Phần 1: Ngôn ngữ lập trình C

Chương 1: Ôn tập một số nội dung chính của NNLT C

Các nội dung chính

- Các đặc điểm nổi bật của C
- Cấu trúc của một chương trình viết bằng C
- Các kiểu dữ liệu cơ bản
- Các thao tác nhập/xuất cơ bản
- Các kiểu dữ liệu có cấu trúc
- Kiểu dữ liệu con trỏ
- Các cấu trúc điều khiển lệnh

Một số đặc điểm chính của C

- C là một ngôn ngữ lập trình bậc trung (kết hợp giữa bậc cao và bậc thấp)
- C là ngôn ngữ có cấu trúc
- C là một ngôn ngữ mạnh và mềm dẻo, có thể viết các chương trình ở nhiều mức khác nhau từ thấp đến cao.
- C tạo ra các chương trình hiệu quả
- C là một ngôn ngữ khả chuyển (portable).
- C có kích thước nhỏ
- C định kiểu không cứng nhắc (loose typing).

Cấu trúc của một chương trình

```
#include <filename.h> /* Chỉ thị gọi tệp thư viện/tiền xử lý */

Kiểu_hàm  main ()      /* Hàm main, thân chương trình chính */
{
   /* Các câu lệnh của thân chương trình chính */
   …
   [return (giá_trị);]  /* trả về giá trị cho hàm main*/
}

/*Định nghĩa các hàm con, là các chương trình con */
Kiểu_hàm   Tên_hàm (các tham số)
{
   /* Các lệnh của hàm */
   …
   [return (giá_trị)]  /* trả về giá trị cho hàm */
}
```

Các thành phần cơ bản của một chương trình

- Các đối tượng dữ liệu: cần được khai báo và gồm có tên và kiểu dữ liệu, và được chia thành 2 loại
  - Biến (variable)
  - Hằng (constant)
- Các lệnh xử lý (statement): được chia làm 2 nhóm:
  - Lệnh đơn: lệnh gán, lệnh khai báo dữ liệu,…
  - Lệnh có cấu trúc: gồm 3 cấu trúc điều khiển

