

# 第 1 章 上机实验指导

## 实验 01 C 语言的运行环境

### 一、实验目的

1. 熟悉 C 语言程序的运行环境，了解所用计算机系统软、硬配置。
2. 初步了解在该集成环境下如何编辑、编译、连接和运行一个 C 程序，即运行一个 C 程序的全过程。
3. 通过运行简单的 C 程序，初步了解 C 程序的基本结构及特性。

### 二、实验内容和步骤

1. 熟悉所用的系统。了解 Windows 资源管理器的使用方法，如文件的查看、复制、运行等，熟悉 Visual C++ 6.0 所在目录、文本文件的建立方法。
2. 进入 Visual C++ 6.0，并新建一个 C++ 源程序文件。
3. 熟悉 Visual C++ 6.0 的集成环境，了解各菜单项有哪些子菜单。
4. 输入下面的程序，注意区分大小写，编译并运行程序。

```
#include<stdio.h>
void main()
{ printf("This is a C program.\n"); }
```

5. 关闭工作区，新建一个程序，编写用 printf 语句将 3 个字符串：good morning, floppy disk, hard disk 在同一行显示的程序。

```
#include<stdio.h>
void main()
{ printf("good morning");
printf("floppy disk");
printf("hard disk \n"); }
```

6. 关闭工作区，新建一个程序，然后输入并运行一个需要在运行时输入数据的程序。

```
#include<stdio.h>
void main()
{int a,b,c;
int max(int x,int y);
printf("input a and b:");
scanf("%d,%d",&a,&b);
c=max(a,b);
printf("\nmax=%d",c);}
int max(int x,int y)
{int z;
if(x>y) z=x;
```

```
else z=y;
return(z);}
```

(1) 运行程序，若程序有错，则修改错误后继续运行程序，当没有错误信息时输入：2，5 并按 Enter 键，查看运行结果。

(2) 将程序的第三行改为“int a;b;c;”，然后编译看结果如何，将函数 max() 的第 3，4 行合并为一行，运行程序，看结果是否相同？

7. 编写一程序，功能是输出两行文字。

8. 编写一程序，功能是用键盘输入语句输入三个数，然后分别求它们的和、差、积、商。

## 实验 02 基本的程序语句 1

### 一、实验目的

1. 掌握 C 语言数据类型，熟悉定义一个整型、字符型和实型的变量，以及对它们赋值的方法。

2. 掌握不同数据类型之间赋值的规律，学会数据输入方式和数据输出格式及各种格式转义字符。

3. 学会使用 C 语言的运算符以及用这些运算符组成的表达式，特别是自增(++)和自减(--)运算符的使用。

4. 进一步熟悉 C 程序的编辑、编译、连接和运行的过程。

### 二、实验内容和步骤

1. 输入并运行下面程序，分析其运行结果。

```
#include<stdio.h>
void main()
{ char c1,c2;c1=46;c2=47;
printf("%3c%3c", c1,c2); printf("%3d%3d", c1,c2); }
```

将程序中 char c1,c2;改为 int c1,c2;再运行，分析其结果。

2. 输入并运行下面程序。

```
#include<stdio.h>
void main()
{ int a,b; float c,d; long e,f; unsigned int u,v; char c1,c2;
scanf("%d,%d",&a,&b); scanf("%f,%f", &c,&d);
scanf("%ld,%ld",&e,&f); scanf("%o,%o",&u,&v);
scanf("%c,%c", &c1,&c2); printf("\n");
printf("a=%4d,b=%4d\n",a,b); printf("c=%8.2f,d=%8.2fn",c,d);
printf("u=%o,v=%o\n",u,v); printf("c1=%c,c2=%c\n",c1,c2);}
```

运行上面程序，分析结果，特别注意输出 c1, c2 的值是什么？什么原因？

(1) 将输入 e 和 f、u 和 v 的语句分别改为：

```
scanf("%d,%d",&e,&f);
scanf("%d,%d",&u,&v);
```

