CHƯƠNG 3 HÀM CON

Trình bày cấu trúc của một chương trình, các bước xây dựng cài đặt chương trình theo phương pháp thủ tục hàm và một số kỹ thuật liên quan.

I. TÓM TẮT LÝ THUYẾT

I.1. Khái niệm

Hàm là một đoạn chương trình độc lập **thực hiện trọn vẹn một công việc nhất định** sau đó trả về giá trị cho chương trình gọi nó, hay nói cách khác hàm là sự chia nhỏ của chương trình.

I.2. Ví dụ

```
//Khai báo thư viện hàm
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<dos.h>
#include<process.h>

//Khai báo biến toàn cục và nguyên mẫu hàm
void ThayThe(char * S, char *St );
void Doc1Sector(int vt);
void Ghi1Sector(int vt);

//Hàm chính
void main()
{
        unsigned char buf[512];
        char S[20], St[20];
        printf("Nhap chuoi can tim: ");
        gets(S);
        printf("Nhap chuoi thay the:");
        gets(St) ;
        printf("\nXin cho…");
        TimVaThayThe(S,St,buf);
        printf("\n Thanh cong.");
        getch();
}

//Cài đặt các hàm con
void ThayThe(char * S, char *St )
{
```

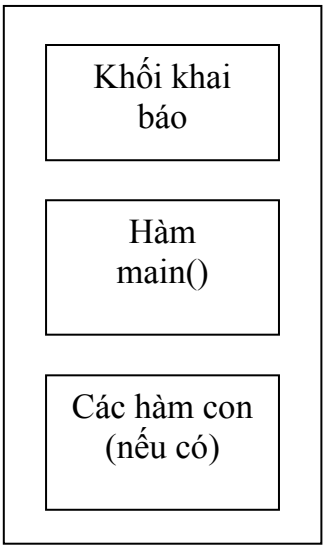
```
        int l=strlen(St);
        for(int i=0;i<l;i++)
                S[i]=St[i];
}

void Doc1Sector(int vt, char buf[512])
{
        if(absread(0,1,vt,buf))
        {
                printf("\n loi doc dia,  nhan enter thoat");
                getch();
                exit(1);
        }
}

void Ghi1Sector(int vt, char buf[512])
{
        if(abswrite(0,1,vt,buf))
        {
                printf("\n loi ghi dia, nhan enter thoat");
                getch();
                exit(1);
        }
}

void TimVaThayThe(char * S, char *St, unsigned char buf[])
{
        for(int i=33;i<=500;i++)
        {
                Doc1Sector(i, buf);
                char * p=strstr(buf, S);
                if(p)
                {
                        ThayThe(p, St);
                        Ghi1Sector(i, buf);
                }
        }
}
```

I.3. Cấu trúc một chương trình C



a. Khối khai báo

Bao gồm các khai báo về sử dụng thư viện, khai báo hằng số, khai báo hàm con (các nguyên mẫu hàm), khai báo các biến toàn cục và khai báo các kiểu dữ liệu tự định nghĩa.

b. Hàm chính (main())

Chứa các biến, các lệnh và các lời gọi hàm cần thiết trong chương trình.

c. Các hàm con

Được sử dụng nhằm mục đích:

- Khi có một công việc giống nhau cần thực hiện ở nhiều vị trí.
- Khi cần chia một chương trình lớn phức tạp thành các đơn thể nhỏ (hàm con) để chương trình được trong sáng, dễ hiểu trong việc xử lý, quản lý việc tính toán và giải quyết vấn đề.

d. Nguyên mẫu hàm

<Kiểu dữ liệu của hàm> Tên hàm ([danh sách các tham số]);

Nguyên mẫu hàm thực chất là dòng đầu của hàm thêm dấu chấm phẩy **(;)** vào cuối, tuy nhiên tham số trong nguyên mẫu hàm có thể bỏ phần tên.