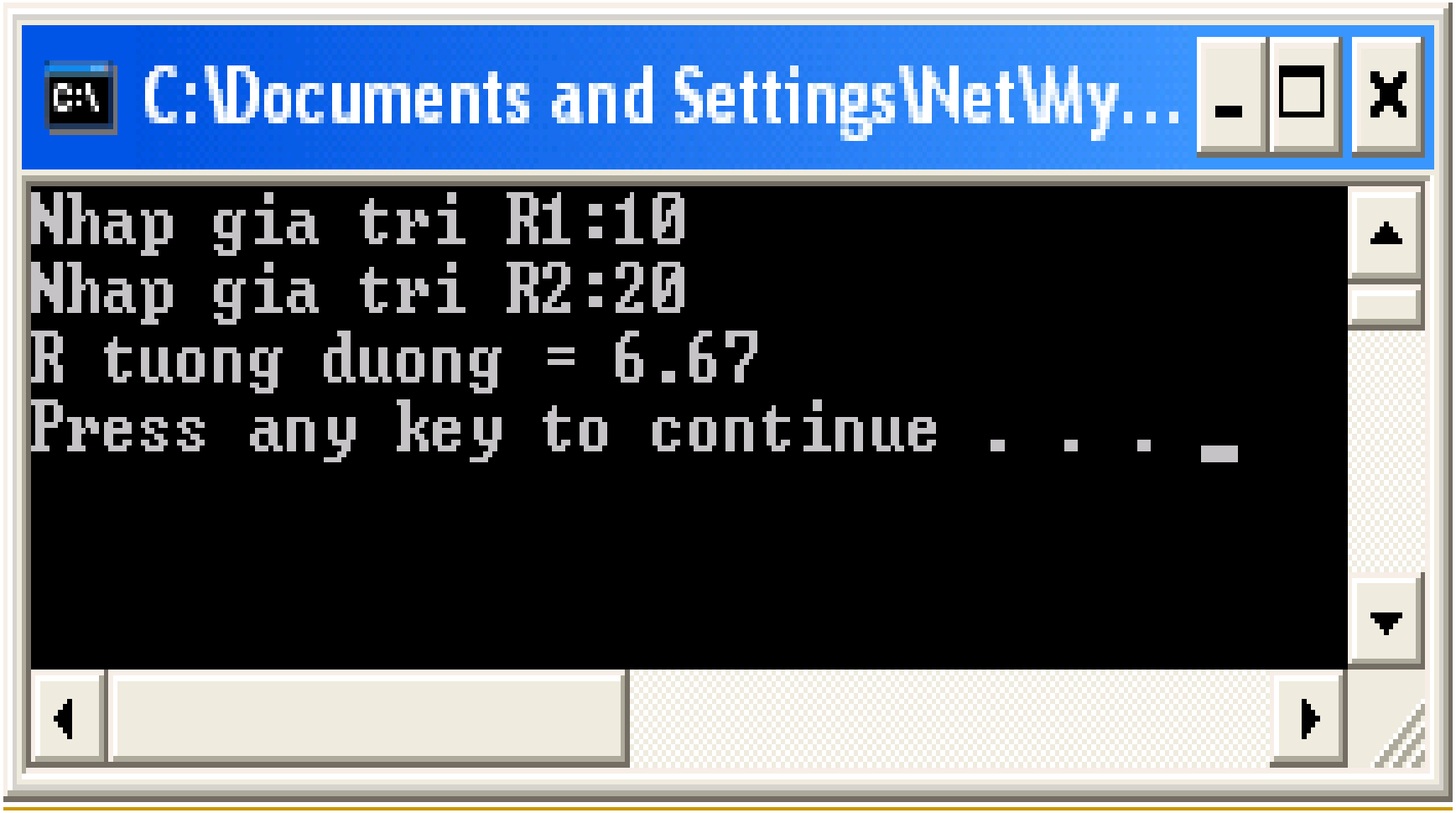
Một số chương trình mẫu

- **Program 1.1**: Viết một chương trình tính điện trở tương đương của 2 điện trở R1 và R2 mắc song song. Giá trị R1 và R2 được nhập từ bàn phím. (Lưu ý: giá trị nhập vào phải hợp lệ)
- **Program 1.2**: Mở rộng Program 1.1, với tính năng kiểm tra tính hợp lệ của giá trị điện trở nhập vào.

Program 1.1: Chương trình tính điện trở tương đương của hai điện trở R1 và R2

```
#include <stdio.h>  //Khai báo các tệp thư viện
#include <stdlib.h>

void main()
{
        float R1, R2;  //Các biến lưu 2 điện trở R1 và R2
        float R;        //Biến lưu điện trở tương đương

        printf("Nhap gia tri R1:");
        scanf("%f",&R1);
        printf("Nhap gia tri R2:");
        scanf("%f",&R2);

        R = 1/(1/R1+1/R2);  //Tính điện trở tương đương và in ra kết quả
        printf("R tuong duong = %.2f\n",R);

        system("PAUSE");
}
```

Kết quả chạy Program 1.1



Các kiểu dữ liệu cơ bản

- Kiểu kí tự (**char**)
- Kiểu số nguyên (**int**, **long**, **unsigned**): unsigned là kiểu số nguyên không dấu (không âm). Kiểu mặc định là có dấu (signed)
- Kiểu logic: không có từ khóa khai báo, mà sử dụng luôn kiểu số nguyên để biểu diễn giá trị logic
- Kiểu số thực (**float**, **double**)
- Kiểu chuỗi: gồm một dãy các ký tự nằm trong cặp “”, và kí tự kết thúc chuỗi ‘\0’.

