và in ra màn hình.

```
#include <iostream>
using namespace std;
//FUNCTION PROTOTYPES
long factorial(int n);
int Cnk(int k, int n);

int main()
{
    int n, k;
    cout << "n = ";
    cin >> n;
    cout << "k = ";
    cin >> k;
    cout << Cnk(k, n);

    cout << endl;
    return 0;
}

long factorial(int n)
{
    long f = 1;
    for (int i = 2; i <= n; i++)
        f *= i;
    return f;
}

int Cnk(int k, int n)
{
    return factorial(n) / (factorial(k) / factorial(n - k));
}
```

Header files

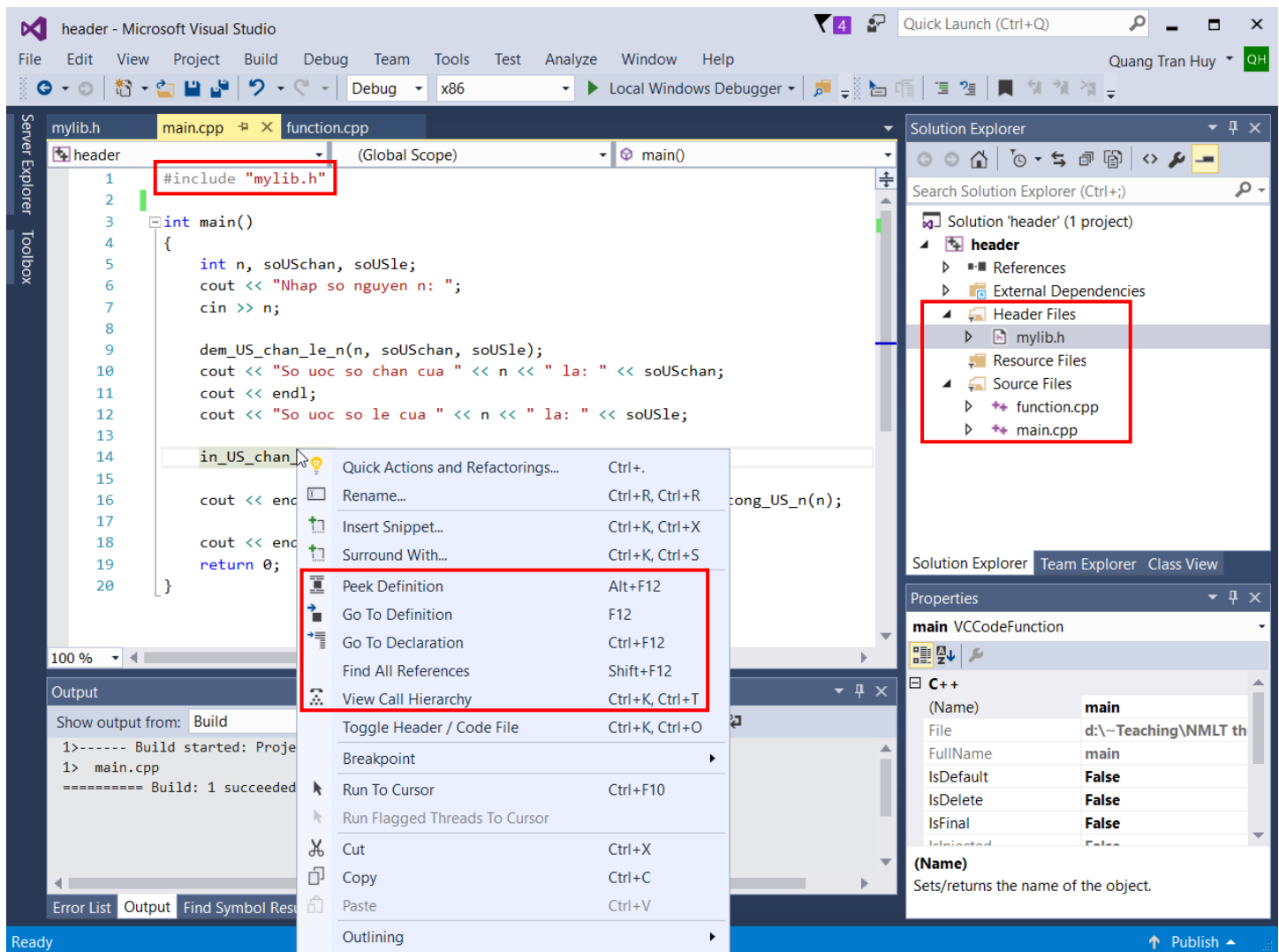
Để có thể sử dụng các hàm thư viện của C/C++ ta phải thực hiện chỉ thị tiền xử lý `#include` khai báo thư viện trong chương trình, ví dụ thực hiện `#include<math.h>` để sử dụng hàm `sqrt`. Các tập tin `.h` được gọi là tập tin header chứa các khai báo của chương trình.

Ta có thể tạo các tập tin header khai báo các hàm tự xây dựng trong những chương trình lớn, tập tin `.h` thường chứa khai báo kiểu dữ liệu, nguyên mẫu hàm, các chỉ thị tiền xử lý...

Ví dụ:

Viết chương trình nhập vào số nguyên `n` và thực hiện:

- Cho biết số ước số chẵn và số ước số lẻ của `n`
- Liệt kê các ước số chẵn và lẻ của `n`
- Tính tổng tất cả các ước số của `n`.



