

Bài thực hành số 1

CÁC LỆNH NHẬP XUẤT

Mục tiêu: Sau khi học xong 2 tuần này, sinh viên sẽ nắm được và vận dụng:

- Ý nghĩa, cách sử dụng hàm printf, scanf
- Sử dụng khuôn dạng, ký tự đặc biệt, ký tự điều khiển trong printf, scanf.

Ví dụ 1.1: In giá trị của biến nguyên a ra màn hình (với a được gán giá trị ban đầu)

```
#include<stdio.h>
#include<conio.h>
main()
{
    int a=2;
    printf("%d\n",a);
    printf("Giá trị của a là %d",a);
    getch();
}
```

Ví dụ 1.2: In giá trị của biến nguyên a ra màn hình với các cách in khác nhau

```
#include<stdio.h>
#include<conio.h>
main()
{
    int a=2, b=10;;
    printf("123456789\n");//Dung de dem do rong ky tu
    printf("%d\n",a);
    printf("%4d\n",a); printf("%-4d\n",a);
    printf("%02d\n",a); printf("%05d\n",a);
    printf("%02d\n",b); printf("%1d\n",b);
    getch();
}
```

Ví dụ 1.3: In giá trị của các biến thực ra màn hình với các cách in khác nhau.

```
#include<stdio.h>
#include<conio.h>
main()
{
    float a=5, b=1.2, c=12.34, d=12.345;
    printf("1234567890\n");//Dung de dem do rong can in ra
    printf("%f\n",a);
    printf("%6f\n",a);
    printf("%f\n",b);
    printf("%6f\n",b);
    printf("%6.2f\n",a);
    printf("%6.2f\n",b);
    printf("%6.2f\n",c);
    printf("%6.2f\n",d);
    printf("%.2f\n",a);
    printf("%.2f\n",b);
    printf("%.1f\n",c);
    printf("%-5.2f %-6.1f\n",b,c);
    getch();
}
```

Ví dụ 1.4 Nhập vào điểm số và điểm chữ, in ra các giá trị vừa nhập?

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float dSo;
    char dChu;
    clrscr();
    printf("Nhap vào diem so: ");
    scanf("%f",&dSo);
    printf("Nhap vào diem chu: ");
    fflush(stdin);
    scanf("%c",&dChu);
    printf("\nDiem so= %.2f, Diem chu=%c",dSo,dChu);
    getch();
}
```

Bài 1: Viết chương trình nhập vào 2 số nguyên a, b. In ra kết quả của phép +, -, *, / và % (phép lấy dư) của a và b.

Nhận xét kết quả của phép chia a và b.

=> Sửa lại phép chia cho đúng?

Bài 2: Viết chương trình tính diện tích các hình sau:

a. Hình chữ nhật: $S=a*b$

b. Hình thang: $S=((a+b)*h)/2$

c. Hình tròn: $S=PI * r * r$ với $PI=3.14$ (PI được khai báo là hằng số)

Bài 3: Viết chương trình nhập vào một số thực x và tính giá trị y của các biểu thức sau:

a. $y=x^7.e^x + 3x - 8$

b. $y=(x^x + 9)/(5x + 7)$

c. $y=|5x - 30| . (2x^2 + 9)$

d. $y= \sqrt{7} + x^2 (x>0)$

HD : Sử dụng các hàm có trong C để tính giá trị cho y.

`pow(double x, double y)` - tính x^y

`exp(x)` - tính e^x , hàm `fabs(x)` - tính trị tuyệt đối của một số thực x

`hàm sqrt(x)` - tính căn bậc 2 của x.

Ví dụ với $y=x^7.e^x + 3x - 8$ thì $y=pow(x, 7)*exp(x)+3*x-8$

Bài 4: Nhập vào 2 số nguyên a, b. Hãy in ra màn hình số lớn hơn.

HD: Sử dụng biểu thức ? :

max= a>=b ? a: b

Bài 5: Nhập vào một số kiểu nguyên có 3 chữ số, hãy in ra các chữ số ở hàng trăm, chục, đơn vị.

Bài 6 : Viết chương trình đổi một số nguyên hệ 10 sang hệ 2

Bài 7 : Viết chương trình đổi một số nguyên hệ 10 sang hệ 16.

Bài 8 : Viết chương trình nhập vào bán kính hình cầu, tính và in ra diện tích, thể tích của hình cầu đó.

Hướng dẫn: $S = 4\pi R^2$ và $V = (4/3)\pi R^3$.

Bài 9 : Viết chương trình nhập vào một số a bất kỳ và in ra giá trị bình phương (a^2), lập phương (a^3) của a và giá trị a^4

Bài 10: Viết chương trình nhập vào giá trị điện trở R1, R2, R3 của một mạch điện và tính tổng trở theo công thức: $\frac{1}{R} = \frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3}$

Bài 11: Nhập vào độ °C chuyển sang °F và ngược lại với công thức chuyển đổi như sau:

Đổi từ	Sang	Công thức
Fahrenheit (°F)	Celsius (°C)	$^{\circ}\text{C} = (^{\circ}\text{F} - 32) / 1.8$
Celsius (°C)	Fahrenheit (°F)	$^{\circ}\text{F} = ^{\circ}\text{C} \times 1.8 + 32$

Bài 12: Viết chương trình nhập vào 4 số thực từ bàn phím, sau đó in ra màn hình giá trị lớn nhất và giá trị nhỏ nhất?

HD: Sử dụng biểu thức điều kiện:

`max = ((a > b ? a : b) > (c > d ? c : d) ? (a > b ? a : b) : (c > d ? c : d));`

`max = ((a < b ? a : b) < (c < d ? c : d) ? (a < b ? a : b) : (c < d ? c : d));`

Bài 13: Viết chương trình nhập vào các tử và mẫu của hai phân số. Hãy cộng 2 phân số này và xuất kết quả ra màn hình.

Bài thực hành số 2

CẤU TRÚC RỄ NHÁNH

Mục tiêu: Sau khi học xong các tuần này, sinh viên sẽ hiểu và vận dụng được:

- Ý nghĩa, cú pháp các lệnh if, if...else, switch.
- Cách sử dụng các cấu trúc if...else lồng nhau
- So sánh sử dụng if và switch.

Ví dụ 2.1: Nhập vào 1 số nguyên trong khoảng từ 1-50, nếu số đó lớn hơn 10 thì in ra số lớn hơn 10. In ra số vừa nhập.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n;
    printf("\nNhập vào n trong khoảng 1-50: ");
    scanf("%d", &n);
    if(n>10)
        printf("\nSố vừa nhập lớn hơn 10");
    printf("\n Số đã nhập là %d", n);
    getch();
}
```

