BÀI 3: CẤU TRÚC LẶP

- Cấu trúc for…
- Cấu trúc while…
- Cấu trúc do…while…

1. Bài tập hướng dẫn

1.1. Bài toán 1

Viết chương trình in ra màn hình bảng cửu chương (từ bảng 2 đến bảng 9).

Phương pháp: Sử dụng cấu trúc for… lồng nhau.

Chương trình

```
#include<conio.h>
#include<iostream.h>
#include<iomanip.h>
void main()
{
    int j,i;
    cout<<"Bang cuu chuong 2-10\n";
    for (i=1; i<=10; i++)
    {
        for (j=2; j<=10; j++)
        {
             cout<<j<<"x"<<setw(2)<<i;
             cout<<"="<<setw(2)<<(i*j)<<" ";
        }
        cout<<endl;
    }
    getch();
}
```

1.2. Bài toán 2

Chương trình kiểm tra một số nguyên nhập vào có là số nguyên tố hay không?

Phương pháp:

Số nguyên tố là một số nguyên dương lớn 1, chỉ có 2 ước là 1 và chính nó.

Để kiểm tra một số n có là số nguyên tố hay không làm theo 1 trong 2 cách:

Cách 1: Đếm xem trong số các số từ 2 đến $[\sqrt{n}]$ có bao nhiêu số là ước của n. Nếu không có ước nào thì n là số nguyên tố.

Cách 2: Kiểm tra xem trong số các số từ 2 đến $[\sqrt{n}]$ có ước nào của n không.

Chương trình

a. Cách 1: Sử dụng cấu trúc lặp for…

```
#include<conio.h>
#include<iostream.h>
#include<math.h>
void main()
{
    int i,dem;
    long n;
    do{
       cout<<"Nhap so nguyen duong n>1: ";
       cin>>n;
    }while (n<=1);
    dem=0;
    int m=(int)sqrt(n);
    for (i=2; i<=m; i++)
       if (n%i==0)
            dem++;
```

```
    if (dem==0)
       cout<<n<<" la so nguyen to";
    else
       cout<<n<<" la hop so";
    getch();
}
```

b. Cách 2: Sử dụng cấu trúc lặp while.

```
#include<conio.h>
#include<iostream.h>
#include<math.h>
void main()
{
    int i;
    long n;
    bool stop;
    do{
       cout<<"Nhap so nguyen duong n>1: ";
       cin>>n;
    }while (n<=1);
    stop=true;
    int m=(int)sqrt(n);
    i=2;
    while (i<=m && stop==true)
    {
       if (n%i==0)
       stop=false;
       i++;
    }
    if (stop==true)
       cout<<n<<" la so nguyen to";
    else
       cout<<n<<" la hop so";
    getch();
}
```

So sánh việc sử dụng for và while

Với bài toán này việc sử dụng while chương trình sẽ chạy nhanh hơn khi số n nhập vào là hợp số. Chẳng hạn với n=10000.

Vòng lặp while sẽ dừng lại ngay sau lần lặp đầu tiên vì 10000%2=0 nên stop=false.

Vòng lặp for vẫn thực hiện đủ 100 lần để đếm ước.

1.3. Bài toán 3

Chương trình tính tiền điện tiêu thụ của các hộ gia đình trong một tháng. Với công thức tính tiền như sau:

- Từ kwh thứ nhất đến kwh thứ 100 là 1000 đồng/1kwh
- Từ kwh thứ 101 đến kwh thứ 200 là 1500 đồng/1kwh.
- Từ kwh thứ 201 trở đi là 3000 đồng/1kwh.

Chương trình cho phép người dùng lặp lại công việc nhiều lần dựa vào một câu trả lời.

Phương pháp: Sử dụng cấu trúc lặp do…while…

Chương trình

```
#include<conio.h>
#include<iostream.h>
#include<ctype.h>
void main()
{
```

```
    int chisodau, chisocuoi, sokwh;
    long tientra;
    char traloi;
    do{
       cout<<"\nNhap chi so dien dau thang: ";
      cin>>chisodau;
       cout<<"Nhap chi so dien cuoi thang: ";
      cin>>chisocuoi;
      if (chisocuoi>chisodau && chisodau>=0)
      {
            sokwh=chisocuoi-chisodau;
            if (sokwh<=100)
              tientra=sokwh*1000L; //hang kieu long
            else if (sokwh<=200)
               tientra=100*1000L + (sokwh-100)*1500L;
            else
               tientra=100*1000L+100*1500L+
                           (sokwh-200)*3000L;
            cout<<"So tien phai tra: "<<tientra;
      }
      else
            cout<<"Du lieu khong hop le";
      cout<<"\nBan co tiep tuc voi gia dinh khac khong?(C/K):";
      traloi=getche();
    }while (toupper(traloi)=='C');
    getch();
}
```

