

## BÀI 4: HÀM

- Cấu trúc của một hàm
- Định nghĩa hàm trong chương trình
- Các bước xây dựng một hàm
- Sử dụng hàm trong chương trình
- Truyền tham số: truyền tham trị, truyền tham chiếu.

### 1. Bài tập hướng dẫn

#### 1.1. Bài toán 1

Viết chương trình nhập vào 3 số a, b, c là độ dài 3 cạnh của một tam giác, tính và in ra màn hình chu vi và diện tích tam giác nếu a, b, c thỏa mãn là 3 cạnh của tam giác.

#### Bố cục chương trình:

- Hàm kiểm tra 3 số thỏa mãn là 3 cạnh tam giác.
- Hàm tính chu vi tam giác.
- Hàm tính diện tích tam giác.
- Hàm main()

#### Chương trình:

```
#include<conio.h>
#include<math.h>
#include<iostream.h>
bool latamgiac(float a, float b, float c);
float chuvi(float a, float b, float c);
float dientich(float a, float b, float c);
void main()
{
    float a,b,c;
    cout<<"Nhập do dai 3 canh tam giac: ";
    cin>>a>>b>>c;
    if (latamgiac(a,b,c))
    {
        float cv=chuvi(a,b,c);
        float s=dientich(a,b,c);
        cout<<"Chu vi tam giac la "<<cv<<endl;
        cout<<"Dien tich tam giac la "<<s;
    }
    else
        cout<<"3 so khong thoa man 3 canh tam giac";
    getch();
}
bool latamgiac(float a, float b, float c)
{
    if (a>0&&b>0&&c>0&&a+b>c&&b+c>a&&c+a>b)
        return true;
    else
        return false;
}
float chuvi(float a, float b, float c)
{
    return a+b+c;
}
float dientich(float a, float b, float c)
{
    float p=chuvi(a,b,c)/2;
    return sqrt(p*(p-a)*(p-b)*(p-c));
}
```

```
}
```

## 1.2. Bài toán 2

Viết chương trình nhập vào số nguyên dương  $k < 100$ , in ra màn hình  $k$  số nguyên tố đầu tiên, và tổng của chúng.

### Bố cục chương trình

- Hàm nhập số nguyên dương.
- Hàm kiểm tra một số nguyên dương là số nguyên tố hay không.
- Hàm in ra màn hình dãy  $k$  số nguyên tố và trả về tổng của  $k$  số nguyên tố.
- Hàm main().

### Chương trình

```
#include<conio.h>
#include<math.h>
#include<iostream.h>
int nhap();
bool languyento(int n);
long daynguyento(int k);
void main()
{
    int k=nhap();
    long t=daynguyento(k);
    cout<<"Tong "<<k<<" so nguyen to la "<<t;
    getch();
}
int nhap()
{
    int a;
    do{
        cout<<"Nhap so nguyen duong k: ";
        cin>>a;
    }while (a<1);
    return a;
}
bool languyento(int n)
{
    int i=2;
    bool kt=true;
    int m=(int)sqrt(n);
    while (i<=m && kt==true)
    {
        if (n%i==0)
            kt=false;
        i++;
    }
    return kt;
}
long daynguyento(int k)
{
    int i=2, dem=0;
    long t=0;
    cout<<"Day "<<k<<" so nguyent to: ";
    while (dem<k)
    {
        if (languyento(i))
```

```

        {
            cout<<i<<" ";
            dem++;
            t = t+i;
        }
        i++;
    }
    cout<<endl;
    return t;
}

```

### 1.3. Bài toán 3

Viết chương trình nhập vào số nguyên dương  $n$ , tính và in ra màn hình tổng của các số chẵn và tổng của các số lẻ trong đoạn  $[1, n]$ .

#### Bố cục chương trình

- Hàm nhập số nguyên dương.
- Hàm tính tổng các số chẵn và tổng các số lẻ.
- Hàm main().