Các thao tác nhập/xuất cơ bản

- Các thao tác nhập dữ liệu
  - Các thao tác nhập kí tự: *getchar()*; *getch()*; *getche()*
  - Nhập chuỗi kí tự: *gets()*
  - Hàm đa năng *scanf()*
- Các thao tác xuất dữ liệu
  - Xuất kí tự: *putchar()*
  - Xuất chuỗi kí tự: *puts()*
  - Hàm đa năng: *printf()*

Các lệnh điều khiển có cấu trúc

- Lệnh điều khiển tuần tự (khối lệnh)
- Lệnh chọn rẽ nhánh: if; if … else; switch… case
- Lệnh lặp: while ; do …while; for

Program 1.2: Mở rộng Program 1.1, với việc kiểm tra tính hợp lệ của giá trị điện trở nhập vào

```
#include <stdio.h>  //Khai báo các tệp thư viện
#include <stdlib.h>
void main()
{
      float R1, R2;
      float R;       //Điện trở tương đương

//nhập và kiểm tra các điện trở đưa vào
…
//Tính và in ra điện trở tương đương
…

}
```

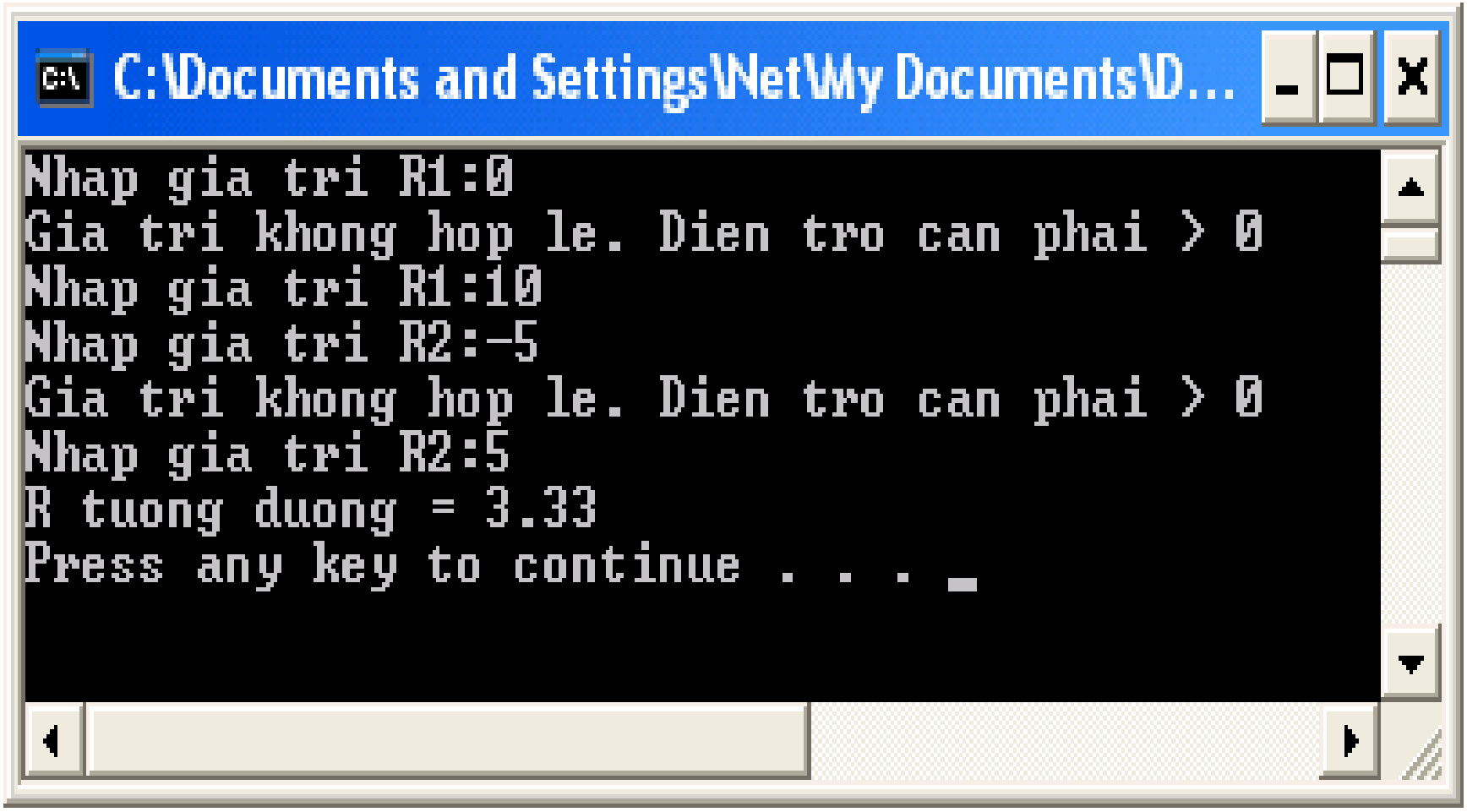
Program 1.2 (tiếp)

```
void main()
{
    //nhập và kiểm tra các điện trở đưa vào
    do {
    printf("Nhap gia tri R1:");
    scanf("%f",&R1);
    if (R1<=0) printf("Gia tri khong hop le. Dien tro can phai > 0\n");
    } while (R1<=0);

    do {
        printf("Nhap gia tri R2:");
        scanf("%f",&R2);
        if (R2<=0) printf("Gia tri khong hop le. Dien tro can phai > 0\n");
    } while (R2<=0);
    //Tính và in ra điện trở tương đương
}
```