Ví dụ 2.1: Viết chương trình nhập vào 2 số nguyên a, b. Nếu a=b thì in ra “a=b”, ngược lại in ra “a#b”.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b;
    printf("\nNhập vào a,b: ");
    scanf("%d%d", &a, &b);
    if(a==b)
        printf("\n a=b");

    else
        printf("\n a#b ");
    getch();
}
```

Ví dụ 2.3: Viết chương trình cho bài toán:

Nhập vào 2 số nguyên a, b. Nếu a>b thì in ra “a>b”, a<b thì ra “a<b”, a=b thì in ra “a=b”.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a, b;
    clrscr();
    printf("Nhập vào a, b:");
    scanf("%d%d", &a, &b);
    if(a>b)
        printf("\a>b");
    else
```

```
        if(a<b)

                printf("\na<b");

    else

                printf("\n a=b ");

    getch();
}
```

Ví dụ 2.4: Vẽ lưu đồ thuật toán và viết chương trình giải phương trình bậc nhất $ax+b=0$

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a, b;
    clrscr();
    //Lệnh nhập vào a, b
    if(a==0)
        if(b==0)
            printf("\nPT VSN");
        else
            else
                printf("\nPT VN");

    printf("\n PT có 1 nghiệm x= %.2f",-(float)b/a);
    getch();
}
```

Ví dụ 2.5: Nhập vào 1 số từ 1-12. In ra quý tương ứng. Ngoài các số này, chương trình sẽ in ra thông báo: Bạn phải nhập số từ 1..12.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int thang;
    clrscr();
    //Lệnh nhập vào tháng
    switch(thang)
    {
        case 1:
        case 2:
        case 3: printf("\n Qui 1 "); break;
        case 4:
        case 5:
        case 6: printf("\n Qui 2 "); break;
        case 7:
        case 8:
        case 9: printf("\n Qui 3 "); break;
        case 10:
        case 11:
        case 12: printf("\n Qui 4 "); break;
        default:
            printf("\n Bạn phải nhập tháng từ 1-12");
    }
    getch();
}
```

Bài 1: Giải phương trình bậc 2: $ax^2+bx+c=0$. HD:

- Nếu $a=0$:

Nếu $b=0$:

Nếu $c=0$: PTVSN.

Ngược lại: PTVN.

Ngược lại($b \neq 0$): PT có 1 nghiệm $x = -c/b$.

- Ngược lại($a \neq 0$):

Tính $\Delta = b^2 - 4 * a * c$.

Nếu $\Delta < 0$: PTVN

Ngược lại

Nếu $\Delta = 0$: PT có 1 nghiệm kép $x = -b/(2*a)$

Ngược lại: PT có 2 nghiệm phân biệt:

Bài 2: Viết chương trình nhập vào hai số nguyên a, b và 1 trong 4 toán tử +, -, *, /, nếu toán tử là + thì in ra kết quả $x + y$, nếu là - thì in ra $x - y$, nếu là * thì in ra $x * y$, nếu là / thì in ra x / y (xét nếu $y = 0$ thì thông báo không chia được).

- Sử dụng if...else
- Sử dụng switch

Bài 3: Viết chương trình nhập vào điểm Toán, Lý và Hóa của một sinh viên. Sau đó tính và xuất ra màn hình điểm trung bình cùng kết quả xếp loại của sinh viên này biết:

$dtb < 4$: Loại F

$4 \leq dtb < 5.5$: Loại D

$5.5 \leq dtb < 7$: Loại C

$7 \leq dtb < 8.5$: Loại B

$8.5 \leq dtb \leq 10$: Loại A

Bài 4: Viết chương trình nhập vào 4 số nguyên. Tìm và in ra số lớn nhất.

Hướng dẫn: Ta có 4 số nguyên a, b, c, d. Tìm 2 số nguyên lớn nhất x, y của 2 cặp (a, b) và (c, d). Sau đó so sánh 2 số nguyên x, y để tìm ra số nguyên lớn nhất.

Bài 5: Viết chương trình nhập vào điểm 3 môn thi: Toán, Lý, Hóa của học sinh. Nếu tổng điểm ≥ 15 và không có môn nào dưới 4 thì in kết quả đậu. Nếu đậu mà các môn đều lớn hơn 5 thì in ra lời phê "Học đều các môn", ngược lại in ra "Học chưa đều các môn", các trường hợp khác là "Thi hỏng".

Bài 6: Viết chương trình tính tiền cước TAXI. Biết rằng:

- Km đầu tiên là 12000đ.
- Mỗi km tiếp theo là 11000đ.
- Nếu lớn hơn 30 km, thì mỗi km sẽ thêm 9000đ.

Hãy nhập số km, sau đó in ra số tiền phải trả.

Bài 7: Viết chương trình nhập 3 số nguyên từ bàn phím.

- Xét xem 3 số vừa nhập có tạo thành tam giác hay không?

- Nếu có hãy in ra đó là tam giác gì?(đều, cân, vuông hay vuông cân,thường)
- Tính chu vi và diện tích tam giác.