**Lưu ý:** Cách sử dụng hàm truyền tham chiếu.

```

#include<conio.h>
#include<math.h>
#include<iostream.h>
int nhap();
void tong(int n, int &t1, int &t2);
void main()
{
    int n=nhap();
    int tc, tl;
    tong(n,tc,tl);
    cout<<"Tong cac so chan la "<<tc<<endl;
    cout<<"Tong cac so le la "<<tl;
    getch();
}
int nhap()
{
    int a;
    do{
        cout<<"Nhap so nguyen duong k: ";
        cin>>a;
    }while (a<1);
    return a;
}
void tong(int n, int &t1, int &t2)
{
    t1=t2=0;
    for (int i=1; i<=n; i++)
        if (i%2==0)
            t1 += i;
        else
            t2 += i;
}

```

## 2. Bài tập tự làm

1. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:

- Hàm nhập vào một số nguyên dương, nếu không thỏa mãn yêu cầu nhập lại.

- Hàm tính  $k!$  với  $k$  nguyên dương bất kỳ.
  - Hàm tính tổ hợp chập  $k$  của  $n$  theo công thức:  $C_n^k = \frac{n!}{k!(n-k)!}$  (sử dụng hàm tính  $k!$ )
  - Hàm `main()` sử dụng các hàm trên để nhập vào 2 số nguyên dương  $n$  và  $k$ , tính và in ra màn hình tổ hợp chập  $k$  của  $n$ .
2. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:
- Hàm nhập vào số thực  $x$  và số thực  $\varepsilon \leq 0.001$ , nếu  $\varepsilon$  không thỏa mãn yêu cầu nhập lại.
  - Hàm tính  $k!$  với  $k$  nguyên dương bất kỳ.
  - Hàm tính  $x^n$  với  $x$  thực và  $n$  nguyên dương bất kỳ.
  - Hàm tính gần đúng  $e^x$  với sai số  $\varepsilon$  bất kỳ theo công thức  $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^i}{i!} + \dots$
  - Hàm `main()` sử dụng các hàm trên để nhập vào số thực  $x$  và số thực  $\varepsilon$ , tính và in ra màn hình  $e^x$  với sai số  $\varepsilon$ .
  - Chương trình cho phép tính toán với nhiều giá trị  $x$  và  $\varepsilon$  khác nhau theo yêu cầu của người dùng.
3. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:
- Hàm nhập vào 2 số nguyên dương  $a$  và  $b$ , nếu  $a, b$  không thỏa mãn yêu cầu nhập lại.
  - Hàm tìm ước số chung lớn nhất của nguyên dương bất kỳ theo thuật toán sau:  
 Bước 1: Nếu  $a < b$  thì đổi chỗ  $a$  và  $b$ .  
 Bước 2: Tính  $r = a \% b$ .  
 Bước 3: Kiểm tra  $r != 0$ .  
     Nếu là true sang bước 4.  
     Nếu là false sang bước 5.  
 Bước 4:  
      $a = b$ ;  
      $b = r$ ;  
      $r = a \% b$ ;  
     Quay lại bước 3.  
 Bước 5: Trả về  $b$  là ước chung lớn nhất, kết thúc.
- Hàm `main()` sử dụng các hàm trên nhập vào 2 số nguyên dương  $a, b$  tìm và in ra màn hình ước chung lớn nhất của  $a$  và  $b$ .
4. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:
- Hàm nhập vào 2 số nguyên dương  $n$ , nếu  $n$  không thỏa mãn yêu cầu nhập lại.
  - Hàm tìm in ra màn hình số các chữ số của số nguyên dương  $n$  bất kỳ theo một trong hai cách:  
 Cách 1: Dùng kỹ thuật lặp.  
 Cách 2: Dùng kỹ thuật đệ quy.
  - Hàm `main()` sử dụng các hàm trên nhập vào số nguyên dương  $n$ , tìm và in ra màn hình số chữ số của  $n$ .
5. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:
- Hàm nhập vào chỉ số điện đầu tháng ( $dt$ ) và cuối tháng ( $ct$ ) của một hộ gia đình thỏa mãn  $0 < dt < ct$ , nếu không thỏa mãn yêu cầu nhập lại.
  - Hàm tính tiền điện của hộ gia đình theo đơn giá như sau:  
 100 kwh đầu tiên 1000 đồng/1kwh.  
 100 kwh tiếp theo 1700 đồng/1kwh.  
 Còn lại là 3500 đồng/1kwh.
  - Hàm `main()` nhập vào chỉ số điện đầu tháng và cuối tháng của một hộ gia đình, và in ra màn hình số kwh điện tiêu thụ và tiền phải trả của gia đình.
  - Chương trình cho phép người sử dụng tính toán với nhiều hộ gia đình theo ý muốn.
6. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:
- Hàm nhập vào số nguyên dương  $n > 1$ , nếu không thỏa mãn yêu cầu nhập lại.
  - Hàm tính số Fibonacci thứ  $n$  ( $n > 1$ ) theo định nghĩa:

$$F(n) = \begin{cases} 1 & \text{nếu } n \leq 2 \\ F(n-1) + F(n-2) & \text{nếu } n > 2 \end{cases}$$

bằng một trong 2 cách:

Cách 1: Sử dụng kỹ thuật lặp.

Cách 2: Sử dụng kỹ thuật đệ quy.

- Hàm in ra màn hình dãy k số Fibonacci đầu tiên, k nguyên dương bất kỳ.
  - Hàm main() sử dụng các hàm trên nhập vào số nguyên dương  $n > 1$ , tìm và in ra màn hình dãy n số Fibonacci đầu tiên.
7. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:
- Hàm nhập vào bán kính của một hình tròn là một số thực dương, nếu không thỏa mãn yêu cầu nhập lại.
  - Hàm tính chu vi và diện tích hình tròn (gợi ý: sử dụng cách truyền tham chiếu).
  - Hàm main() sử dụng các hàm trên nhập vào bán kính của một hình tròn, tính và in ra màn hình chu vi và diện tích hình tròn.
  - Chương trình cho phép người dùng tính toán với nhiều hình tròn theo ý muốn.
8. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:
- Hàm nhập vào một số nguyên dương  $n > 10$ , nếu không thỏa mãn yêu cầu nhập lại.
  - Hàm phân tích số nguyên dương n bất kỳ thành các thừa số nguyên tố.
  - Hàm main() sử dụng các hàm trên nhập vào số nguyên dương  $n > 10$ , phân tích n thành các thừa số nguyên tố, in kết quả ra màn hình.
9. Viết chương trình gồm các hàm thực hiện các yêu cầu sau:
- Hàm kiểm tra một ký tự là chữ cái in hay chữ cái thường trong bảng chữ cái latin hay không phải là chữ cái.
  - Hàm chuyển một chữ cái in thành một chữ cái thường.
  - Hàm chuyển một chữ cái thường thành một chữ cái in.
  - Hàm main() nhập vào một ký tự, sử dụng các hàm trên cho biết ký tự đó là chữ cái in hay chữ cái thường, hay không phải chữ cái. Nếu là chữ cái in thì đổi thành chữ cái thường và ngược lại, in kết quả ra màn hình.
10. Một người đưa thư cần giao n bức thư đến các gia đình, biết mỗi ngày người đưa thư giao hết một nửa số bức thư mà mình đang giữ. Hãy viết chương trình cho biết phải mất bao nhiêu ngày người đưa thư giao hết n bức thư bằng một trong hai cách.
- Cách 1: Sử dụng kỹ thuật lặp.
- Cách 2: Sử dụng kỹ thuật đệ quy.
11. Viết chương trình in ra màn hình mã nhị phân của một số nguyên dương n bằng một trong hai cách.
- Cách 1: Sử dụng kỹ thuật lặp.
- Cách 2: Sử dụng kỹ thuật đệ quy.