I.4. Cách xây dựng một hàm con

a. Kiểu dữ liệu của hàm

Xác định dựa vào kết quả của bài toán (Output). Gồm 2 loại :

- **void:** Hàm không trả về giá trị. Những hàm loại này thường rơi vào những **nhóm chức năng: Nhập / xuất dữ liệu , thống kê, sắp xếp, liệt kê**.

```
void Tên_hàm (danh sách các tham số)
{
        Khai báo các biến cục bộ
        Các câu lệnh / khối lệnh hay lời gọi đến hàm khác.
}
```

- **Kiểu dữ liệu cơ bản (rời rạc/ liên tục) hay kiểu dữ liệu có cấu trúc:** Kiểu dữ liệu tùy theo mục đích của hàm cần trả về giá trị gì thông qua việc phân tích bài toán. Những hàm loại này thường được sử dụng trong các trường hợp: **Đếm, kiểm tra, tìm kiếm, tính trung bình, tổng, tích**, …

```
<Kiểu dữ liệu> Tên_hàm ([danh sách các tham số])
{
        <Kiểu dữ liệu> kq;
        Khai báo các biến cục bộ
        Các câu lệnh / khối lệnh hay lời gọi đến hàm khác.

        return kq;
}
```

✍ *Đối với những hàm trả về nhiều loại giá trị cho từng trường hợp cụ thể (chẳng hạn như kiểm tra: đúng hay sai, so sánh: bằng , lớn hơn hay nhỏ hơn, ...) thì cần ghi chú rõ giá trị trả về là gì cho từng trường hợp đó.*

b. Tham số

Xác định dựa vào dữ liệu đầu vào của bài toán (Input). Gồm 2 loại :

- **Tham số không là con trỏ (tham trị):** Không thay đổi hoặc không cần lấy giá trị mới của tham số sau lời gọi hàm. Tham số dạng này chỉ mang ý nghĩa là **dữ liệu đầu vào**.
- **Tham số con trỏ (tham biến):** Có sự thay đổi giá trị của tham số trong quá trình thực hiện và cần lấy lại giá trị đó sau khi ra khỏi hàm. Ứng dụng của tham số loại này có thể là **dữ liệu đầu ra** (kết quả) hoặc cũng có thể **vừa là dữ liệu đầu vào vừa là dữ liệu đầu ra**.

c. *Tên hàm*

Đặt tên theo **quy ước đặt tên trong C** sao cho tên gọi **đúng với chức năng hay mục đích thực hiện của hàm và gợi nhớ**.

d. *Ví dụ*

Ví dụ 1: Viết chương trình nhập số nguyên dương n và in ra màn hình các ước số của n

Phân tích bài toán:

- **Input:** n (Để xác định tham số)
  - Kiểu dữ liệu: số nguyên dương (*unsigned int*).
  - Giá trị n không bị thay đổi trong quá trình tìm ước số → Tham số của hàm *không là con trỏ*.
- **Output:** In ra các ước số của n (Để xác định kiểu dữ liệu hàm)
  - Không trả về giá trị.
  - Kiểu dữ liệu của hàm là *void* .
- **Xác định tên hàm:** Hàm này dùng in ra các ước số của n nên có thể đặt là *LietKeUocSo*

Ta có nguyên mẫu hàm:

void LietKeUocSo (unsigned int n);

```
#include<conio.h>
#include<stdio.h>

//Khai bao nguyen mau ham
void LietKeUocSo ( unsigned int n );

void main()
{
        unsigned int n;

        printf("Nhap n = ");
        scanf("%u",&n);
        printf("Cac uoc so cua n : " );
        LietKeUocSo(n);
        getch( );
}

void LietKeUocSo (unsigned int n)
{
```

```
        for(int i=1; i<=n; i++)
                printf("%u\t", i);
}
```

✍ **Lưu ý cách gọi hàm**: *Đối với hàm có kiểu dữ liệu hàm là **void** thì khi gọi **không cần phải gán giá trị vào biến**, **ngược lại phải gọi như trong ví dụ 2** (Phải khai báo **tương ứng kiểu** với kiểu dữ liệu hàm sẽ gọi và gán giá trị trả về vào biến đó).*

Ví dụ 2: Viết chương trình nhập số nguyên dương n và tính tổng

$S = 1 + 2 + 3 + \cdots + n$, với n>0

Phân tích bài toán:

- **Input:** n (Để xác định tham số)
  - Kiểu dữ liệu: số nguyên dương (*unsigned int*).
  - Giá trị n không bị thay đổi trong quá trình tính tổng → Tham số của hàm *không là con trỏ*.
- **Output**: Tổng S (Để xác định kiểu dữ liệu hàm)
  - Trả về giá trị của S.
  - S là tổng các số nguyên dương nên S cũng là số nguyên dương → Kiểu trả về của hàm là *unsigned int* (hoặc *unsigned long* cho trường hợp giá trị của tổng lớn hơn 2 bytes).
- **Xác định tên hàm**: Hàm này dùng tính tổng S nên có thể đặt là *TongS.*

Ta có nguyên mẫu hàm:

unsigned long TongS (unsigned int n);

```
#include<conio.h>
#include<stdio.h>

//Khai bao nguyen mau ham
unsigned long TongS ( unsigned int n );

void main()
{
        unsigned int n;
        unsigned long kq;

        printf("Nhap n = ");
```

```
        scanf("%u",&n);
        kq  = TongS ( n );
        printf("Tong can tinh la: %lu ", kq);
        getch( );
}
unsigned long TongS (unsigned int n)
{
        unsigned long S=0;
        int i=1;
        while(i<=n)
        {
                S+=i;
                i++;
        }
        return S;
}
```

II. BÀI TẬP

II.1. Bài tập cơ bản

1. Cài đặt lại tất cả các bài tập ở chương 2 theo phương pháp hàm.
2. Viết chương trình tính diện tích và chu vi của hình chữ nhật với chiều dài và chiều rộng được nhập từ bàn phím.
3. Viết chương trình tính diện tích và chu vi hình tròn với bán kính được nhập từ bàn phím.
4. Nhập số nguyên dương n (n>0). Liệt kê tất cả các số nguyên tố nhỏ hơn n.
5. Nhập số nguyên dương n (n>0). Liệt kê n số chính phương đầu tiên.
6. Nhập số nguyên dương n (n>0). Đếm xem có bao nhiêu số hoàn thiện nhỏ hơn n.
7. Nhập số nguyên dương n (0 <= n< 1000) và in ra cách đọc của n.
   Ví dụ: Nhập *n = 105*. In ra màn hình: *Mot tram le nam*.
8. Viết chương trình tính tiền thuê máy dịch vụ Internet và in ra màn hình kết quả. Với dữ liệu nhập vào là giờ bắt đầu thuê (GBD), giờ kết thúc thuê (GKT), số máy thuê (SoMay).
   - Điều kiện cho dữ liệu nhập: 6<=GBD<GKT<=21. Giờ là số nguyên.
   - Đơn giá: 2500đ cho mỗi giờ máy trước 17:30 và 3000đ cho mỗi giờ máy sau 17:30.
9. Viết chương trình tính tiền lương ngày cho công nhân, cho biết trước giờ vào ca, giờ ra ca của mỗi người.

Giả sử rằng:

- Tiền trả cho mỗi giờ trước 12 giờ là 6000đ và sau 12 giờ là 7500đ.
- Giờ vào ca sớm nhất là 6 giờ sáng và giờ ra ca trễ nhất là 18 giờ *(Giả sử giờ nhập vào nguyên).*

10. Nhập vào 2 số nguyên p, q và tính biểu thức sau:
$(-q/2+(p^3/27+q^2/4)^{1/2})^{1/3} + (-q/2 - (p^3/27+q^2/4)^{1/2})^{1/3}$
11. Nhập vào 3 số thực a, b, c và kiểm tra xem chúng có thành lập thành 3 cạnh của một tam giác hay không? Nếu có hãy tính diện tích, chiều dài mỗi đường cao của tam giác và in kết quả ra màn hình.
    - Công thức tính diện tích s = sqrt(p*(p-a)*(p-b)*(p-c))
    - Công thức tính các đường cao: $h_a = 2s/a$, $h_b=2s/b$, $h_c=2s/c$.