HD: - Nếu a,b,c là 3 cạnh của 1 tam giác thì phải thỏa mãn điều kiện:

$$(a+b)>c \text{ và } (a+c)>b \text{ và } (b+c)>a$$

- Chu vi tam giác: $p = a+b+c$
- Diện tích tam giác $s = \sqrt{p(p-a)(p-b)(p-c)}$

Bài 8: Viết chương trình nhập vào tháng, in ra tháng đó có bao nhiêu ngày.

Hướng dẫn: Nhập vào tháng

Nếu là tháng 1, 3, 5, 7, 8, 10, 12 thì có 30 ngày

Nếu là tháng 4, 6, 9, 11 thì có 31 ngày

Nếu là tháng 2 và là năm nhuận thì có 29 ngày ngược lại 28 ngày

(Năm nhuận là năm chia chắn cho 4)

Bài 9: Viết chương trình nhập vào 1 số có 3 chữ số (Kiểm tra trường hợp nhập sai) Hãy in ra cách đọc số này.

Ví dụ: Số= 125, In ra: “Một trăm hai mươi lăm”.

Số= 105, In ra “Một trăm lẻ năm”

HD: - Nhập số có 3 chữ số.

- Lấy được các chữ số hàng trăm, hàng chục hàng đơn vị.
- Dùng 3 câu trúc switch để xét từng chữ số: trăm, chục, đơn vị

Bài 10: Viết chương trình nhập vào giờ, phút, giây, giây thêm (số giây thêm vào). In ra kết quả tương ứng.

Ví dụ: 00:01:15, nhập số giây thêm= 50 thì in ra kết quả: 00:02:05

HD: - Khai báo 3 biến giờ, phút, giây, giâythêm.

Nếu giây+giâythêm<60: giây= giây+giây thêm.

Ngược lại

$$\text{giây} = (\text{giây} + \text{giây thêm}) - 60$$

$$\text{phút} = \text{phút} + 1$$

Nếu phút>=60 thì

$$\text{phút} = \text{phút} - 60$$

$$\text{giờ} = \text{giờ} + 1.$$

Bài thực hành số 3

CẤU TRÚC LẶP

Mục tiêu: Sau khi học xong các tuần này, sinh viên sẽ hiểu và vận dụng được:

- Ý nghĩa, hoạt động, cú pháp của các vòng lặp: for, while, do...while
- So sánh giữa các vòng lặp.
- Lệnh goto, break và continue.
- Các vòng lặp lồng nhau.

Ví dụ 3.1: Nhập n không âm, in ra các số từ n đến 0.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i; clrscr();
    printf("Nhap n: ");
    scanf("%d",&n);
    for( i=n; i>=0; i--)
        printf("%d\t",i);
    getch();
}
```

Lưu ý: Không được có dấu ; sau câu lệnh for

Ví dụ 3.2 : Nhập n dương, tính tổng các số từ 1-n.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,s;
    clrscr();
    printf("Nhap n: ");
    scanf("%d",&n);
    s=0;
    for( i=1; i<=n; i++)
        s=s+i; //hoặc có thể viết s+=i;
    printf("\nTong cac so tu 1 den %d la %d",n,s);
    getch();
}
```

Ví dụ 3.3: Nhập 3 số thực từ bàn phím, hãy tính tổng của chúng.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float n, s;
    int i;
    clrscr();
    s=0;
    for( i=1; i<=3; i++)
    {
        printf("Nhap so thu %d ",i);
        scanf("%f",&n);
        s+=n;
    }
    printf("\nTong 3 so nhap vao la %.2f",s);
    getch();
}
```


Ví dụ 3.4: Nhập n không âm, in ra các số từ n đến 0.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n; clrscr();
    printf("Nhap n: ");
    scanf("%d", &n);
    while(n>=0)
    {
        printf("%d\t", n);
        n--;
    }
    getch();
}
```

Chạy chương trình và nhận xét kết quả.

Ví dụ 3.5: Viết chương trình nhập vào một số nguyên n (kiểm tra nếu $n<0$ thì nhập lại và lặp lại việc nhập này cho đến khi $n\geq 0$). Sau đó xuất ra giá trị n vừa nhập.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n;
    clrscr();
nhaplai:
    printf("Nhap n: ");
    scanf("%d", &n);
    if(n<0)
    {
        printf("Ban da nhap sai, nhap lai n: ");
        goto nhaplai;
    }
    printf("\nGia tri n=%d", n);
    getch();
}
```

Cách 2: Sử dụng lệnh while

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n; clrscr();
    printf("Nhap n: ");
    scanf("%d", &n);
    while(n<0)
    {
        printf("Ban da nhap sai, nhap lai n: ");
        scanf("%d", &n);
    }
    printf("\nGia tri n=%d", n);
    getch();
}
```

Lệnh break và continue:

a. Lệnh break: Dùng để thoát khỏi vòng lặp chứa nó khi thỏa mãn điều kiện nào đó.

Ví dụ 3.6: Cho phép người dùng nhập liên tục số n nguyên, cho đến khi nhập số âm thì dừng

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n;
    clrscr();
    while(1)
    {
        printf("Nhap so thu n: ");
        scanf("%d",&n);
        if(n<0) break;
    }
    getch();
}
```

Chạy chương trình và nhận xét.

Thử thay vòng lặp while ở trên bằng vòng lặp for.

Lệnh continue: Dừng để bỏ qua 1 lần lặp khi thỏa mãn một điều kiện nào đó.

Ví dụ 3.7: In ra màn hình các số từ 1 đến 10, bỏ qua số 2 và số 5.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i;
    clrscr();
    for(i=1; i<=10; i++)
    {
        if(i==2 || i==5) continue;
        printf("%d\t",i);
    }
    getch();
}
```

Ví dụ 3.8: Viết chương trình vẽ một hình chữ nhật bằng các dấu * với chiều dài và chiều rộng do người dùng nhập vào.

```
include<stdio.h>
#include<conio.h>
void main()
{
    int cd,cr,i,j;
    clrscr();
    printf("Nhap chieu dai, chieu rong: ");
    scanf("%d%d",&cd,&cr);
    printf("\n");
    for(i=1; i<=cr; i++)
    {
        for(j=1; j<=cd ; j--)
            printf("*"); //In mot hang dau *
        printf("\n"); //Xuong hang khi in xong 1 hang o tren
    }
    getch();
}
```

Bài tập

Bài 1: Viết chương trình nhập n (có kiểm tra điều kiện n) để tính các tổng sau:

a) $S1 = 1 + 3 + 5 + \dots + (2 \cdot n + 1)$.

b) $S2 = 2 + 4 + 6 + \dots + 2 \cdot n$

c) $S3 = 1 + 1/2 + 1/3 + \dots + 1/n$

d) $S4 = 1 + 1/2! + 1/3! + \dots + 1/n!$

Bài 2: Nhập vào số nguyên dương n, hãy in ra các ước số của n.

