

BỘ GIÁO DỤC VÀ ĐÀO TẠO



HUTECH
ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ TP.HCM



TÀI LIỆU THỰC HÀNH LẬP TRÌNH C CƠ BẢN

Biên soạn:

Thực Hành Lập Trình C Cơ Bản

Ấn bản 2015

MỤC LỤC

MỤC LỤC	1
HƯỚNG DẪN	3
BÀI 1: TỔNG QUAN BORLAND C	5
1.1 MỤC ĐÍCH	5
1.2 NỘI DUNG	5
BÀI 2: CẤU TRÚC RỄ NHÁNH if VÀ if ... else	9
2.1 Cấu trúc rẽ nhánh if	9
2.2 Cấu trúc rẽ nhánh if ... else	11
CÂU HỎI VÀ BÀI TẬP	13
BÀI 3: CẤU TRÚC LỰA CHỌN SWITCH	14
3.1 GIỚI THIỆU CẤU TRÚC SWITCH	14
CÂU HỎI VÀ BÀI TẬP	16
BÀI 4: CẤU TRÚC VÒNG LẶP FOR	17
4.1 Cấu trúc vòng lặp for	17
4.2 Các ví dụ	18
CÂU HỎI VÀ BÀI TẬP	20
BÀI 5: CẤU TRÚC LẶP do ... while	21
5.1 Cú pháp vòng lặp do ... while	21
5.2 Các ví dụ	22
CÂU HỎI VÀ BÀI TẬP	23
BÀI 6: CHƯƠNG TRÌNH CON	24
6.1 ĐỊNH NGHĨA HÀM	24
6.2 KHAI BÁO HÀM	24
6.3 CÁC VÍ DỤ	25
CÂU HỎI VÀ BÀI TẬP	27
BÀI 7: KIỂU DỮ LIỆU MẢNG	29
7.1 VÍ DỤ MỞ ĐẦU	29
7.2 VÍ DỤ MẢNG 1 CHIỀU	30
7.3 CÁCH KHAI BÁO MẢNG 1 CHIỀU	31
CÂU HỎI VÀ BÀI TẬP	33

TÀI LIỆU THAM KHẢO.....	Error! Bookmark not defined.
-------------------------	-------------------------------------

HƯỚNG DẪN

MÔ TẢ MÔN HỌC

Môn học này cung cấp cho sinh viên những kiến thức cơ bản về lập trình thông qua ngôn ngữ lập trình C. Môn học này là nền tảng để tiếp thu hầu hết các môn học khác trong chương trình đào tạo. Mặc khác nắm vững ngôn ngữ C là cơ sở để phát triển các ứng dụng.

Học xong môn này, sinh viên phải nắm được các vấn đề sau:

- Khái niệm về ngôn ngữ lập trình.
- Khái niệm về kiểu dữ liệu.
- Khái niệm về giải thuật.
- Sử dụng lưu đồ để biểu diễn các giải thuật.
- Tổng quan về Ngôn ngữ lập trình C.
- Các kiểu dữ liệu trong C.
- Các lệnh có cấu trúc.
- Cách thiết kế và sử dụng các hàm trong C.
- Một số cấu trúc dữ liệu trong C.

NỘI DUNG MÔN HỌC

- Bài 1:
- Bài 2:
- Bài 3:
- Bài 4:
- Bài 5:

- Bài 6:
- Bài 7:
- Bài 8:

KIẾN THỨC TIỀN ĐỀ

Ngôn ngữ lập trình Pascal.

YÊU CẦU MÔN HỌC

Người học phải dự học đầy đủ các buổi thực hành ở phòng máy và làm bài tập đầy đủ ở nhà.

CÁCH TIẾP NHẬN NỘI DUNG MÔN HỌC

Để học tốt môn này, người học chú ý các nội dung được học trên lớp giờ lý thuyết; tham gia đều đặn và tích cực trên lớp; hiểu các khái niệm, tính chất và ví dụ tại lớp học. Làm tốt các bài thực hành giao viên yêu cầu làm ở phòng máy. Về nhà thực hiện lại các bài thực hành đã học để nắm vững bài học.

PHƯƠNG PHÁP ĐÁNH GIÁ THỰC HÀNH

- Điểm thi thực hành: 30%. Hình thức và nội dung do GV quyết định, phù hợp với quy chế đào tạo và tình hình thực tế tại nơi tổ chức học tập.

BÀI 1: TỔNG QUAN BORLAND C

Sau khi thực hành xong bài này, sinh viên có thể:

- Làm quen với cấu trúc chung của một chương trình C đơn giản.
- Các lệnh nhập, xuất dữ liệu (`printf`, `scanf`).
- Các kiểu dữ liệu chuẩn (`int`, `long`, `char`, `float`, ...).
- Các phép toán và các hàm chuẩn của ngôn ngữ lập trình C.
- Thực hiện viết các chương trình hoàn chỉnh sử dụng các lệnh đơn giản.

1.1 MỤC ĐÍCH

Làm quen với cấu trúc chung của một chương trình C đơn giản và các lệnh nhập xuất dữ liệu (`printf`, `scanf`), các kiểu dữ liệu chuẩn (`int`, `long`, `char`, `float`...), các phép toán và các hàm chuẩn của ngôn ngữ lập trình C.