运行并分析结果。

(2) 将程序的第一行加命令行：#include <math.h>，运行并分析结果。

3. 编写一个程序，求表达式  $x-z\%2*(x+y)\%2/2$  的值，设  $x=8.5$ ,  $y=2.5$ ,  $z=4$ 。
4. 先分析下面程序的结果，然后再上机运行，看结果上否一致？

```
#include<stdio.h>
void main()
{ int x,y,z; x=y=z=3;
  y=x++-1; printf("%4d%4d",x,y);
  y=++x-1; printf("%4d%4d",x,y);
  y=z--+1; printf("%4d%4d",z,y);
  y=--z+1; printf("%4d%4d",z,y);}
```

注：本例注意自增自减运算符，先赋值后自增（自减）和先自增（自减）后赋值的问题。

5. 下列程序错误在哪里？

- (1) /\*给定半径 r，求圆的面积 s \*/

```
#include<stdio.h>
void main()
{ float r,s; s=3.14*r*r;
  printf("s=%f\n",s) }
```

- (2) /\*给定长和宽 L 和 w，求矩形面积 s \*/

```
#include<stdio.h>
void main()
{ int L,w,s; scanf("%d%d",L,w);
  s=L*w; printf(L,w,s);}
```

- (3) 运行看看结果如何？为什么？

```
#include<stdio.h>
void main()
{ int i,j,k;float x,y,z;
  scanf("%d%f%f",&i,&j,&k);
  scanf("%d%f%f",x,y,z);
  i=i+x;y=y+j;z=z+i;k=x*y;
  printf("%d%f%f\n",i,j,k);
  printf("%f%d%d\n',x,y,z); }
```

- (4) 运行看看结果如何？为什么？

```
#include<stdio.h>
void main()
{ int i;float x;long y; i=100;x=200;y=300;
  printf("i=%d,x=%d,y=%d\n",i,x,y);
  printf("i=%f,x=%f,y=%f\n",i,x,y);
  printf("i=%ld,x=%ld,y=%ld\n",i,x,y); }
```

## 实验 03 基本的程序语句 2

### 一、实验目的

1. 掌握 C 语言数据类型，熟悉定义一个整型、字符型和实型的变量，以及对它们赋值的方法。
2. 掌握不同数据类型之间赋值的规律，学会数据输入方式和数据输出格式及各种格式转义字符。

3. 学会使用 C 语言的运算符以及用这些运算符组成的表达式，特别是自增(++)和自减(--)运算符的使用。
4. 进一步熟悉 C 程序的编辑、编译、连接和运行的过程。

## 二、实验内容和步骤

1. 输入并运行下列程序。

```
#include<stdio.h>
void main()
{ char  c1,c2;
  c1='a';
  c2='b';
  printf("%c,%c",c1,c2); }
```

- (1) 运行程序，写出运行结果。
- (2) 加入下面的一个语句作为“}”前的最后一个语句“printf("%d,%d\n",c1,c2);”。
- (3) 将第 3 行改为“int c1,c2;”然后再运行程序，并观察结果是否相同？
- (4) 将第 3 行改为“int c1,c2;”，将第 4, 5 行依次改为：

```
c1=a;c2=b;
c1="a";c2="b"
c1=300;c2=400;
```

每修改一次后运行程序，观察结果。

2. 输入并运行下面的程序。

```
#include<stdio.h>
void main()
{ int a,b;unsigned c,d;long e,f;
  a=100;b=-100;e=50000;f=32767;c=a;d=b;
  printf("%d,%d\n",a,b);
  printf("%o,%x\n",a,b);
  c=a;e;d=b;f;
  printf("%d,%d\n",a,b);
  printf("%o,%f\n",c,d); }
```

根据程序和运行结果分析：将一个负整数赋给一个无符号的变量，会得到什么结果？

3. 运行程序，并分析运行结果。

```
#include<stdio.h>
void main()
{ int i,j;i=8;j=10;
  printf("%d,%d\n",++i,++j);
  i=8;j=10;
  printf("%d,%d\n",i++,j++);
  i=8;j=10;
  printf("%d,%d\n",++i,i);
  i=8;j=10;
  printf("%d,%d\n",i++,i); }
```

4. 要将“China”译成密码，密码规律是：用原来的字母后面第 4 个字母代替原来的字母。如：字母“A”后面第 4 个字母是“E”，用“E”代替“A”，因此，“China”应译为“Glmre”。

请编写一程序，用赋初值方法使 c1、c2、c3、c4、c5 这 5 个变量的值分别为 'C'、'h'、'l'、'n'、'a'，经过计算，使 c1、c2、c3、c4、c5 分别变为 'G'、'l'、'm'、'r'、'e'，并输出。