// mylib.h

```
#include<iostream>
```

```
using namespace std;
```

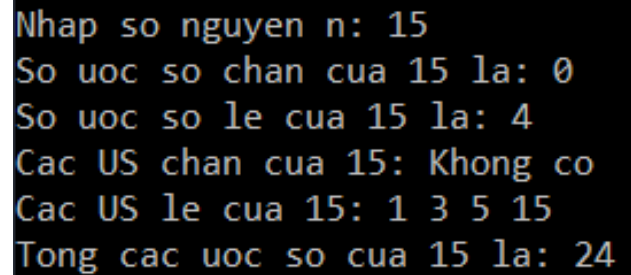
```
void dem_US_chan_le_n(int n, int &soUSChan, int &soUSle);  
void in_US_chan_le_n(int n, int soUSChan, int soUSle);  
int tong_US_n(int n);
```

//main.cpp

```
#include "mylib.h"  
int main()  
{  
    int n, soUSChan, soUSle;  
    cout << "Nhap so nguyen n: ";  
    cin >> n;  
  
    dem_US_chan_le_n(n, soUSChan, soUSle);  
    cout << "So uoc so chan cua " << n << " la: " << soUSChan;  
    cout << endl;  
    cout << "So uoc so le cua " << n << " la: " << soUSle;  
  
    in_US_chan_le_n(n, soUSChan, soUSle);  
  
    cout << endl << "Tong cac uoc so cua " << n << " la: ";  
    cout << tong_US_n(n);  
  
    cout << endl;  
    return 0;  
}
```

// function.cpp

```
#include "mylib.h"  
void dem_US_chan_le_n(int n, int &soUSChan, int &soUSle)  
{  
    int i;  
    soUSChan = 0;  
    for (i = 2; i <= n; i += 2)  
        if (n%i == 0)  
            soUSChan++;  
    soUSle = 0;
```



Nhap so nguyen n: 15
So uoc so chan cua 15 la: 0
So uoc so le cua 15 la: 4
Cac US chan cua 15: Khong co
Cac US le cua 15: 1 3 5 15
Tong cac uoc so cua 15 la: 24

```
        for (i = 1; i <= n; i += 2)
            if (n%i == 0)
                soUSle++;
    }
void in_US_chan_le_n(int n, int soUSchan, int soUSle)
{
    int i;
    cout << endl << "Cac US chan cua " << n << ":";
    if (soUSchan == 0)
        cout << " Khong co";
    else
        for (i = 2; i <= n; i += 2)
            if (n%i == 0)
                cout << " " << i;

    cout << endl << "Cac US le cua " << n << ":";
    if (soUSle == 0)
        cout << "Khong co";
    else
        for (i = 1; i <= n; i += 2)
            if (n%i == 0)
                cout << " " << i;
}

int tong_US_n(int n)
{
    long s = 0;
    for (int i = 1; i <= n; i++)
        if (n%i == 0)
            s += i;
    return s;
}
```

Exercises

Lưu ý: Nếu CT có sử dụng nhiều hơn 2 hàm (không tính main) thì xây dựng Header và tách mã nguồn vào các tập tin khác nhau.

1. VCT nhập vào số tự nhiên N và thực hiện:

- Tìm số M bằng cách loại bỏ đi trong N tất cả các chữ số 0 và 5 và giữ nguyên thứ tự của các chữ số còn lại. Ví dụ N = 591090 thì M = 919.
- Phân tích N thành tích các thừa số nguyên tố.

2. Một số Armstrong có đặc điểm: số đó có n ký tự, tổng các lũy thừa bậc n của các ký số bằng chính nó. Ví dụ số 153, có 3 chữ số và $1^3 + 5^3 + 3^3 = 153$.

- Viết hàm kiểm tra số nguyên dương x có thỏa điều kiện không.
- Nếu x thỏa, hãy liệt kê các số Armstrong nhỏ hơn x^n .

3. VCT nhập vào số nguyên dương N và thực hiện:

- Kiểm tra N có phải là số Palindrom không, nếu các chữ số của nó viết theo thứ tự ngược lại thì bằng chính nó, ví dụ 121 hay 1221.
- Tìm tất cả các số Palindrom >10 và <100 mà bình phương của chúng cũng là số Palindrom.
- Nếu N không phải là số Palindrom thì thay đổi thứ tự của các chữ số của nó theo thứ tự ngược lại và cộng với số ban đầu. Nếu tổng chưa phải là một số Palindrom thì lặp lại quá trình trên đối với tổng đó cho đến khi nhận được một số Palindrom.

4. VCT nhập x, n và tính:

$$\bullet \cos(x) = 1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 - \frac{1}{720}x^6 + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} \text{ với } -\infty < x < \infty$$

$$\bullet \operatorname{arccotan}(x) = \frac{\pi}{2} - x + \frac{1}{3}x^3 - \frac{1}{5}x^5 + \frac{1}{7}x^7 - \frac{1}{9}x^9 + \dots = \frac{\pi}{2} - \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1}$$

$$\bullet \arctan(x) = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \dots = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n-1} x^{2n-1} \text{ với } -1 < x < 1$$

$$\bullet e^x = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \dots = \sum_{n=0}^{\infty} \frac{1}{n!} x^n \text{ với } -\infty < x < \infty$$

$$\bullet \ln(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \dots = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} x^n \text{ với } -\infty < x < \infty$$

$$\bullet \ln\left(\frac{1+x}{1-x}\right) = 2x + \frac{2}{3}x^3 + \frac{2}{5}x^5 + \frac{2}{7}x^7 + \dots = \sum_{n=1}^{\infty} \frac{2}{2n-1} x^{2n-1} \text{ với } -\infty < x < \infty$$

$$\bullet \sin(x) = x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \text{ với } -\infty < x < \infty$$