Bài 3: Kiểm tra xem một số nguyên dương n có phải là số hoàn hảo hay không? (Số hoàn hảo là số có tổng tất cả các ước số bằng hai lần chính nó).

Bài 4: Kiểm tra xem một số nguyên dương n có là số nguyên tố hay không?

Bài 5: Nhập vào 1 số n nguyên dương. In ra số ngược lại. Ví dụ n=123, in ra 321.

Bài 6: Tính tổng của các chữ số của một số tự nhiên n. Đếm xem n có bao nhiêu chữ số. HD: Dùng vòng lặp: Để lấy chữ số hàng đơn vị, ta thực hiện phép lấy phần dư: $dv = n \% 10$; Sau đó, cộng vào tổng: $s += dv$; sau đó, Tăng biến đếm $dem++$ (để tính số chữ số), tiếp tục chia số $n/10$ để lấy hàng chục, rồi cộng vào s, tiếp tục lặp cho đến khi $n=0$ (tức đã lấy hết các chữ số của n).

Bài 7: Viết chương trình tính giá trị của đa thức

$$P_n = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0$$

Hướng dẫn đa thức có thể viết lại

$$P_n = (\dots(a_n x + a_{n-1})x + a_{n-2})x + \dots + a_0$$

Như vậy trước tiên tính $a_n x + a_{n-1}$, lấy kết quả nhân với x, sau đó lấy kết quả nhân với x cộng thêm a_{n-2} , lấy kết quả nhân với x ... n gọi là bậc của đa thức.

Bài 8: Viết chương trình tìm các số nguyên a, b, c, d khác nhau trong khoảng từ 0 tới 10, thỏa mãn điều kiện $a \cdot d \cdot d = b \cdot c \cdot c \cdot c$.

Bài 9: Viết chương trình tìm các số nguyên gồm 3 chữ số sao cho tích của 3 chữ số bằng tổng 3 chữ số. Ví dụ: $1 \cdot 2 \cdot 3 = 1 + 2 + 3$.

Bài 10: Nhập số n gồm k chữ số, tính giá trị trung bình các chữ số của n. Ví dụ: n=1234. In ra TB= 2.50

HD: Tương tự bài 6 nhưng thêm phần tính giá trị trung bình các chữ số được xác định:

$$TB = (\text{float}) s / dem;$$

Bài 11: Tìm ước số chung lớn nhất và bội số chung nhỏ nhất của hai số nguyên x, y.

HD: Thuật toán phải dùng cho cả số dương lẫn số âm do đó ta phải dùng hàm trị tuyệt đối của 2 số.

Dùng thuật toán lặp: tính UCLN bằng cách lặp việc tính hiệu của trị tuyệt đối x, y cho đến khi hiệu bằng 0.

$$\text{Tính BCNN} = |x \cdot y| / \text{UCLN}.$$

Nếu một trong 2 số bằng 0 ví dụ $x=0$ và $y \neq 0$ thì $\text{UCLN} = |y|$.

Các bước: - Nhập giá trị x, y.

- Gọi hàm $a=\text{abs}(x)$, $b=\text{abs}(y)$; //Cần giữ lại giá trị x , y .
- Lặp $a!=b$ và $a*b!=0$ (xét TH một trong 2 số bằng 0)
Nếu $a>b$ thì $a-=b$, ngược lại thì $b-=a$.
- Nếu $a*b=0$ thì in ra $\text{UCLN}=a+b$, $\text{BCNN}=|x*y|/(a+b)$.
- Ngược lại thì in ra $\text{UCLN}=a$, $\text{BCNN}=|x*y|/a$.

Bài 12: Viết chương trình giải bài toán cổ:

Vừa gà vừa chó
Bó lại cho tròn
36 con, 100 chân chẵn
Hỏi có bao nhiêu gà, bao nhiêu chó?

HD:

```
Gọi x là số gà, y là số chó
x>0 và y>0 x+y=36 , x<36
{
for (x=1; x<36; x++)
    for (y=1; y<25; y++)
        if (2*x+4*y==100 && x+y==36)
            printf("\nSố gà: %d, và số chó: %d", x, y);
}
```

Bài 13: Viết chương trình giải bài toán cổ:

Trăm trâu, trăm cỏ
Trâu đứng ăn năm
Trâu nằm ăn ba
Trâu già ba con một bó
Cho biết có bao nhiêu trâu đứng, trâu nằm, trâu già?

HD: Tương tự như bài toán trên, tuy nhiên biểu thức điều kiện sẽ

```
khác: if (td*5+tn*3+tg*1.0/3==100 && td+tn+tg==100)
```

Bài 14: Viết chương trình in ra bảng cửu chương.

Bài 15: Viết chương trình in ra dãy Fibonacci $f_1 =$

$f_0 = 1$;

$f_n = f_{n-1} + f_{n-2}$; ($n > 1$)

Bài 16: Viết chương trình lặp đi lặp lại nhiều lần công việc nhập 1 ký tự và in ra mã ASCII của ký tự đó, khi nào nhập số 0 thì dừng.