提示：

main()函数算法如下：

定义 char 型变量 c1, c2, c3, c4, c5;

给字符型变量赋值 c1=c1+4;c2=c2+4;c3=c3+4;c4=c4+4;c5=c5+4;

输出 c1, c2, c3, c4, c5。

5. 参考下面程序，如何改写输入函数，并配合正确的键盘输入方法才能使 x, y 和 ch 分别获得值 10、100 和 'A'？

```
#include<stdio.h>
void main()
{ int x,y;char ch;
scanf("x=%d,y=%d,ch=%c",&x,&y,&ch);
printf("x=%d,y=%d,ch=%c\n",x,y,ch); }
```

若将输入函数改为 scanf("%d,%d,%c",&x,&y,&ch);

或 scanf("%d,%c,%d",&x,&ch,&y);

或 scanf("%c,%d,%d",&ch,&x,&y);

结果将会如何？

6. 下面程序对输入有何要求？利用它可以做什么？

```
#include<stdio.h>
void main()
{ int x,y,z; long m;
scanf("%d%o%x",&x,&y,&z);
scanf("%ld",&m);
printf("x=%d,%o;%x\n",x,x,x);
printf("y=%d,%o;%x\n",y,y,y);
printf("z=%d,%o;%x\n",z,z,z);
printf("m=%ld,%lo;%lx\n",m,m,m);}
```

7. 运行下列输出宽度及控制的程序，分析运行结果。

```
#include<stdio.h>
void main()
{ int i,j;float x,y; long int m;i=688;j=-32765;
x=12345.678;y=-48765.432;m=1234567890;
printf("%d,%8d,%08d,%-8d\n",i,i,j,j);
printf("%f,%12.2f,%12.2f,%-12.2f\n",x,x,y,y);
printf("%ld,%lu,%12ld,%-12d\n",m,m,m,m);}
```

## 实验 04 选择结构程序设计 1

### 一、实验目的

1. 熟悉关系表达式和逻辑表达式的使用。
2. 掌握 break 的使用。

3. 熟悉掌握 if 语句和 switch 语句的应用。

## 二、实验内容和步骤

1. 三个整数 a, b, c, 由键盘输入这三个数, 求三个数中的最大值。

```
#include<stdio.h>
void main()
{ int a,b,c;
  printf("input a,b,c:");
  scanf("%d,%d,%d",&a,&b,&c);
  if(a<b)
    if( b<c)   printf("max=%d\n",c);
    else      printf("max=%d\n",b);
  else if(a<c) printf("max=%d\n",c);
    else      printf("max=%d\n",a); }
```

2. 运行下面程序, 分析 if 和 else 是哪两个相互“配对”, 在书写程序时, 分出层次, 这样有利于程序的可读性, 容易查找出错误。此程序还有更加简明的方法实现, 就是利用条件表达式。

```
#include<stdio.h>
void main()
{ int a,b,c,max,t;
  printf("input a,b,c:");
  scanf("%d,%d,%d",&a,&b,&c);
  t=(a>b)? a : b;
  max=(t>c)?t : c;
  printf("max=%d",max); }
```

可利用此程序考虑怎样修改求出三个数中的最小值。

3. 先读下面程序, 分析出程序的执行结果, 然后再上机运行, 结果是否一致?

```
#include<stdio.h>
void main()
{ int x,y=1,z=1;
  if (y!=0) x=5;
  printf("x=%d\t",x);
  if (y==0) x=3;
  else      x=5;
  printf("x=%d\t\n",x);
  x=1;
  if(z<0)
    if(y>0) x=3;
    else    x=5;
  printf("x=%d\t\n",x);
  if (z=y<0) x=3;
  else if (y==0) x=5;
    else    x=7;
  printf("x=%d\t",x);
  printf("z=%d\t\n",z);
  if (x=z=y) x=3;
  printf("x=%d\t",x);
  printf("z=%d\t\n",z); }
```

4. 用 scanf 函数输入 x 的值, 求 y 的值, 编程解决问题, 然后上机调试运行程序。

提示:

main()函数结构如下:

```

定义实型变量 x 与 y
使用 scanf 函数输入 x 的值
if x<1
    y=x
else
if x<10
    y=2x-1
else
    y=3x-11
输出 x 的值与 y 的值

