

第 1 章

(1) C 程序的编译过程 (P2)

(2) 算法的特点 (P3)

有穷性、确定性、有效性、有 0 个或多个输入、有 1 个或多个输出

(3) C 程序的流程图 (P3)

(4) C 语言的特点 (P6)

第 2 章

(1) 合法的标识符 (P23)

(2) C 语言的关键字 (P23)

(3) 三种基本数据类型及其取值范围 int char float (P24 表 2-1), 如

unsign u=65535;

int i;

i=u;

printf(“%d”,i); (备注: 1111 1111 1111 1111 就是-1 的补码)

(4) 整型常量: 十进制、八进制、十六进制 (P26)

如: **018 (错误)、0x4af、100、017 等**

(5) 实型常量: 表示方式, 合法的实型常量 (P27)

不分单、双精度

(6) 字符型常量: 用单引号括起来一个字符, 转义字符 (P28 表 2-2)

如: **‘\n’、‘\\’、‘\0’、‘\071’、‘\081’ (错误)、‘\x35’ 等**

大小写字母的转换: 相差 32

(7) 字符串常量: 用双引号括起来的多个字符, 用字符数组表示;

无字符串变量 (P28), 用字符数组或字符指针表示

(8) 符号常量, 如: #define PI 3.14159 (P28)

(9) 语句与表达式的区别: a+b; a+b

(10) 赋值运算: 左值只能是变量, 不能是表达式和常量 (P31)

(11) 自增和自减运算, 区别 a++、++a (P32)

```
a=6; while(--a); printf("%d", a);
```

```
a=6; while(a--); printf("%d", a);
```

(12) 运算符 (P36 表 2-3)

单 双 (算术、关系、逻辑) 三 赋值 逗号

(13) sizeof 运算符与字符串函数 strlen 区别如: "hello"

(14) 各种数据类型间的混合运算 (P36), 如:

```
int i=2;
```

```
float f=3;
```

i/3 与 i/f 进行比较

第 3 章

(1) printf 用法, 非特殊字符原样输出, 特殊字符有 "%", "\"

```
printf("%d,%d\n",a,b);
```

(2) 输出函数 printf 格式控制符 (P52 表 3-1 和 3-2)

%m.nf 与 %m.ns 的用法 (P53)

(3) o、x (或 X) 格式符 (P53)

(4) 输入函数 scanf 用法, 如:

```
scanf("%d,%d",&a,&b);
```

三种错误的形式:

int a;

float f;

1、 scanf("%d",a);

2、 scanf("%d\n",&a);

3、 scanf("%5.3f",&f);

(5) putchar 函数与 getchar 函数用法 (P56)

(6) 交换语句: **t=a;a=b;b=t;** 区别: t=a, a=b, b=t;

第 4 章

(1) 关系运算 (P66): **0<x<2** 用法应写为: **0<x&& x<2**

(2) 逻辑运算中的求值短路 (P66), 如:

1||? 或 0&&?

例如: **(j++||k++) && i++**

(3) 条件运算符, 唯一的三目运算 (P66)

例如: **x>y ? x: y > z ? y : z** 等价于 **x>y ? x: (y > z ? y : z)**

(4) if 语句的用法 (P67), 如:

int a=0;

if(a=1) *;**

else *;**

例如:

int x=10, y=20, z=30;

if(x>y)

z=x; x=y; y=z;

(5) if 语句的嵌套 (P68) else 的匹配原则

(6) switch 用法, 举例说明 (P68-70)

```
switch (a)
case 0: x++;
case 1: y++;
case 2: x++;y++;
```

例如:

```
x=1, y=0, a=0, b=0;
switch(x)
{
    case 1:  switch(y)
              {   case 0: a++;  break;
                  case 1: b++;  break;
              }
    case 2: a++; b++; break;
}
```

```
x=1, y=0, a=0, b=0;
switch(x)
{
    case 1:  switch(y)
              {   case 0: a++;  break;
                  case 1: b++;  break;
              }  break;
    case 2: a++; b++; break;
}
```