Program 1.2 (tiếp và kết thúc)

```
void main()
{
     //Tính và in ra điện trở tương đương
    R = 1/(1/R1+1/R2);
    printf("R tuong duong = %.2f\n",R);

    system("PAUSE");
}
```

Kết quả chạy Program 1.2

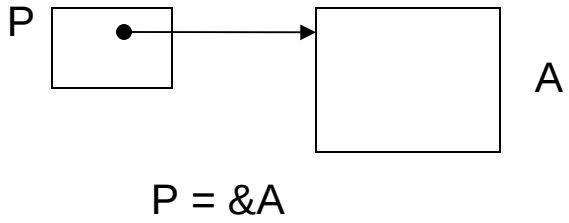


Các kiểu dữ liệu có cấu trúc

- **Kiểu mảng**: chứa các phần tử cùng một kiểu dữ liệu và có kích thước cố định. Mảng có thể là 1 hay nhiều chiều
- **Kiểu struct**: là cấu trúc cho phép chứa các thành phần dữ liệu khác, được gọi là các trường. Các trường có thể có các kiểu dữ liệu khác nhau, và cũng có thể lại là kiểu có cấu trúc
- **Kiểu union**: là một kiểu struct đặc biệt có kích thước lưu trữ bằng với kích thước của trường lớn nhất.
- **Kiểu bit field** (trường bit): là một kiểu struct đặc biệt mà kích thước mỗi trường có thể được xác định theo số bit

Kiểu dữ liệu con trỏ

- Giới thiệu: Là đối tượng DL mà giá trị của nó là địa chỉ của các đối tượng khác (có thể là chính nó) trong bộ nhớ



Kiểu dữ liệu con trỏ

- Vai trò:
  - Quản lý các đối tượng DL động và cấu trúc lưu trữ động (như CTLT móc nối) để cài đặt lưu trữ các CTDL động như danh sách, cây,…
  - Định vị, truy nhập vào các thành phần của các kiểu DL có cấu trúc nhằm tăng tốc độ thực hiện và độ linh hoạt trong xử lý. Ta hay dùng con trỏ để truy nhập vào mảng, bản ghi (struct).
  - Tổ chức các tham số đóng vai trò đầu ra của các chương trình con (hàm con).

Kiểu dữ liệu con trỏ

- Các thao tác cơ bản trên con trỏ:
  - Khai báo
  - Phép lấy địa chỉ
  - Phép gán
  - Truy nhập vào đối tượng được trỏ
  - Phép tăng/giảm địa chỉ

```
int * pi;          //Con trỏ kiểu int
float * pf[20];    //Mảng các con trỏ kiểu float
float (*pf)[20]; //Con trỏ kiểu mảng
void * p;          //Con trỏ tổng quát.

int i=20;
pi = &i;    //gán giá trị cho con trỏ
p = pi;     //gán con trỏ cho con trỏ
*pi = 30; //truy nhập vào đối tượng đc trỏ,
            //i = 30 now
int a[10];
pi = a;     //pi = &a[0]
pi[2]=20;//a[2] = 20
pi++;       //pi = &a[1]
```

Một số chương trình mẫu

- **Program 1.3**: mở rộng Program 1.2 với N điện trở mắc song song
- **Program 1.4**: thay đổi Program 1.3, với việc tính điện trở tương đương được thực hiện bằng một hàm con.