Thực hiện viết các chương trình hoàn chỉnh sử dụng các lệnh đơn giản và các kiểu dữ liệu chuẩn đó.

1.2 NỘI DUNG

Bài 1: Làm quen với cấu trúc chung của một chương trình C đơn giản:

```
//Khai báo thư viện
```

```
#include<stdio.h> //thư viện chứa lệnh printf() và scanf()
```

```
#include<conio.h> //thư viện chứa lệnh clrscr() và getch() –
```

```
void main()
```

```
{
```



```

clrscr(); //câu lệnh xóa màn hình
//khai báo các biến
...
//các câu lệnh giải quyết các bài toán tương ứng
printf(...); //xuất dữ liệu ra màn hình
scanf(...); //lấy dữ liệu từ bàn phím lưu vào biến
...
getch(); //lệnh này dùng để làm gì?
}

```

- Mở Borland C lên, vào File/New
- Gõ đoạn code sau:

```

#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf("Chao cac ban");
    printf("Chao cac ban\n");
    printf("Chao\n cac ban");
    printf("Chao \t cac ban");
    printf("\n"Chao cac ban\n");
    getch();
}

```

- Nhấn tổ hợp phím Ctrl + F9 phím để biên dịch và chạy chương trình, xem kết quả hiện ra màn hình và rút ra nhận xét

Bài 2: Viết chương trình in lên màn hình một thiệp mời dự sinh nhật có dạng:

THIỆP MỜI

Than moi ban: "Le Loi"

Toi du le sinh nhat cua minh

Vao luc 19h ngay 13/10/2013

Tai: 05/42 Vinh Vien – TP. HCM

Rat mong duoc don tiep!

Ho Le Thu

HD: Áp dụng \t (tab), \n (xuống dòng), \" (in ra dấu ")

Bài 3: Viết chương trình nhập vào bán kính r của một hình tròn. Tính chu vi và diện tích của hình tròn. In các kết quả lên màn hình.

HD:

- Khai báo biến r để lưu trữ bán kính của hình tròn, khai báo hằng số $PI=3.14$

Khai báo biến: Kiểu_ dữ_ liệu Tên_biến; VD: int a;

Khai báo hằng: #define Tên_hằng Giá_trị VD: #define MAX 100

hoặc: const Kiểu_dữ_liệu Tên_hằng=Giá trị;

VD: const int MAX=100;

- Lệnh nhập dữ liệu từ bàn phím:

scanf("chuỗi qui định kiểu dữ liệu nhập vào", địa chỉ của biến);

- Chuỗi qui định kiểu dữ liệu nhập vào:

%d: kiểu số nguyên

%c: kiểu kí tự

%f: kiểu số thực

Bài 4: Viết chương trình nhập vào độ dài 3 cạnh a, b, c của một tam giác. Tính chu vi và diện tích của tam giác theo công thức:

Chu vi $CV = a+b+c$

Diện tích $S = \sqrt{p(p-a)(p-b)(p-c)}$

Trong đó: $p=CV/2$

In các kết quả lên màn hình

Hướng dẫn: Hàm căn bậc hai \sqrt{x} trong thư viện `math.h`

Bài 5: Viết chương trình đảo ngược một số nguyên dương có đúng 3 chữ số.

VD: Nhập vào $n=234 \rightarrow$ In ra: 432

Hướng dẫn: Lần lượt lấy các chữ số

(sử dụng phép chia $/$ và phép chia lấy phần dư $\%$)

và in ra màn hình theo thứ tự:

Chữ số hàng đơn vị

Chữ số hàng chục

Chữ số hàng trăm

BÀI 2: CẤU TRÚC RỄ NHÁNH if VÀ if ... else

Sau khi học xong bài này, học viên có thể:

- *Làm quen và nắm vững các cấu trúc rẽ nhánh if và if ... else*

2.1 Cấu trúc rẽ nhánh if

Cấu trúc lệnh if

if (biểu thức điều kiện)

{

 <công việc>;

}

Nếu biểu thức điều kiện cho kết quả là 1 tức là đúng thì công việc sẽ được thực hiện. Ngược lại, biểu thức điều kiện cho kết quả là 0 tức là sai thì công việc sau if sẽ được bỏ qua và thực hiện câu lệnh tiếp theo.

Ví dụ 1: Viết chương trình thực hiện:

- Nhập 2 số từ bàn phím, tìm số lớn nhất trong 2 số đó, in kết quả lên màn hình.

```

1 #include <stdio.h>
2 #include <conio.h>
3 void main ()
4 {
5     clrscr();
6     int a,b;
7     printf ("Moi ban nhap gia tri cua A = ");
8     scanf ("%d",&a);
9     printf ("Moi ban nhap gia tri cua B = ");
10    scanf ("%d",&b);
11    if (a > b)
12        printf ("So lon nhat la: %d\n",a);
13    if (a < b)
14        printf ("So lon nhat la: %d\n",b);
15    if (a == b)
16        printf ("Hai so bang nhau\n");
17    getch();
18 }