第 5 章

(1) while 与 do...while 的区别

(2) while 与 for 用法

表达式 1;	for(表达式 1; 表达式 2; 表达式 3)
while(表达式 2)	循环体;

```
{ 循环体;  
    表达式 3; }
```

(3) break 与 continue 的用法及区别 (P82)

```
for(表达式 1; 表达式 2; 表达式 3)  
{  
    if(条件语句)  
        break;  
    循环体;  
}  
只用于循环和 switch 语句中
```

```
for(表达式 1; 表达式 2; 表达式 3)  
{  
    if(条件语句)  
        continue;  
    循环体;  
}  
只用于循环中
```

在双重循环中的应用:

```
for( i=1; i<10; i++ )  
{  
    for( j=1; j<10; j++ )  
    {  
        if(条件语句)  
            break;  
        循环体 A;  
    }  
    循环体 B;  
}
```

```
for( i=1; i<10; i++ )  
{  
    for( j=1; j<10; j++ )  
    {  
        if(条件语句)  
            continue;  
        循环体 A;  
    }  
    循环体 B;  
}
```

(4) 循环的嵌套 (P84)

(5) 几种常用的算法:

1、累加求和、求积:

方法一: `s=s+i`

方法二: `int fun(int n)`

```
{  int s;  
    if(n==1)  s=1;
```

```
else s=fun(n-1)+n;  
return s; }
```

2、求最大值和最小值

3、判断素数算法

4、判断完数、水仙花数、同构数的方法

5、九九乘法表、金字塔的打印输出

(6) 重点掌握变量跟踪法，读程序结果。

第 6 章

(1) 数组的定义 (P92)，比较下列两种形式：

```
int n=10;                                #define N 10  
int a[n];                                int a[N];  
int a[10]; (只能是常量、整数)
```

(2) 数组元素的初始化 (P94)

例如：int a[10]={1,2,3,4};

```
int a[10]; a={1,2,3,4};
```

区别：int a[10]; 等价于 auto int a[10];

```
static int a[10];
```

(3) 二维数组的定义、引用及初始化 (P97-99)，

二维数组的转置算法 (P100 例 6.6) : b[i][j]=a[j][i];

(4) 字符数组的定义及初始化 (P103)

字符数组因为不一定是字符串，可以没有'\0'

例如：char ch[3]={'h', 'o', 'w'};

所以输入与输出只能用循环，逐个字符的输入与输出。

```
for(i=0; i<3; i++) printf("%c", ch[i]);
```

例如: `char ch[4]={'h', 'o', 'w'};`

此时相等于把“how”字符串对字符数组赋值，用字符串的输出。

```
printf("%s", ch);
```

(5) 字符串的输入 (P105)

```
char str[10];
```

定义数组的同时给出其初值:

```
char ch[3]={'h', 'o', 'w'};
```

```
char ch[ ]={'h', 'o', 'w'};
```

```
char ch[4]={'h', 'o', 'w'};
```

```
char ch[4]="how";
```

```
char ch[3]="how";
```

采用键盘的方式对字符串的输入与输出:

```
scanf("%s", str);
```

 ----以空格和回车结束

```
gets(str);
```

 ----只以回车结束

(6) 字符串的输出 (P105)

```
printf("%s", str);
```

 ----输出字符

```
puts(str);
```

 ----输出字符 + '\n'

(7) 字符串常用函数 (P108), 必须在#include<string.h>

关键点: 起始地址、字符串的结束标志'\0'

第 7 章

(1) 指针的定义及初始化 (P125), 如

int i,*p;

p=&i; 不能写成: *p=i;

(2) 指针与一维数组 (P129), 如:

int a[10], *p, i;

p=a;

***(p+i)⇔p[i]⇔a[i]⇔*(a+i) -----表示元素**

p+i⇔&p[i]⇔&a[i]⇔a+i -----表示地址

特别注意: 变量 p 与数组 a 的区别

例如: 当 flag 为 1 时, 进行由小到大排序, flag=0 时, 由大到小排序。

void f(int b[], int n, int flag)