Bài thực hành số 4

HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Mục tiêu: Sau khi hoàn tất bài này sinh viên sẽ hiểu và vận dụng được:

- Khái niệm, cách khai báo về hàm.
- Sử dụng biến cục bộ, toàn cục trong hàm.
- Sử dụng tiền xử lý #define
- Đệ qui

Ví dụ 4.1: Nhập vào 2 số nguyên, in ra số lớn hơn.

```
#include<stdio.h>
#include<conio.h>
int max2so(int x, int y); //Khai báo nguyên mẫu hàm.
void main()
{
    int a,b;
    clrscr();
    printf("Nhap 2 so: ");
    scanf("%d%d",&a,&b);
    printf("\nSo lon hon la %d ",max2so(a,b));
    getch();
}
int max2so(int x, int y)
{
    return (x>y?x:y);
}
```

Ví dụ 4.2: Viết chương trình in ra các ước số của n dương nhập từ bàn phím, chương trình có sử dụng hàm.Có thể viết theo cách 1 hoặc cách 2 đều được.

```
#include<stdio.h>
#include<conio.h>
void InUS(int n)
{
    int i;
    for(i=1; i<=n; i++)
        if(n%i==0)
            printf("%d\t",i);
}
void main()
{
    int n; clrscr();
    printf("Nhap n: ");
    scanf("%d",&n);
    printf("\nCac uoc so cua %d la \n ",n); InUS(n);
    getch();
}
```

Ví dụ 4.3: Xây dựng hàm tính giai thừa của một số n.

$n! = 1*2*3*...*(n-2)*(n-1)*n$ với $n \geq 1$ và $0! = 1$.

Hàm tính giai thừa của n không sử dụng đệ qui (dùng vòng lặp)

```
#include<stdio.h>
#include<conio.h>
long giaithua(int n)
{
    int i;
    long gt=1;
    if(i==0) return 1;
    else
        for(i=1;i<=n;i++)
            gt*=i;
    return gt;
}
void main()
{
    int n;
    clrscr();
    printf("Nhap vao n: ");
    scanf("%d",&n);
    printf("\n n!= %ld ",giaithua(n));
    getch();
}
```

Hàm tính giai thừa của n sử dụng đệ qui.

```
long giaithua(int n)
{
    if(n==0)
        return 1;
    else
        return n*giaithua(n-1)
}
```

Ví dụ 4.4: Viết chương trình in ra dãy Fibonacci

$f_1 = f_0 = 1$;

$f_n = f_{n-1} + f_{n-2}$; $(n > 1)$

HD: Hàm đệ qui dãy Fibonacci

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

Bài tập

Viết chương trình có sử dụng hàm:

Bài 1: Tính: $S = 1 + 1/2! + 1/3! + \dots + 1/n!$

Bài 2: Tính $n!$

Bài 3: Viết hàm tính diện tích hình tròn và hình thang, hình chữ nhật. Hãy viết hàm menu để cho phép người dùng lựa chọn để tính diện tích từng hình.

HD: Sử dụng cấu trúc switch để lựa chọn giá trị người dùng nhập vào.

Bài 4: Viết hàm kiểm tra n có phải là số hoàn hảo hay không?

Bài 5: Xây dựng hàm kiểm tra số n có phải là số nguyên tố hay không. In ra các số nguyên tố <1000.

Bài 6: Xây dựng hàm tính UCLN và BCNN (phương pháp lặp trừ) **Bài**

7: Xây dựng hàm đệ qui tính UCLN theo thuật toán Euclid: HD: -

Thuật toán Euclid được mô tả như sau:

Giả sử $a = bq + r$, với a, b, q, r là các số nguyên, ta có:

$$\text{UCLN}(a, b) = \begin{cases} b & \text{nếu } r = 0 \\ \text{UCLN}(b, r) & \text{nếu } r \neq 0 \end{cases}$$

Trong đó $r = a \bmod b$

Bài 8: Viết hàm đệ qui tính tổng n số nguyên dương đầu tiên:

$\text{tong}(n) = n + \text{tong}(n - 1)$

Bài 9: Viết hàm đệ qui tính $S(n) = 1 + 3 + 5 + \dots + (2.n + 1)$ với $n \geq 0$.

Bài 10: Hàm tìm ước số chung của hai số nguyên dương. Trong hàm `main()` thực hiện công việc cộng hai phân số kết quả cuối cùng là một phân số có rút gọn.

Bài 11: Viết hàm đệ qui tính $S(n) = 1 + 1/2 + 1/3 + \dots + 1/n$ với $n > 0$

Bài 12: Xây dựng hàm đệ qui tính x^n

Bài 13: Viết hàm đệ qui tính số hạng thứ n trong dãy Fibonacci.

Bài 14: Viết hàm đệ qui in đảo ngược số n nguyên dương.

Bài 15: Viết hàm đệ qui tính đổi sang hệ nhị phân của số nguyên dương n .

Bài thực hành số 5

CON TRỎ

Mục tiêu: Sau khi học xong phần này, sinh viên sẽ hiểu và vận dụng được:

- Ý nghĩa, phương pháp hoạt động của đệ quy.
- Có thể thay vòng lặp bằng đệ quy.

Ví dụ: Xét 2 hàm hoán vị của 2 số nguyên a, b được viết theo 2 cách:

```
#include<stdio.h>
#include<conio.h>
void ThamTri(int a, int b);
//Khái báo hàm hoán vị theo kiểu truyền tham trị
void ThamChieu(int *a, int *b);
//Khái báo hàm hoán vị theo kiểu truyền tham chiếu
void ThamTri(int a, int b)
{
    int tam;
    tam=a;
    a=b;
    b=tam;
}
void ThamChieu(int *a, int *b)
{
    int tam;
    tam=*a;
    *a=*b;
    *b=tam;
}
void main()
{
    int a,b;
    clrscr();
    printf("Nhap 2 so a,b: ");
    scanf("%d%d",&a,&b);
    printf("Truyen theo kieu tham tri: ");
    ThamTri(a,b);
    printf("\nSau khi hoan doi: a=%d, b=%d",a,b);
    printf("\n+++++Truyen theo kieu tham chieu+++++\n");
    ThamChieu(&a,&b);
    printf("\nSau khi hoan doi: a=%d, b=%d",a,b);
    getch();
}
```

Bài tập

Bài 1: Viết chương trình nhập, xuất giá trị của n.