```

- Nhập 3 số từ bàn phím, tìm số lớn nhất trong 3 số đó, in kết quả lên màn hình

```

1 #include <stdio.h>
2 #include <conio.h>
3 void main ()
4 {
5     clrscr();
6     int a,b,c;
7     int max; //bien max luu gia tri lon nhat
8     printf ("Moi ban nhap gia tri cua A = ");
9     scanf ("%d",&a);
10    printf ("Moi ban nhap gia tri cua B = ");
11    scanf ("%d",&b);
12    printf ("Moi ban nhap gia tri cua B = ");
13    scanf ("%d",&c);
14    max = a; //mac dinh gia tri lon nhat la a.
15    if (max < b) //max so sanh voi b
16        max = b;
17    if (max < c) //max so sanh voi c
18        max = c;
19    printf ("So lon nhat la: %d\n",max);
20    getch();
21 }

```

2.2 Cấu trúc rẽ nhánh if ... else ...

Cấu trúc lệnh if ... else

if (biểu thức điều kiện)

{

 <công việc>;

}

else

{

 <công việc>;

}

Nếu biểu thức điều kiện có kết quả là 1 tức là đúng thì công việc trước else sẽ được thực hiện. Ngược lại, biểu thức điều kiện có kết quả là 0 tức là sai thì công việc sau else sẽ được thực hiện. Nếu công việc cần thực hiện trước và sau else chỉ có một câu lệnh duy nhất thì không cần cặp dấu ngoặc {}.

Ví dụ: Viết chương trình giải phương trình bậc 2 $ax^2 + bx + c = 0$ với a, b, c là 3 số không âm nhập từ bàn phím.

```
1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     clrscr();
6     float a,b,c,delta,x1,x2;
7     printf ("Chương trình giải phương trình bậc 2\n");
8     printf ("Nhập vào các giá trị sau:\n");
9     printf ("A = ");
10    scanf ("%f",&a);
11    printf ("B = ");
12    scanf ("%f",&b);
13    printf ("C = ");
14    scanf ("%f",&c);
15    delta = b*b-4*a*c;
16    if (delta < 0)
17        printf ("Phương trình vô nghiệm\n");
18    else
19        if (delta == 0)
20        {
21            x1 = -b/(2*a);
22            printf ("Phương trình có nghiệm kép x1 = x2 = %0.2f",x1);
23        }
24        else
25        {
26            x1 = ((-b)+sqrt(delta))/(2*a);
27            x2 = ((-b)-sqrt(delta))/(2*a);
28            printf ("Phương trình có hai nghiệm x1 = %0.2f và x2 = %0.2f\n",x1,x2);
29        }
30    getch();
31 }
```

CÂU HỎI VÀ BÀI TẬP

Bài 1. Viết chương trình giải phương trình bậc nhất $ax + b = 0$ với a, b nhập từ bàn phím.

Hướng dẫn:

- Nếu a, b khai báo kiểu số nguyên thì khi tính nghiệm $x = -b/a$ bạn phải thực hiện thao tác ép kiểu cho biến a hoặc biến b sang kiểu số thực.

Vd: $a = 4, b = 10$

`float x = (float)-b/a;`

Bài 2. Nhập vào 3 số nguyên dương a, b, c . Kiểm tra xem 3 số đó có lập thành 3 cạnh của tam giác không? Nếu có hãy cho biết tam giác đó thuộc loại nào (cân, vuông, đều)

Hướng dẫn:

- a, b, c là số nguyên dương. Nên ta khai báo biến thuộc kiểu unsigned int. Chuỗi định dạng kiểu này là %u.
- Điều kiện để 3 số tạo thành tam giác là phải thỏa bất đẳng thức trong tam giác, độ dài 2 cạnh phải lớn hơn cạnh còn lại.
- Loại tam giác: tam giác cân ($a = b$ hoặc $b = c$ hoặc $a = c$), tam giác đều ($a = b = c$); tam giác vuông ($a^2 = b^2 + c^2$ hoặc $b^2 = a^2 + c^2$ hoặc $c^2 = a^2 + b^2$)

Bài 3. Viết chương trình nhập vào số nguyên dương n . kiểm tra xem n có phải là số chính phương hay không ?

Hướng dẫn: số chính phương là số khi lấy căn bậc 2 có kết quả là số nguyên. Tức là

`if (sqrt(n) == n)`

`printf ("%n la so chinh phuong\n",n);`

else

`printf ("%n khong phai la so chinh phuong\n",n);`

BÀI 3: CẤU TRÚC LỰA CHỌN SWITCH

Sau khi học xong bài này, học viên có thể:

- *Làm quen và nắm vững cấu trúc lựa chọn switch*

3.1 GIỚI THIỆU CẤU TRÚC SWITCH

switch (biểu thức)

```
{  
    case giá_trị_1:  
        //các câu lệnh  
        break; //lệnh break giúp chương trình thoát khỏi lệnh switch sau khi  
                // thực hiện xong một trường hợp  
    ...  
    case gia_tri_n:  
        //các câu lệnh  
        break;  
    [default: //các câu lệnh]  
}
```