```

5. 给出一个百分制的成绩, 输出成绩等级 A, B, C, D, E。90 分及以上为 A, 80—89 为 B, 70—79 为 C, 60—69 为 D, 60 分以下为 E。要求从键盘输入成绩, 然后输出相应等级, 用 if 语句实现。

提示:

使用 if 语句的 main()函数结构如下:

```

定义 float 型变量 score, char 型变量 grade
输入百分制成绩赋给 score
if score>=90
    grade='A'
else if score>=80
    grade='B'
else if score>=70
    grade='C'
else if score>=60
    grade='D'
else grade='E'

```

输出百分制成绩和等级。

## 实验 05 选择结构程序设计 2

### 一、实验目的

1. 熟悉关系表达式和逻辑表达式的使用。
2. 掌握 break 语句的使用。
3. 熟悉掌握 if 语句和 switch 语句的应用和区别。

### 二、实验内容和步骤

1. 有一函数:

$$y = \begin{cases} x & (x < 0) \\ 3x - 2 & (10 \leq x < 50) \\ 4x + 1 & (50 \leq x < 100) \\ 5x & (x \geq 100) \end{cases}$$

输入  $x$  的值, 求  $y$  的值。

程序如下:

```
#include<stdi.h>
void main()
{ int x,y,t;
  printf("input x=:");
  scanf("%d",&x);
  if(x<10) t=0;
  if(x>=100) t=10;
  else      t=x/10;
  switch(t)
  { case 0:  y=x; break;
    case 1:
    case 2:
    case 3:
    case 4:  y=3*x - 2;break;
    case 5:
    case 6:
    case 7:
    case 8:
    case 9:  y=4*x+1;break;
    case 10: y=5*x;  }
  printf("y=%d",y); }
```

运行程序, 写出执行结果。本题还可以单独用 `if` 语句实现, 方法简单, 程序可读性好, 学生自己编程, 上机运行。

2. 给出一个百分制的成绩, 输出成绩等级 A, B, C, D, E。90 分及以上为 A, 80—89 为 B, 70—79 为 C, 60—69 为 D, 60 分以下为 E。要求从键盘输入成绩, 然后输出相应等级, 用 `switch` 语句实现。

**提示:**

使用 `switch` 语句的 `main` 函数结构如下:

```
定义 float 型变量 score, char 型变量 grade
输入百分制成绩赋给 score
switch((int) (score/10))
{ case 10:
  case 9: grade='A';break;
  case 8: grade='B';break;
  case 7: grade='C';break;
  case 6: grade='D';break;
  default: grade='E';break; }
输出百分制成绩和等级
```

3. 编程实现: 输入一个不多于 5 位的正整数, 要求: ①输出它是几位数, ②分别输出每一位数字; ③按逆序输出各位数字, 如原数为 321, 则应输出 123。

应准备以下测试数据: 要处理的数为 1 位正整数; 要处理的数为 2 位正整数; 要处理的



数为3位正整数；要处理的数为4位正整数；要处理的数为5位正整数。

除此之外，程序还应当对不合法的输入作必要的处理。例如：输入负数；输入的数超过5位。

**提示：**

main()函数结构如下：

```

定义 int 型变量 c1,c2,c3,c4,c5,num
输入一个不超过 5 位的正整数赋给 num
if num>99999
输出：输入的数超过 5 位
else if num<0
输出：输入的数是一个负数
else
{ 求得 num 的各位数分别赋给 c1,c2,c3,c4,c5
  c1=num/10000;
  c2=(num-c1*10000)/1000;
  c3=(num/100)%10;
  c4=(num/10)%10;
  c5=num%10;}
if(c1>0)
  {printf("\n%d 是一个 5 位数\n",num);
  printf("其各位分别为： %d,%d,%d,%d,%d\n",c1,c2,c3,c4,c5);
  printf("逆序输出为： %d,%d,%d,%d,%d\n",c5,c4,c3,c2,c1); }
else if(c2>0) 是 4 位数,输出其各位,格式与 5 位数类似
else if(c3>0) 是 3 位数,输出其各位,格式与 5 位数类似
else if(c4>0) 是 2 位数,输出其各位,格式与 5 位数类似
else if(c5>0) 是 1 位数,输出其各位,格式与 5 位数类似 }