    (Với p là nữa chu vi của tam giác).
12. Nhập vào 6 số thực a, b, c, d, e, f . Giải hệ phương trình sau :

$$\begin{cases} ax+by=c \\ dx+ey=f \end{cases}$$

13. Viết chương trình nhập 2 số nguyên dương a, b. Tìm USCLN và BSCNN của hai số nguyên đó.
14. Viết chương trình tính tổng nghịch đảo của n giai thừa.
15. Cho 2 số nguyên a, b. Viết hàm hoán vị giá trị 2 số trên.
16. **(*)** Viết chương trình nhập số nguyên dương n gồm 5 chữ số, kiểm tra xem các chữ số n có phải là số đối xứng hay không.

    Ví dụ: *Đối xứng: 13531*

    Không đối xứng: 13921
17. Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, đếm xem n có bao nhiêu chữ số chẵn và bao nhiêu chữ số lẻ.
18. Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, đếm xem n có bao nhiêu chữ số là số nguyên tố.
19. Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, tính tổng các ước số dương của n.

    Ví dụ: *Nhập n=6*

    Tổng các ước số từ 1 đến n: 1+2+3+6=12.
20. Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, tìm ước số lẻ lớn nhất của n.

Ví dụ: *Ước số lẻ lớn nhất của 27 là 9.*

21. Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, kiểm tra xem các chữ số của n có toàn lẻ hay toàn chẵn không.
22. **(*)** Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, sắp xếp các chữ số của n theo thứ tự tăng dần.

    Ví dụ: *Nhập n=1536*

    Kết quả sau khi sắp xếp: 1356.

II.2. Bài tập luyện tập và nâng cao

23. Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, sau đó nhập một số nguyên x, tìm vị trí xuất hiện của chữ số có giá trị x trong n.

    Ví dụ: *Nhập n=1526, x=2*

    Kết quả: Chu so 2 o vi tri thu 3.
24. Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, kiểm tra xem các chữ số của n có được sắp thứ tự không.

    Ví dụ: *Nhập n=1569 hoặc n=8521*

    Kết quả: Có thứ tự.
25. Viết chương trình nhập 2 số a, b sao cho: số lớn nhất trong 2 số phải là một số dương và chia hết cho 7. Nếu nhập sai phải yêu cầu nhập lại cho đến khi đúng.
26. Viết chương trình nhập số nguyên dương n gồm k chữ số $(0 < k \le 5)$, tính giá trị trung bình các chữ số chẵn trong n.
27. **(*)** Viết chương trình in ra màn hình ngày/tháng/năm của ngày hiện tại, cho phép sử dụng các phím mũi tên lên, xuống để tăng hoặc giảm một ngày.
28. **(*)** Viết chương trình in ra màn hình giờ:phút:giây hiện tại, cho phép sử dụng các phím mũi tên lên, xuống để tăng hoặc giảm một giây.

III. KẾT LUẬN

- Trước khi xây dựng một hàm ta phải **xác định mục đích của hàm** là dùng để làm gì, trên cơ sở đó, ta mới xác định được các thành phần của hàm và xây dựng nguyên mẫu hàm.
- Mỗi hàm phải **thực hiện một chức năng độc lập** và tách biệt với các hàm khác *(không được lồng nhau)*.

- Đối với hàm có giá trị trả về phải lưu ý kiểu dữ liệu phải tương ứng kiểu dữ liệu cả giá trị trả về và kiểu dữ liệu của biến được gán khi gọi hàm. Trường hợp hàm trả về từ **hai loại giá trị trở lên thì phải có dòng chú thích cho trường hợp tương ứng** để khi gọi hàm biết được kết quả *(chẳng hạn như tìm kiếm, kiểm tra, so sánh, ... giá trị trả về có 2 trường hợp: Có hoặc không có phần tử cần tìm, thỏa điều kiện kiểm tra hay không? Do vậy ta phải quy ước giá trị cho từng trường hợp).*
- Nên đặt tên hàm sao cho **gợi nhớ** được chức năng, đặt tên **theo quy tắc nhất định** để **tránh việc gọi sai tên hàm** do lẫn lộn giữa ký tự hoa và thường, có dấu gạch nối giữa các từ trong hàm hay không?
- Khi gọi hàm phải truyền **đủ tham số**, **đúng kiểu dữ liệu** và **đúng thứ tự** của tham số.