{

int i, j, t;

for(i=0; i<n-1; i++)

for(j=i+1; j<n; j++)

if(flag?b[i]>b[j]:b[i]<b[j]) { t=b[i]; b[i]=b[j]; b[j]=t; }

}

void main()

{

int a[10]={5,4,3,2,1,6,7,8,9,10}, i;

f(&a[2],5,0); f(a,5,1);

for(i=0; i<10; i++) printf(“%d”, a[i]);

}

(3) 指针与二维数组 (P131)

```
int a[3][4], *p, (*q)[4];
```

```
p=&a[0][0]; p=a[0]; p=*(a+0); 相当于所有的列地址
```

```
p=&a[1][1]; p=a[1]+1; p=*(a+1)+1;
```

```
q=a; q=a+1; 相当于所有的行地址
```

(4) 指针数组 (P135)

```
char *p[]={"123", "12", "12345", "1"}
```

(5) 动态分配内存 (P136), 如

```
void *malloc(unsigned size)
```

```
void *calloc(unsigned n, unsigned size)
```

```
void free(void *ptr)
```

例:

```
int *p;
```

```
p=(int*)malloc(100);
```

```
free(p)
```

例如:

```
amovep(int *p, int(*a)[3], int n)
```

```
{
```

```
int i, j;
```

```
for(i=0; i<n; i++)
```

```
for(j=0; j<n; j++)
```

```
{ *p=a[i][j]; p++; }
```

```

}
void main( )
{
    int *p, a[3][3]={{1,3,5},{2,4,6}};
    p=(int *)malloc(100);
    amovep(p, a, 3);
    printf("%d, %d\n", p[2], p[5]);
    free(p);
}

```

(6) 指针与字符串 (P141)

```

char str[20];
str= "I love China!" ;

```

```

char *cp;
cp= "I love China!" ;

```

```

char str[10];
scanf("%s",str);

```

```

char *cp, str[10];
cp=str;
scanf("%s",cp);

```

如: `char s [] ="ABCD", *p;`
`for (p=s+1; p<s+4; p++)`
`printf ("%s \n",p) ;`

(7) 多级指针 (P144)

第 8 章

(1) 函数定义 (P158) 和函数调用 (P163)

(2) 函数的返回值：以函数的返回值类型为主 (P163)

(3) 函数的参数传递，值传递和地址传递 (P166)

P166 例题 8.6

```
如： fun (int x,int y,int z)
    {   z=x*y; }
void main ( )
{
    int a=5, b=3, c=6;
    fun (a,b,c) ;
    printf ("%d",c) ;
}
```

(4) 函数的嵌套调用 (P169) 和递归调用 (P170)

函数只能嵌套调用，不能嵌套定义

(5) 函数原型 (P175)

int max(int x, int y); int max(int a, int b); int max(int, int);

当函数的定义出现在函数调用之前时，可以省略其函数原型。

(6) 一维数组与函数参数 (P178)

int a[10]; 函数调用 **fun(a);** 则对应的形式参数有如下形式：

void fun(int p[10]); void fun(int p[]); void fun(int *p);

(7) 二维数组与函数参数 (P181)

int a[3][4]; 函数调用 **fun(a);** 则对应的形式参数有如下形式：

void fun(int p[3][4]); void fun(int p[][4]); void fun(int (*p)[4]);

(8) 指针型函数 (P183)

int a(); int *p();

(9) 函数指针 (**P184**), 如

int (*p)(); int max();

p=max; 函数的调用形式则为: (*p)();

(10) 带参数的 main 函数 (**P191**)

P191 例题 8.25

(11) 变量的作用域: 局部变量和全局变量 (**P192**)

(12) 变量的存储类型: 动态变量 **auto** 与静态变量 **static** (**P194**)

(13) 带参数的宏定义 (**P201**)

重点: 带参数的宏定义的展开

P205 习题选择题 7、9、10

第 9-10 章

(1) 结构体 (**P211**)、共用体 (**联合体 P228**) 和枚举类型 (**P230**)

重点: 1、结构体名、结构体变量、结构体成员

2、结构体与共用体存储空间的区别

(2) 文件 (**P241**)