```

4. 编程实现：输入4个整数，要求按由小到大的顺序输出。得到正确结果后，修改程序使之按由大到小的顺序输出。

**提示：**

main()函数结构如下：

```

int a,b,c,d,t;
输入 4 个整数:赋给 a,b,c,d;
if(a>b) 交换 a,b
if(a>c) 交换 a,c
if(a>d) 交换 a,d
if(b>c) 交换 b,c
if(b>d) 交换 b,d
if(c>d) 交换 c,d
输出 a,b,c,d

```

5. 输入一个字符，如果是大写字母改变为小写字母；如果是小写字母，则把它变为大写字母；若是其他字符则不变。

6. 输入两个数  $x$  和  $y$ ，以及一个符号  $c$ ，若为“+”、“-”、“\*”、“/”，则输出  $x+y$ 、 $x-y$ 、 $x*y$ 、 $x/y$ ，若  $c$  是其他符号，则输出错误信息。

## 实验 06 循环结构程序设计 1

### 一、实验目的

1. 了解 goto 语句和 if 语句构成的循环。
2. 熟悉使用 while 语句、do-while 语句和 for 语句实现循环的方法。
3. 掌握在程序设计中用循环的方法实现一些常用算法（如穷举、迭代、递推等）。
4. 熟悉掌握三种循环语句的应用，会使用循环嵌套进行编程。

### 二、实验内容和步骤

1. 输入两个正整数 m 和 n，求出它们的最大公约数和最小公倍数。

输入时，使  $m < n$ ，观察结果是否正确？再输入时，使  $m > n$ ，观察结果是否正确？修改程序使对任何的整数都能得到正确的结果。

提示：

main()函数结构如下：

```
int m,n,r,tm,tn;
输入两个正整数赋给 m,n
tm=m;
tn=n;
if(m<n)交换 m,n
r=m%n;
while(r)
{ m=n;
  n=r;
  r=m%n;}

```

输出最大公约数 n 和最小公倍数  $tm*tn/n$ 。

2. 编写程序利用公式：
$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$
求 e 的近似值，精确到小数点后 6 位。

提示：

main()函数结构如下：

```
定义 int 型变量 i,double 型变量 e,p,t
e=1.0;t=1.0;p=1.0;i=1
while(t>=1e-7)
{ e=e+t;
  i++;
  p=p*i;
  t=1.0/p;}

```

输出 e 的值

3. 求  $\sum_{n=1}^{100} n(n+1)(1*2+2*3+3*4+\dots+99*100)$ 。上机运行，并记录下结果，然后用另外两种循环语句实现上述功能。

4. 根据以下公式计算 s，n 从键盘输入。

$$s=1+1/(1+2)+1/(1+2+3)+\dots+1/(1+2+3+\dots+n)$$

例如，若  $n$  输入为 11 时，函数的值为 1.833333。

## 实验 07 循环结构程序设计 2

### 一、实验目的

1. 了解 goto 语句和 if 语句构成的循环。
2. 熟悉使用 while 语句、do-while 语句和 for 语句实现循环的方法。
3. 掌握在程序设计中用循环的方法实现一些常用算法（如穷举、迭代、递推等）。
4. 熟悉掌握三种循环语句的应用，会使用循环嵌套进行编程。

### 二、实验内容和步骤

1. 编程求 1 到  $n$  中能被 3 或 7 整除的数之和，分别用 for 循环语句和 while 循环语句完成本题。

提示：

for 循环语句的主要算法如下：

```
for(i=1;i<=n;i++)
    if i 能被 3 或 7 整除
        sum=sum+i
```

while 循环语句的主要算法如下：

```
while(i<=n)
    if i 能被 3 或 7 整除
        sum=sum+i;
```

```
i++;
```

2. 求 2~1000 之间的全部素数（每行显示 10 个数）。

参考程序：

```
#include<stdio.h>
#include<math.h>
void main()
{ int m=3,k,i,n=1;   printf("%7d",2);
  do { if(n%10==0) printf("\n");
      k=sqrt(m);
      for(i=2;i<=k;i++)
          if(m%i==0) break;
      if(i>=k+1)      { printf("%8d",m); n++;}
  }while(++m<=1000);
  printf("\n"); }
```