Ví dụ. viết chương trình nhập số có hai chữ số, hiển thị cách đọc số đó.

```
1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     clrscr();
6     int n;
7     printf ("Moi ban nhap so co hai chu so: ");
8     scanf ("%d",&n);
9     if (n < 10 || n > 99)
10         printf ("So nhap vao khong phai la so co hai chu so\n");
11     else
12     {
13         switch (n/10)
14         {
15             case 1: printf ("Muoi "); break;
16             case 2: printf ("Hai muoi "); break;
17             case 3: printf ("Ba muoi "); break;
18             case 4: printf ("Bon muoi "); break;
19             case 5: printf ("Nam muoi "); break;
20             case 6: printf ("Sau muoi "); break;
21             case 7: printf ("Bay muoi "); break;
22             case 8: printf ("Tam muoi "); break;
23             case 9: printf ("Chin muoi "); break;
24         }
25         switch (n%10)
26         {
27             case 0: printf ("\n"); break;
28             case 1: printf ("mot\n"); break;
29             case 2: printf ("hai\n"); break;
30             case 3: printf ("ba\n"); break;
31             case 4: printf ("bo\n"); break;
32             case 5: printf ("nam\n"); break;
33             case 6: printf ("sau\n"); break;
34             case 7: printf ("bay\n"); break;
35             case 8: printf ("tam\n"); break;
36             case 9: printf ("chin\n"); break;
37         }
38     }
39     getch();
40 }
```

CÂU HỎI VÀ BÀI TẬP

Bài 1: viết chương trình nhập số có ba chữ số, hiển thị cách đọc số đó.

Bài 2: viết chương trình nhập vào tháng của một năm, cho biết số ngày của tháng đó. Nếu tháng nhập vào <1 hoặc >12 thì thông báo “không tồn tại tháng này”

BÀI 4: CẤU TRÚC VÒNG LẶP FOR

Sau khi học xong bài này, học viên có thể:

- Làm quen và nắm vững cấu trúc lặp for.

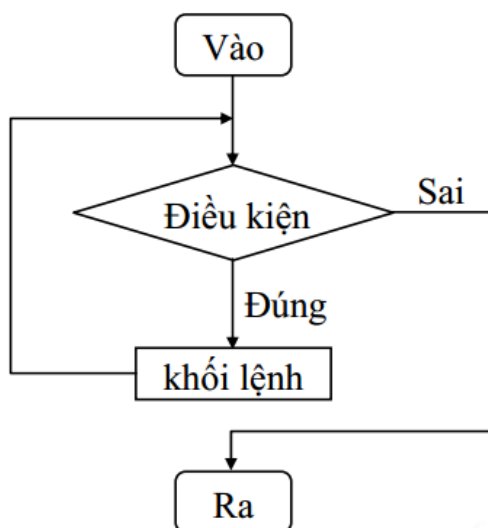
4.1 Cấu trúc vòng lặp for

- Cú pháp lệnh

for (biểu thức 1; biểu thức 2; biểu thức 3)

khối lệnh;

- Từ khóa for phải viết bằng chữ thường
- Nếu khối lệnh bao gồm từ 2 lệnh trở lên thì phải đặt trong dấu { }



- Biểu thức 1: khởi tạo giá trị ban đầu cho biến điều khiển.
- Biểu thức 2: là quan hệ logic thể hiện điều kiện tiếp tục vòng lặp.
- Biểu thức 3: phép gán dùng thay đổi giá trị biến điều khiển.

4.2 Các ví dụ

Ví dụ 1: Viết chương trình hiện ra màn hình 10 dòng: "DAI HOC CONG NGHE – HUTECH".

```
1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     clrscr();
6     int i;
7     for (i = 0; i < 10; i++)
8         printf ("DAI HOC CONG NGHE TP.HCM - HUTECH\n");
9     getch();
10 }
```

Ví dụ 2: Viết chương trình hiện ra các số chẵn trong phạm vi từ 1 đến n, với n nhập từ bàn phím.

```
1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     clrscr();
6     int i,n;
7     printf ("Moi ban nhap vao so n = ");
8     scanf ("%d", &n);
9     for (i = 1; i < n; i++)
10         if (i%2==0) printf ("%d\t",i);
11     getch();
12 }
```

Ví dụ 3: viết chương trình nhập vào số dòng n. Và xuất ra hình sau:

Hiển thị trên n dòng, dòng thứ i có i dấu *

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
```

```
1 #include <stdio.h>
2 #include <conio.h>
3 int main()
4 {
5     clrscr();
6     int i,j, n;
7     printf ("Moi ban nhap vao so dong n = ");
8     scanf ("%d",&n);
9     for (i = 1; i <= n; i++)
10    {
11        for (j = 1; j <= i; j++)
12        {
13            printf ("*");
14            if (i == j) printf ("\n");
15        }
16    }
17    getch();
18 }
```

CÂU HỎI VÀ BÀI TẬP

Câu 1: Viết chương trình hiện ra các số lẻ trong phạm vi từ 1 đến n , với n nhập từ bàn phím.

Câu 2: Viết chương trình tính tổng các số từ 1 đến n . n được nhập từ bàn phím.

Câu 3: Viết chương trình tính tổng các giá trị lẻ trong phạm vi từ 1 đến n , với n được nhập từ bàn phím.

Câu 4: Viết chương trình tính giá trị của $n!$ với n được nhập từ bàn phím.

Câu 5: Viết chương trình hiển thị bảng cửu chương n ra màn hình, với n được nhập từ bàn phím, điều kiện $n > 0$ và $n < 10$.

Câu 6: Có 2 phương thức gửi tiết kiệm:

- Gói không thời hạn lãi suất 2.4%/tháng, mỗi tháng tính lãi 1 lần.
- Gói kỳ hạn 3 tháng lãi suất 4%/tháng, 3 tháng tính lãi một lần

Viết chương trình tính ca vốn lẫn lãi sau một thời gian gửi.

BÀI 5: CẤU TRÚC LẶP

do ... while

Sau khi học xong bài này, học viên có thể:

- *Nắm được hoạt động vòng lặp không xác định số lần.*
- *Nắm được cách sử dụng vòng lặp do ... while.*
- *Vận dụng các câu lệnh có cấu trúc để làm các bài tập thực hành.*

5.1 Cú pháp vòng lặp do ... while

do

{

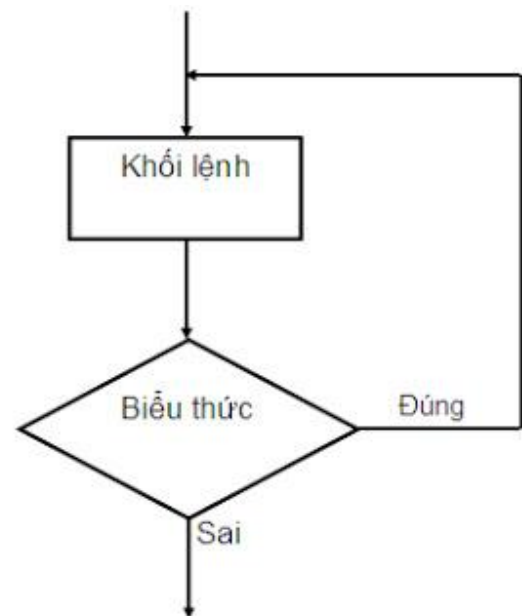
 //<công việc>

} while (biểu thức logic);

Giải thích cú pháp:

Bước 1: thực hiện khối lệnh sau **do**.

Bước 2: kiểm tra giá trị biểu thức logic sau **while**, nếu có giá trị **đúng** thì lặp lại bước 1, nếu có giá trị **sai** tức là 0 thì kết thúc vòng lặp.



5.2 Các ví dụ

Ví dụ 1: Nhập vào 1 số nguyên dương ($n \geq 0$) nếu không phải thì yêu cầu nhập lại.

```

1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     clrscr();
6     int n;
7     do
8     {
9         printf ("Nhap vao so nguyen duong n = ");
10        scanf ("%d",&n);
11        if (n < 0) printf ("Ban da nhap sai, yeu cau nhap lai!\n");
12    } while (n < 0);
13    printf ("Ban da nhap dung yeu cau\n");
14    getch();
15 }
```

Ví dụ 2: Tính tổng giá trị từ 1 \rightarrow n, với n được nhập từ bàn phím với điều kiện là ($10 < n < 20$). Nếu n không thỏa điều kiện thì yêu cầu nhập lại giá trị của n.

```

1 #include <stdio.h>
2 #include <conio.h>
3 void main()
4 {
5     clrscr();
6     int i, n, tong = 0;
7     do
8     {
9         printf ("Nhap vao so nguyen duong (10 < n < 20) n = ");
10        scanf ("%d",&n);
11        if (n <= 10 || n >= 20) printf ("Ban da nhap sai, yeu cau nhap lai!\n");
12    } while (n <= 10 || n >= 20);
13    for (i = 1; i < n; i++)
14        tong = tong + i;
15    printf ("Tong tu 1 den %d la %d\n",n,tong);
16    getch();
17 }
```

CÂU HỎI VÀ BÀI TẬP

Câu 1: Viết chương trình nhập vào một số nguyên, $0 \leq n < 100$. Nếu n không thỏa điều kiện thì yêu cầu nhập lại.

- Kiểm tra n có phải là số nguyên tố hay không ?
- Kiểm n có phải là số chính phương hay không ?
- Kiểm tra n có phải là số hoàn thiện hay không ?

Hướng dẫn:

- Số n được gọi là số nguyên tố khi và chỉ khi n chỉ chia hết cho 1 và chính nó.
- Số n được gọi là số chính phương khi và chỉ khi căn bậc hai của n là số nguyên.
- Số n là số hoàn thiện khi và chỉ khi tổng các ước bằng hai lần chính nó.

Câu 2: Viết chương trình nhập vào số n với điều kiện là $0 \leq n < 100$. Nếu n không thỏa điều kiện thì yêu cầu nhập lại.

- Hiển thị các số nguyên tố trong phạm vi từ 1 đến n .
- Hiển thị các số chính phương trong phạm vi từ 1 đến n .
- Hiển thị các số hoàn thiện trong phạm vi từ 1 đến n .

Câu 3: Viết chương trình nhập vào số n , tính giá trị của $n!$ với điều kiện $0 \leq n < 10$. Nếu n nhập sai, thì yêu cầu nhập lại.

BÀI 6: CHƯƠNG TRÌNH CON

Sau khi học xong bài này, học viên có thể:

- *Nắm vững cách định nghĩa các hàm và cách dùng chúng. Kết hợp các phần đã học trong các chương trình trước để viết các chương trình con.*

6.1 ĐỊNH NGHĨA HÀM

Một chương trình viết trong ngôn ngữ C là một dãy các hàm, trong đó có một hàm chính (hàm main()). Hàm chia các bài toán lớn thành các công việc nhỏ hơn, giúp thực hiện những công việc lặp lại nào đó một cách nhanh chóng mà không phải viết lại đoạn chương trình. Thứ tự các hàm trong chương trình là bất kỳ, song chương trình bao giờ cũng đi thực hiện từ hàm main(). Trong C chương trình con chỉ tồn tại dưới dạng hàm chứ không có thủ tục.

Hàm có thể xem là một đơn vị độc lập của chương trình. Các hàm có vai trò ngang nhau, vì vậy không có phép xây dựng một hàm bên trong các hàm khác.

6.2 KHAI BÁO HÀM

Xây dựng hàm gồm: khai báo kiểu hàm, đặt tên hàm, khai báo các đối và đưa ra câu lệnh cần thiết để thực hiện yêu cầu đề ra cho hàm. Hàm được viết theo mẫu sau:

```
<Kiểu_trả_về> <tên_hàm> ( [khai báo các tham số hình thức] )  
{  
    [Khai báo các biến cục bộ]  
    [Các câu lệnh]  
    [return[biểu thức];]  
}
```

Giải thích:

- **<Kiểu_trả_về>**: giá trị kiểu dữ liệu của dữ liệu sẽ trả về cho hàm
- **<tên_hàm>**: tên của hàm mà bạn muốn định nghĩa, được đặt theo qui tắc đặt tên của C
- **[khai báo các tham số hình thức]**: các tham số hình thức và kiểu của chúng
- **[Khai báo các biến cục bộ]**: khai báo các biến cục bộ, các biến này chỉ có tác dụng trong nội bộ hàm
- **[return]**: là lệnh thực hiện gán giá trị trả về cho hàm
- **[biểu thức]**: là giá trị trả về cho hàm, có thể là biến, hằng, biểu thức nhưng phải có giá trị xác định và có kiểu dữ liệu là kiểu đã khai báo cho hàm.

Lời gọi hàm

Cú pháp:

tên hàm ([Danh sách các tham số thực]);

Danh sách các tham số thực phải bằng số tham số hình thức và lần lượt chúng có kiểu tương ứng với nhau.

@ Lưu ý:

- Hàm có thể có giá trị trả về hoặc không, giá trị trả về phải cùng kiểu với kiểu trả về đã khai báo hàm. Nếu hàm không có giá trị trả về thì đặt từ khóa void trước tên hàm để báo hiệu là hàm không cần giá trị trả về cho hàm.
- Khi hàm khai báo không có kiểu ở trước nó thì nó được mặc định là kiểu int.
- Không nhất thiết phải khai báo nguyên mẫu hàm. Nhưng nói chung nên có vì nó cho phép chương trình biên dịch phát hiện lỗi khi gọi.
- Nguyên mẫu của hàm thực chất là dòng đầu tiên của hàm thêm vào dấu;. Tuy nhiên, trong nguyên mẫu có thể bỏ tên các tham số hình thức.

6.3 CÁC VÍ DỤ

Ví dụ 1: Hàm tìm giá trị lớn nhất giữa hai giá trị.

```
1 int tim_max(int a, int b)
2 {
3     if (a>=b)
4         return a;
5     else
6         return b;
7 }
```

Ví dụ 2: Hàm in ra dãy số từ 1 đến n.

```

1 void in_day_so (int n)
2 {
3     int i;
4     for (i = 1; i <= n; i++)
5         printf ("%d\t", i);
6 }

```

Ví dụ 3: Viết chương trình nhập vào số phút đổi ra thành giờ : phút. Chương trình sử dụng hàm.