HD: Xây dựng chương trình gồm 2 hàm con nhập và xuất với tham số của hàm là n, hàm main gọi 2 hàm này.

Bài 2: Viết chương trình xây dựng hàm tính diện tích hình thang với kết quả trả về nằm ở biến chứa kết quả kq.

HD: Để trả về kết quả ở tham số là biến diện tích, thì biến diện tích phải là con trỏ.

Định nghĩa hàm:

```
void dththang(float daylon, float daynho, float chieucaao, float *kq)
```

Với:

```
*kq=(daylon+daynho)*chieucaao/2;
```

Bài 3: Viết chương trình xây dựng hàm giải phương trình bậc hai với nghiệm nằm ở biến chứa nghiệm x1,x2.

HD: Xây dựng hàm:

```
void ptbac2(float a, float b, float c, float *x1, float *x2)
```

Bài 4: Xây dựng hàm tính tổng các chữ số của 1 số nguyên dương n (kiểm tra nhập n). Và đếm các chữ số của n.

HD: thuật toán tương tự, nhưng ở đây trả về 2 giá trị: tổng chữ số và đếm chữ số, do đó ta dùng 2 tham số kiểu con trỏ để xác định 2 giá trị này.

Bài thực hành số 6

DỮ LIỆU KIỂU MẢNG

Mục tiêu: Sau khi học xong phần này, sinh viên sẽ hiểu và vận dụng được:

- Ý nghĩa, cách sử dụng mảng.
- Các thao tác cơ bản trên mảng
- Dùng mảng làm tham số cho hàm.

1. Mảng một chiều

Ví dụ : Nhập vào một số nguyên n ($n > 0$), tính tổng các phần tử có trong mảng.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[20],n,i,s;
    clrscr();
    printf("Nhập vào n: ");
    scanf("%d",&n);
    //Doan code nhap cac phan tu cho mang
    for(i=0; i<n;i++)
    {
        printf("\nNhập phần tử a[%d]: ",i);
        scanf("%d",&a[i]);
    }
    //Doan code in cac phan tu cua mang ra man hinh
    for(i=0; i<n;i++)
        printf("\n a[%d]=%d",i,a[i]);
    //Tính tổng các phần tử của mảng
    s=0;
    for(i=0; i<n;i++)
        s+=a[i];
    //In ra giá trị tổng
    printf("\nTổng các phần tử có trong mảng=%d",s);
    getch();
}
```

2. Mảng 2 chiều

Ví dụ : Nhập dữ liệu cho mảng 2 chiều

```
//Doan code nhap cac phan tu cho mang 2 chieu
for(i=0; i<n;i++) //chạy từ 0->n-1 để nhập giá trị cho hàng
for(j=0; j<m ; j++)//chạy từ 0->m-1 để nhập giá trị cho cột
{
    printf("\nNhập phần tử a[%d][%d]: ",i,j);
    scanf("%d",&a[i][j]);
}
```

Ví dụ: In các phần tử của mảng 2 chiều

```
//Doan code nhap cac phan tu cho mang 2 chieu
for(i=0; i<n;i++) //chạy từ 0->n-1 để in phần tử thuộc hàng
{
    for(j=0; j<m ; j++)//chạy từ 0->m-1 để in phần tử cho cột
        printf("\n a[%d][%d]= %d ",i,j,a[i][j]);
    printf("\n");//Xuống hàng để in hàng tiếp theo
}
```

3. Dùng mảng làm tham số cho hàm

Ví dụ: Xây dựng các hàm nhập, xuất mảng, và hàm tìm số lớn nhất trong mảng:

```
#include<stdio.h>
#include<conio.h>
//Khai bao cac nguyen mau ham
void nhap(int a[],int n);
void xuat(int a[], int n);
int max(int a[],int n);
void main()
{
    int a[20],n,i;
    clrscr();
    printf("Nhap so phan tu cua mang: ");
    scanf("%d",&n);
    printf("Nhap cac phan tu:");
    nhap(a,n);
    printf("\nMang da nhap:");
    xuat(a,n);
    //In ra gia tri max
    printf("\n Phan tu lon nhat trong mang=%d",max(a,n));
    getch();
}
void nhap(int a[], int n)
{
    int i;
    //Doan code nhap cac phan tu cho mang
    for(i=0; i<n;i++)
    {
        printf("\nNhap phan tu a[%d]: ",i);
        scanf("%d",&a[i]);
    }
}
void xuat(int a[], int n)
{
    int i;
    for(i=0; i<n;i++)
        printf("\n a[%d]=%d",i,a[i]);
}
int max(int a[], int n)
{
    int i, max;
    max=a[0]; //Gan max= phan tu dau tien
    for(i=1;i<n;i++)
        if(a[i]>max)
            max=a[i];
    return max;
}
```

Bài tập

Bài 1: Viết chương trình thao tác trên một mảng số nguyên gồm n phần tử. Yêu cầu:

- a. Viết hàm nhập mảng
- b. Viết hàm xuất mảng
- c. In ra các phần tử là số nguyên tố có trong mảng.
- d. Tính tổng các số nguyên dương chia hết cho 5.

- e. **Đếm số lần xuất hiện của một giá trị X nào đó được nhập từ bàn phím.**
- f. Tìm kiếm 1 phần tử X có trong mảng hay không?
- g. Thay thế phần tử X có trong mảng bằng phần tử Y nhập từ bàn phím.
- h. Sắp xếp mảng theo thứ tự tăng dần.

Bài 2: Viết hàm kiểm tra mảng có đối xứng hay không, mảng có phải là mảng tăng hay không? Nếu không phải mảng tăng, hãy sắp xếp mảng tăng dần.

Bài 3: Viết hàm kiểm tra mảng có phải là mảng chứa toàn số nguyên tố?

HD: Xây dựng hàm kiểm tra số nguyên tố. Sau đó, lặp xét từng phần tử của mảng, nếu có 1 phần tử không phải là số nguyên tố thì thoát khỏi vòng lặp và mảng không chứa toàn số nguyên tố. Ngược lại, mảng chứa toàn số nguyên tố.

Bài 4. Viết chương trình nhập mảng các số nguyên N và liệt kê các số nguyên tố nhỏ hơn N, nếu mảng không tồn tại số nguyên tố nào nhỏ hơn N thì phải xuất ra một thông báo.

Bài 5. Viết hàm đếm các phần tử là số nguyên tố trong mảng.

Bài 6. Viết chương trình nhập vào mảng một chiều các số nguyên và xuất ra màn hình các phần tử là số chính phương nằm tại những vị trí lẻ trong mảng.

Bài 7: Viết chương trình nhập vào 1 dãy số thực dương rồi in tổng các số dương đó (với số lượng phần tử chưa biết trước, số lượng phần tử mảng chứa tối đa là 20). Xét trường hợp người dùng nhập số lượng phần tử >20?

Bài 8: Hãy viết chương trình gồm các hàm sau:

- Nhập ma trận
- In ma trận ra màn hình
- Cộng 2 ma trận
- Nhân 2 ma trận

Trong hàm main() yêu cầu nhập hai ma trận A nxn, B nxn , In ra màn hình ma trận A, ma trận B , ma trận tổng, ma trận tích.

HD:

```
void tong_matran(float A[n][n], float B[n][n], float T[n][n])
{
    for(int i=1; i<=n; i++)
        for(int j=1; j<=n; j++)
            T[i][j]=A[i][j]+B[i][j];
}

void tich_matran(float A[n][n], float B[n][n], float Ti[n][n])
{
    for(int i=1; i<=n; i++)
        for(int j=1; j<=n; j++)
        {
            float sohang=0;
            for(int k=1; k<=n; k++)
                sohang=sohang+A[i][k]*B[k][j];
            Ti[i][j]=sohang;
        }
}
```

Bài thực hành số 7
DỮ LIỆU KIỂU XÂU KÝ TỰ

Mục tiêu: Sau khi học phần này, sinh viên sẽ hiểu và vận dụng được:

- Cách sử dụng chuỗi.
- Các hàm về chuỗi và các thao tác trên nó

Bài tập

Bài 1: Nhập vào 1 chuỗi, đổi chuỗi thành chữ hoa, chữ thường, chữ hoa đầu từ.

HD: Dùng các hàm có sẵn.

Đối với thuật giải đổi chữ hoa đầu từ:

trước hết cần chèn ký tự trống ' ' vào phía trước chuỗi st để quy luật được áp dụng cho cả chuỗi sử dụng hàm for cho biến i chạy với điều kiện $i+1 \leq \text{len}(st)-1$ và thực hiện: Kiểm tra nếu $st[i]=' '$ và $st[i+1]!=' '$ thì chuyển $st[i+1]$ thành chữ hoa bằng hàm `toupper()`

Bài 2: Nhập vào 1 chuỗi, kiểm tra chuỗi đã nhập có phải là chuỗi đối xứng hay không?

Ví dụ:

- Chuỗi S1= "aBcdCBa"; //S1 là chuỗi đối xứng.
- Chuỗi S2="aBcdBCa"; //S2 là chuỗi không đối xứng.

Bài 3: Nhập vào 1 chuỗi, tìm ký tự xuất hiện nhiều nhất trong chuỗi và số lần xuất hiện của nó.

Bài 4: Nhập vào 1 chuỗi và 1 từ, hãy cho biết từ đó xuất hiện trong chuỗi bao nhiêu lần.

Bài 5: Viết chương trình đếm có bao nhiêu khoảng trắng trong chuỗi.

Bài 6. Viết chương trình nhập vào một chuỗi, hãy loại bỏ những khoảng trắng thừa trong chuỗi.

Bài 7. Viết chương trình đổi những ký tự đầu tiên của mỗi từ thành chữ in hoa.

Bài 8. Nhập một chuỗi bất kỳ, yêu cầu nhập một ký tự muốn xóa. Thực hiện xóa tất cả những ký tự đó trong chuỗi.

Bài thực hành số 7

DỮ LIỆU TỰ ĐỊNH NGHĨA VÀ DỮ LIỆU KIỂU CẤU TRÚC

Mục tiêu: Sau khi học xong phần này, sinh viên sẽ nắm và vận dụng được:

- Cách dùng các kiểu dữ liệu tự định nghĩa, kiểu cấu trúc.
- Các thao tác trên kiểu cấu trúc

Ví dụ 7.1: Nhập vào số trong khoảng từ 0-6, in ra thứ tương ứng

```
#include<stdio.h>
#include<conio.h>
enum Days {Sun, Mon, Tue, Wed, Thu, Fri, Sat};
void main()
{
    enum Days d;
    int i;
    clrscr();
    printf("Nhap 1 thu (0:Sun-->6: Sat): ");
    scanf("%d", &i);
    switch(i){
        case Sun:
            printf("Sunday");
            break;
        case Mon:
            printf("Monday");
            break;
        case Tue:
            printf("Tuesday");
            break;
        case Wed:
            printf("Wednesday");
            break;
        case Thu:
            printf("Thursday");
            break;
        case Fri:
            printf("Friday");
            break;
        case Sat:
            printf("Saturday");
            break;
    }
    getch();
}
```

Ví dụ 7.2: Nhập và in danh sách n sinh viên gồm các thông tin mã sv, hoten, điểm trung bình.