Program 1.3

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

//#define N 5

void NhapDienTro(float *R);

int main()
{
	const int N = 5;
	float R[N];
	float Rtd;		//Dien tro tuong duong
	int i;

	//Nhập các giá trị điện trở và tính điện trở tương đương …
}
```

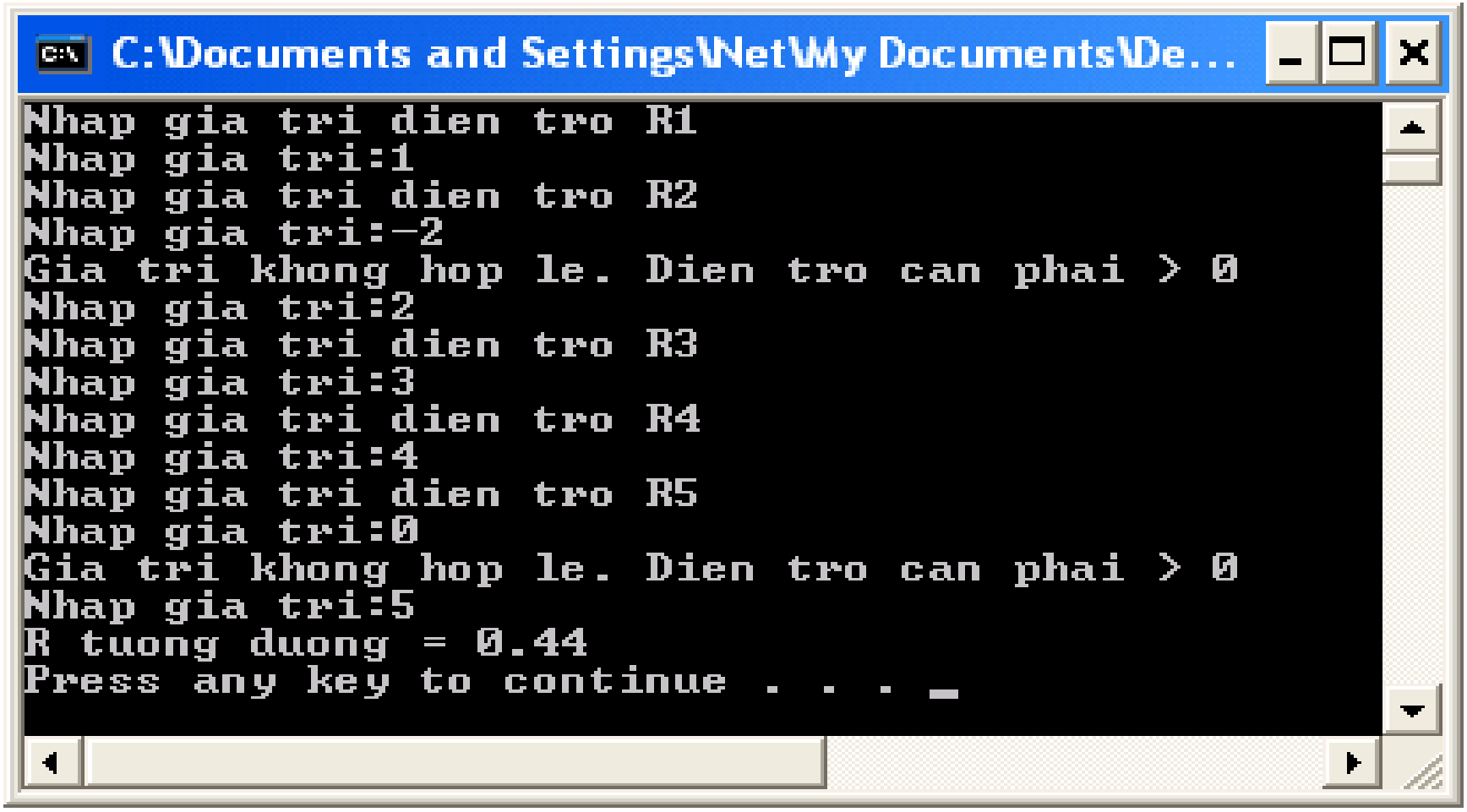
Program 1.3 (tiếp)