2. Bài tập tự làm

1. Viết chương trình tính n!
2. Nhập vào một số nguyên, kiểm tra xem một số vừa nhập có phải là số nguyên tố không, in kết luận ra màn hình.
3. Viết chương trình nhập vào một số nguyên n, sau đó tính giá trị biểu thức:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}$$

4. Viết chương trình nhập vào một số nguyên n, sau đó tính giá trị biểu thức:

$$F = \begin{cases} 1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + ... + \frac{1}{2^n} & \text{Nếu n chẵn} \\ \sqrt{n^2+1} & \text{Nếu n lẻ} \end{cases}$$

5. Viết chương trình nhập vào một số thực x và số nguyên n, sau đó tính giá trị biểu thức:

$$S = \begin{cases} x + \frac{x^2}{3} + \frac{x^3}{3^2} + ... + \frac{x^n}{3^{n-1}} & \text{Nếu n chẵn} \\ 0 & \text{Nếu n lẻ} \end{cases}$$

6. Viết chương trình nhập vào một số nguyên n trong khoảng [10, 20] (nếu số nhập vào không thuộc khoảng đó thì yêu cầu nhập lại tới khi thoả mãn). Sau đó tính tổng các số liên tiếp từ 1 tới n.
7. Viết chương trình nhập vào một số nguyên dương n, sau đó tính tổng các giá trị chẵn, lẻ thuộc đoạn [1, n].
8. Viết chương trình nhập vào các số nguyên dương n, m, sau đó in ra:
   - Tổng các số chẵn dương trong khoảng [- n, m].
   - Tổng các số chẵn âm trong khoảng [- n, m].
   - Tổng các số lẻ dương trong khoảng [- n, m].

- Tổng các số lẻ âm trong khoảng [- n, m].

9. Viết chương trình nhập vào một số nguyên dương n, sau đó tính tổng các số nguyên tố thuộc đoạn [2..n]. Cho biết có bao nhiêu số nguyên tố thuộc đoạn đó.
10. Viết chương trình nhập vào số nguyên dương n, phân tích số n thành các thừa số nguyên tố nếu n không phải là số nguyên tố. In kết quả ra màn hình. (Ví dụ: 18=2x3x3).
11. Dùng while (sau đó viết lại, dùng do/ while) để viết chương trình tìm và in ra số a nhỏ nhất sao cho $a^2 > 1000$.
12. Viết chương trình tìm số nguyên dương n nhỏ nhất thoả mãn: 1 + 2 + 3 + … + n > 1000.
13. Để tính căn bậc hai của một số dương a, ta sử dụng công thức lặp sau:

    x(0) = a;

    x(n+1) = [x(n) * x(n) + a]/[2* x(n)] với n >0.

    Quá trình lặp kết thúc khi |(x(n+1) – x(n))/x(n)| < ε.

    và khi đó x(n+1) được xem là giá trị gần đúng của $\sqrt{a}$.

    Viết chương trình nhập số thực dương a, tính căn bậc hai của a với độ chính xác ε = 0.00001.
14. Viết chương trình tính gần đúng sin(x) với độ chính xác ε = 0.00001 theo công thức :

    $\sin(x) = x - x^3/3! + x^5/5! + \ldots + (-1)^n x^{(2n+1)}/(2n+1)!$
15. Viết chương trình tính gần đúng cos(x) với độ chính xác ε = 0.00001 theo công thức :

    $\cos(x) = 1 - x^2/2! + x^4/4! - \ldots + (-1)^n x^{2n}/(2n)!$
16. Viết chương trình tính gần đúng e^x với độ chính xác ε = 0.00001 theo công thức :

    $e^x = 1 + x/1! + x^2/2! + x^3/3! + \ldots + x^n/n!$
17. Lập trình để tính tổ hợp chập m của n theo công thức:

    C(m, n) = (n(n-1)…(n-m+1))/ m!.