```

1 #include <stdio.h>
2 #include <conio.h>
3 // khai bao prototype
4 void time(int &ig, int &ip); //co the k can ghi tham so hinh thuc
5 void time(int &ig, int &ip) // ham doi phut thanh gio:phut
6 {
7     ig = ip/60;
8     ip %= 60;
9 }
10 void main()
11 {
12     clrscr();
13     int igio, iphut;
14     printf("Nhap vao so phut: ");
15     scanf("%d", &iphut);
16     time(igio, iphut);
17     printf("%02d:%02d\n", igio, iphut);
18     getch();
19 }

```

Giải thích chương trình:

Hàm time có hai tham số hình thức là ig, ip có kiểu int. 2 tham số này có toán tử địa chỉ & đi trước cho biết 2 tham số này là dạng truyền tham biến.

CÂU HỎI VÀ BÀI TẬP

Câu 1: Viết chương trình thực hiện các chức năng sau (dùng hàm):

- a. Nhập vào một số n ($0 < n < 100$).
- b. Kiểm tra n có phải là số nguyên tố không ?
- c. Liệt kê các số nguyên tố trong phạm vi từ 1 đến n .
- d. Tính tổng các số nguyên tố trong phạm vi từ 1 đến n .

Câu 2: viết chương trình cài đặt các hàm sau:

- a. Nhập vào một số nguyên n .
- b. Tính tổng các chữ số của n .
- c. Kiểm tra n có chia hết cho 9 không. (yêu cầu không dùng phép chia lấy dư %, mà vận dụng hàm tính tổng ở trên)

Hướng dẫn: một số chia hết cho 9 khi tổng các chữ số của nó chia hết cho 9.

Câu 3: Viết chương trình nhập vào số kw tiêu thụ, tính tiền điện theo yêu cầu sau:

- Bật 1: cho kWh từ 0 – 50: 1.484 VNĐ
- Bật 2: Cho kWh từ 51 – 100: 1.533 VNĐ
- Bật 3: Cho kWh từ 101 – 200: 1.768 VNĐ
- Bật 4: Cho kWh từ 201 – 300: 2.242 VNĐ
- Bật 5: Cho kWh từ 301 – 400: 2.503 VNĐ
- Bật 6: Cho kWh từ 401 trở lên: 2.587 VNĐ

BÀI 7: KIỂU DỮ LIỆU MẢNG

Sau khi học xong bài này, học viên có thể:

- Làm quen kiểu dữ liệu có cấu trúc trong C – kiểu mảng 1 chiều.
- Thực hiện các bài tập trong phần nội dung bằng cách kết hợp dữ liệu mảng
- Vận dụng các câu lệnh đã học để hoàn thành bài tập.

7.1 VÍ DỤ MỞ ĐẦU

Xét ví dụ nhập 5 số nguyên, xuất ra màn hình 5 số và tính tổng 5 số đó.

```
1 #include <stdio.h>
2 #include <conio.h>
3 int main()
4 {
5     clrscr();
6     int a, b, c, d, e, tong;
7     printf ("A = ");
8     scanf ("%d",&a);
9     printf ("B = ");
10    scanf ("%d",&b);
11    printf ("C = ");
12    scanf ("%d",&c);
13    printf ("D = ");
14    scanf ("%d",&d);
15    printf ("E = ");
16    scanf ("%d",&e);
17    tong = a + b + c + d + e;
18    printf ("Tong cac so tren la %d\n",tong);
19    getch();
20    return 0;
21 }
```


Như ví dụ ta thấy nhận hết 5 số và xuất chúng ra màn hình có vẻ rất vất vả và ta đã phải sử dụng quá nhiều biến, nếu số biến của chúng ta không phải 5 mà là 50, 500, ... thì việc này quả là rất khó khăn. Để khắc phục điều này chúng ta hãy sử dụng kiểu mảng.

Mảng là tập hợp các phần tử có cùng kiểu dữ liệu. Mảng có mảng 1 chiều, mảng 2 chiều, ... và mỗi kiểu dữ liệu thì có 1 kiểu mảng tương ứng (mảng nguyên, mảng thực, mảng ký tự), ... Ta chủ yếu xét về mảng 1 chiều và mảng 2 chiều.

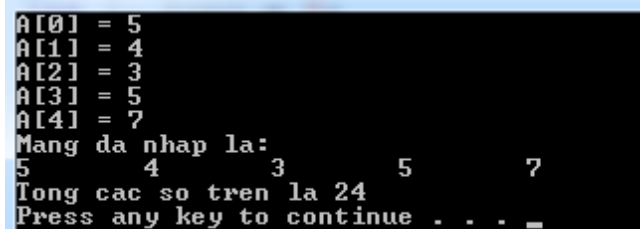
7.2 VÍ DỤ MẢNG 1 CHIỀU

Ví dụ: các bạn đọc và chạy chương trình

```

1 #include <stdio.h>
2 #include <conio.h>
3 int main()
4 {
5     clrscr();
6     int i, tong = 0;
7     int n = 5, a[n]; // khai bao mang a co 5 phan tu.
8     for (i = 0; i < n; i++) //thuc hien nhap gia tri cho mang a
9     {
10         printf ("A[%d] = ", i);
11         scanf ("%d", &a[i]);
12     }
13
14     printf ("Mang da nhap la:\n");
15     for (i = 0; i < n; i++) //xuat gia tri mang a
16         printf ("%d\t", a[i]);
17
18     for (i = 0; i < n; i++) //tinh tong cac gia tri trong mang
19         tong = tong + a[i];
20     printf ("\nTong cac so tren la %d\n", tong);
21     getch();
22     return 0;
23 }
```

Kết quả chạy chương trình:



```

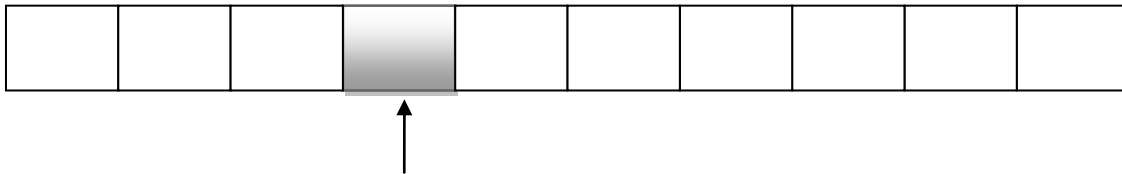
A[0] = 5
A[1] = 4
A[2] = 3
A[3] = 5
A[4] = 7
Mang da nhap la:
5      4      3      5      7
Tong cac so tren la 24
Press any key to continue . . . _
```

7.3 CÁCH KHAI BÁO MẢNG 1 CHIỀU

KHAI BÁO:

<Kiểu dữ liệu> <tên mảng> <[số phần tử tối đa trong mảng]>;

Ví dụ: `int a[10];` mảng 1 chiều a gồm 10 phần tử thuộc kiểu số nguyên. Sau khi khai báo ta có 1 mảng có dạng như sau:

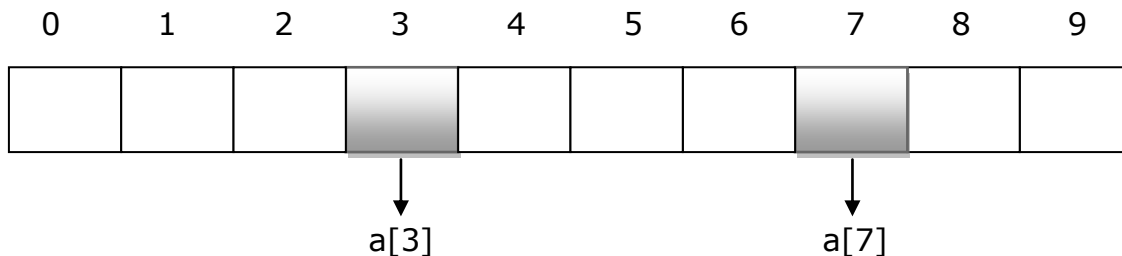


Mỗi phần tử trong mảng có kiểu `int`

TRUY XUẤT:

Sau khi mảng được khai báo, mỗi phần tử trong mảng đều có chỉ số để tham chiếu. chỉ số bắt đầu từ số 0 đến $n-1$ (với n là kích thước mảng). trong ví dụ trên, ta khai báo mảng 10 phần tử thì chỉ số từ 0 đến 9.

Và ta truy xuất qua cú pháp: `<tên mảng> <chỉ số>`



NHẬP XUẤT PHẦN TỬ CỦA MẢNG:

Để nhập dữ liệu cho các phần tử trong mảng ta cần duyệt tới từng phần tử trong mảng và tiến hành nhập bằng một vòng lặp `for`.

```
for (i = 0; i < n; i++) //thuc hien nhap gia tri cho mang a
{
    printf ("A[%d] = ", i);
    scanf ("%d", &a[i]);
}
```

Viết xuất các phần tử cũng được tiến hành tương tự:

```
for (i = 0; i < n; i++) //xuat gia tri mang a
    printf ("%d\t", a[i]);
```

NHẬP XUẤT MẢNG SỬ DỤNG HÀM:

Trong nhiều bài toán, chúng ta phải nhập nhiều mảng và xuất nhiều lần, khi đó hãy đưa các lệnh nhập, xuất vào hàm cho dễ dùng.

```
1 #include <stdio.h>
2 void nhapmang (int n, int a[])
3 {
4     int i;
5     for (i = 0; i < n; i++)
6     {
7         printf ("A[%d] = ", i);
8         scanf ("%d", &a[i]);
9     }
10 }
11 void xuatmang (int n, int a[])
12 {
13     int i;
14     for (i = 0; i < n; i++)
15         printf ("%d\t", a[i]);
16 }
17 int main()
18 {
19     clrscr();
20     int n = 5;
21     int a[n];
22     nhapmang (n, a);
23     xuatmang (n, a);
24     getch();
25     return 0;
26 }
```

Lưu ý: Một mảng có mối quan hệ với con trỏ, mảng có thể coi như con trỏ nên khi truyền mảng vào hàm thì các phần tử của mảng sẽ bị ảnh hưởng nếu có sự thay đổi nào trong mảng.

CÂU HỎI VÀ BÀI TẬP

Câu 1: Viết chương trình thực hiện:

- a. Nhập mảng số nguyên gồm n phần tử ($0 < n \leq 10$)
- b. Xuất mảng vừa nhập

Yêu cầu: Xuất mảng vừa nhập sau:

Mảng vừa nhập là:

0 4 7 8 9 0 7

- c. Liệt kê các phần tử chia hết cho 3 có trong mảng (nếu không có thì xuất ra màn hình thông báo "Mảng không có phần tử nào chia hết cho 3").
- d. Tính tổng các số trong mảng.
- e. Tính tích các số trong mảng.
- f. Tính trung bình cộng của mảng.
- g. Tính trung bình cộng các phần tử dương của mảng.
- h. Tính trung bình cộng các phần tử âm của mảng.
- i. Liệt kê các phần tử là số nguyên tố trong mảng.
- j. Liệt kê các phần tử là số chính phương trong mảng.
- k. Liệt kê các phần tử là số hoàn thiện trong mảng.
- l. Tìm phần tử lớn nhất trong mảng.
- m. Tìm phần tử nhỏ nhất trong mảng.

Câu 2: Làm các câu a, b, d đối với mảng 1 chiều các số thực.