```
int main()
{
    //Nhập các giá trị điện trở và tính điện trở tương đương

    for (i=0;i<N;i++){
        printf("Nhap gia tri dien tro R%d\n",i+1);
        NhapDienTro(&R[i]);
    }

    Rtd=0;
    for (i=0;i<N;i++)  Rtd += (1/R[i]);
    Rtd = 1/Rtd;

    printf("R tuong duong = %.2f\n",Rtd);

    system("PAUSE");
    return 0;
}
```

Program 1.3 (tiếp và hết)

```
void NhapDienTro(float *R){
    float R1;
    do {
        printf("Nhap gia tri:");
        scanf("%f",&R1);
        if (R1<=0) {
            printf("Gia tri khong hop le. Dien tro can phai > 0\n");
        }
    } while (R1<=0);

    *R=R1;
}
```

Kết quả chạy Program 1.3



Program 1.4

```
#include <stdio.h>
#include <stdlib.h>
#define N 5
void NhapDienTro(float *R);  //Ham nhap gia tri dien tro va co kiem tra tinh hop le
float TinhDienTroTD(float R[],int n);//Ham tinh DT tuong duong cua n dien tro mac //

int main()
{
    float R[N];
    float Rtd;          //Dien tro tuong duong
    int i;
    //Nhập các giá trị điện trở và tính điện trở tương đương …
}
```

Program 1.4 (tiếp)

```
int main()
{
      //Nhập các giá trị điện trở và tính điện trở tương đương
     for (i=0;i<N;i++){
            printf("Nhap gia tri dien tro R%d\n",i+1);
            NhapDienTro(&R[i]);
     }

     Rtd=TinhDienTroTD(R,N);
     printf("R tuong duong = %.2f\n",Rtd);

     system("PAUSE");
     return 0;
}
```

Program 1.4 (tiếp và hết)

```
float TinhDienTroTD(float R[],int n){
   float Rtg=0;
   int i;
   for (i=0;i<n;i++)
   Rtg += (1/R[i]);
   return 1/Rtg;
}
```

Kết quả chạy Program 1.4

- Tương tự như kết quả chạy của Program 1.3

Câu hỏi và Bài tập

- Câu hỏi:
  1. Vì sao nói C là ngôn ngữ không định kiểu cứng nhắc. Đưa ví dụ minh hoạ.
  2. Đơn vị dữ liệu nhỏ nhất trong C/C++ là gì. Cho ví dụ minh hoạ.
  3. Từ khoá là gì? Các từ khoá trong C có đặc điểm gì ?
  4. Nêu vai trò của lệnh *break* trong cấu trúc lệnh *switch*. Cho ví dụ để so sánh sự khác nhau khi có sử dụng lệnh *break* và khi không sử dụng lệnh *break*.
  5. Ta có thể khai báo mảng có kích thước không xác định (có thể thay đổi) được không, vì sao?
  6. Nêu các thao tác cơ bản trên dữ liệu kiểu con trỏ. Cho ví dụ.
  7. Dữ liệu kiểu con trỏ có thay thế cho dữ liệu kiểu mảng được không, vì sao. Cho vd.
  8. Mảng 2 chiều có thể được định nghĩa thông qua mảng một chiều được không.

Bài tập

- Bài 1: Viết chương trình giải phương trình bậc 2, với các giá trị a, b, c được nhập từ bàn phím, chương trình sẽ in ra các nghiệm. Chương trình chỉ sử dụng hàm main.
- Bài 2: Mở rộng bài 1, với tính năng kiểm tra giá trị a nhập vào phải hợp lệ (khác 0), và có thêm hàm con tính delta.
- Bài 3: Từ bài 2, viết 1 hàm GiaiPTBac2, với đầu vào là 3 hệ số, đầu ra sẽ cho biết PT có mấy nghiệm và giá trị từng nghiệm nếu có.

Xin cảm ơn!