```
#include<conio.h>
#include<stdio.h>
#define MAX 50
struct sinhvien
{
    int masv;
    char hoten[30];
    float dtb;
};
```

```
void main()
{
    struct sinhvien listSV[MAX];
    int i,n;
    float tam;
    clrscr();
    printf("Nhap so sinh vien: ");
    scanf("%d",&n);
    //Nhap danh sach sinh vien
    for(i=0;i<n;i++)
    {
        printf("\n Nhap sinh vien thu %d:",i+1);
        printf("\nNhap ma sinh vien: ");
        scanf("%d",&listSV[i].masv);
        printf("Nhap ho ten: ");
        fflush(stdin);
        gets(listSV[i].hoten);
        printf("Nhap vao diem trung binh: ");
        scanf("%f",&tam);
        listSV[i].dtb=tam;
    }
    printf("\n*****");
    printf("\nDanh sach sinh vien\n");
    printf("\nMaSV || HoTen|| DTB ");
    for(i=0; i<n; i++)
    printf("\n%d %-20s %8.2f",i+1, listSV[i].hoten, listSV[i].dtb);
    getch();
}
```

Bài tập:

Bài 1: Viết chương trình nhập và in một danh sách n học sinh với các thành phần tên, đtb, lớp. Sau đó sắp xếp đtb theo thứ tự tăng dần và in danh sách đã sắp xếp ra màn hình. Các công việc nhập danh sách, in danh sách, sắp xếp danh sách được thực hiện ở các hàm. Trong đó hàm sắp xếp danh sách có sử dụng hàm hoán vị.

Bài 2: Viết chương trình nhập vào danh sách n học sinh gồm các thành phần: mã sinh viên, họ tên, điểm(trong đó điểm là cấu trúc gồm 3 thành phần: Toán, Lý, Hóa).

Yêu cầu:

- Hãy in danh sách sinh viên dưới dạng cột gồm các thông tin:

Mã sv || Họ tên || Điểm TB

- In ra danh sách sinh viên có điểm TB >5, nếu không có sinh viên nào điểm TB >5 hãy in ra “Không có sinh viên nào có điểm TB trên 5 điểm”.
- Sắp xếp danh sách trên theo thứ tự tăng dần của điểm TB.

Bài 3: Viết chương trình nhập vào tên, địa chỉ, sắp xếp tên và địa chỉ theo thứ tự alphabet, sau đó hiển thị danh sách đã được sắp xếp.

Bài thực hành số 8

THAO TÁC TRÊN FILE

Mục tiêu: Sau khi học xong phần này, sinh viên sẽ nắm và vận dụng được:

- Mở, đóng file
- Ghi, đọc file số nguyên, mảng, chuỗi.
- Một số hàm xử lý tập tin.

Ví dụ 8.1: Đọc ghi 1 số nguyên và ghi dữ liệu vào file

```
/* Ghi n so nguyen vao file va doc ra tu file*/
#include <stdio.h>

#include <conio.h>
#include <stdlib.h>
void main(void)
{
    FILE *f;
    int in, i;
    printf("Nhap vao so n: ");
    scanf("%d", &in);
    //Ghi file
    if((f = fopen("int_data.txt", "wb")) == NULL)    //mo file
    {
        printf("Khong the mo file!.\n");
        exit(0);
    }
    else
        for(i = 1; i <= in; i++)
            fwrite(&i, sizeof(int), 1, f);    //ghi file
    fclose(f); //dong file
    //Doc file
    f = fopen("int_data.txt", "rb");
    while(fread(&i, sizeof(int), 1, f) == 1)
        printf("%d ", i);
    fclose(f);
    getch();
}
```

Ví dụ 8.2: Đọc và ghi một mảng số nguyên.

```
/* Ghi 10 so nguyen vao file va doc ra tu file*/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define MAX 50
void main(void)
{
    FILE *f;
    int i, ia[MAX], ib[MAX];
    for (i = 0; i < 10; i++)
    {
        printf("Nhap vao mot so: ");
        scanf("%d", &ia[i]);
    }
    if((f = fopen("array.txt", "w")) == NULL)
```



```
{
    printf("Khong the mo file!\n");
    exit(0);
}
fwrite(ia, sizeof(ia), 1, f); //ghi mang vao file
fclose(f);
f = fopen("array.txt", "r");
fread(ib, sizeof(ib), 1, f); //doc mang tu file
for (i = 0; i < 10; i++)
    printf("%d ", ib[i]);
fclose(f);
getch();
}
```

Bài tập

Bài 1: Viết chương trình đọc mảng số nguyên từ tập tin MANG.INP (Nội dung tập tin MANG.INP chỉ gồm 1 dòng, ghi giá trị các phần tử của mảng (là các số tự nhiên ≤ 999), mỗi giá trị cách nhau bởi một khoảng trắng) sau đó sắp xếp theo thứ tự tăng dần. Kết quả được ghi vào tập tin MANG.OUT. Ví dụ:

File MANG.INP gồm: 2 5 1 7 -9 8

File MANG.OUT gồm:

Mang doc tu tap tin MANG.INP la:

2 5 1 7 -9 8

Mang sau khi sap xep:

-9 1 2 5 7 8

Bài 2: Thực hiện việc đọc danh sách n sinh viên từ bàn phím gồm các thông tin: masv, họ tên, điểm. Sau đó, ghi dữ liệu này ra file gồm:

Dòng thứ 1: Số sv

Dòng thứ 2: MaSV || Ho Ten || Diem

(Ghi dữ liệu của n sinh viên)

Sau khi dữ liệu được ghi ra file, hãy đọc dữ liệu từ file này ra màn hình.

HD: Đối với kiểu dữ liệu cấu trúc, duyệt đọc từng dữ liệu của mỗi sinh viên, việc ghi dữ liệu như sau:

```
fwrite(&listSV[i], sizeof(sinhvien), 1, f);
```

Tương tự, việc đọc dữ liệu:

```
fread(&listSV[i], sizeof(sinhvien), 1, f);
```

-----❧-----