3. 打印出三位整数中的所有水仙花数。水仙花数满足的条件是该数等于其各位上数字的立方之和。例如： $153=1^3+5^3+3^3$ 。

4.  $sn=a+aa+\dots+a\cdots a$ ，其中  $a$  是 1~9 中的一个数字。 $n$  为一正整数， $a$  和  $n$  均从键盘输入。例如，输入  $n$  和  $a$  均为 4， $sn=4+44+444+4444$ 。

5. 求出 1~1000 内的所有完数。完数满足的条件是该数等于其所有因子（不包括本身）

之和。例如：6=1+2+3。

## 实验 08 循环结构程序设计 3

### 一、实验目的

1. 了解 goto 语句与 if 语句构成的循环。
2. 熟悉使用 while 语句、do-while 语句和 for 语句实现循环的方法。
3. 掌握在程序设计中用循环的方法实现一些常用算法（如穷举、迭代、递推等）。
4. 熟悉掌握三种循环语句的应用，会使用循环嵌套进行编程。

### 二、实验内容和步骤

1. 分别用三种循环语句（while 语句、do-while 语句、for 语句），实现求 1~100 的累加和。编程上机调试，总结出三种循环语句哪种实现起来方便、灵活。

2. 换零钱。把一元钱全兑换成硬币，有多少种兑换方法？

参考程序：

```
#include<stdio.h>
void main()
{ int i,j,k,n;
  n=100,k=0;
  for(i=0;i<=n/5;i++)
    for(j=0;j<=(n-i*5)/2;j++)
      { printf(" 5 cent=%d\t 2 cent=%d\t 1 cent=%d\n",i,j,n-i*5-j*2);
        k++;    }
  printf("total times=%d\n",k);
}
```

3. 计算如下公式的值：

$$y=1+1/2*2+1/3*3+1/4*4+...+1/m*m$$

例如，若 m 的值为 5，则应输出 1.463611。

4. 利用 for 循环结构编写程序，打印以下图案。

```

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

5. 打印九九乘法口诀表。

## 实验 09 函数及预处理命令 1

### 一、实验目的

1. 掌握定义函数的方法。
2. 掌握函数实参及形参的对应关系以及“值传递”方式。
3. 掌握函数的嵌套调用和递归调用的方法。
4. 掌握全局变量和局部变量、动态变量和静态变量的概念与使用方法。
5. 学会使用宏替换编写程序，弄清“文件包含”的作用。

### 二、实验内容和步骤

1. 通过运行下面程序，熟悉函数的调用方法。

```
#include<stdio.h>
fun(int i, int j, int k);
void main()
{ int x,y,z;x=4;y=12;z=6;
  fun(x,y,z);
  printf("x=%d;y=%d;z=%d\n",x,y,z);}
fun(int i, int j, int k)
{ int t;    t=(i+j+k)/2;
  printf("t=%d\n",t); }
```

2. 运行下面程序，写出执行结果。

```
#include<stdio.h>
f(int i,int j);
void main()
{ int i,j,x,y,n,g;    i=4;j=5;g=x=6;y=9;n=7;
  f(n,6);
  printf("g=%d;i=%d;j=%d\n",g,i,j);
  printf("x=%d;y=%d\n",x,y);
  f(n,8); }
f(int i,int j)
{ int x,y,g;    g=8;x=7;y=2;
  printf("g=%d;i=%d;j=%d\n",g,i,j);
  printf("x=%d;y=%d\n",x,y); }
```

3. 编写一个判断素数的函数，在主函数输入一个整数，输出是否是素数的信息。

参考程序：

```
#include<stdio.h>
void main()
{ int m; void prime(int n);
  printf("Please input a data m=:"); scanf("%d",&m);
  prime(m); }
void prime(int n)
```

```

{ int i,k;   k=n/2;
  for(i=2;i<=k;i++) if(n%i==0) break;
  if(i>=k+1) printf("This is a prime number");
  else      printf("This isn't a prime number"); }

```

4. 写出一个判别素数的函数，在主函数输入一个整数，输出是否素数的信息。本程序应准备以下测试数据：17，34，2，1，0，分别输入数据，运行程序并检查结果是否正确。

提示：

求素数函数如下：

```

int prime(int n)
{if n<2 return 0
  for(i=2;i<=n/2;i++)
    if n 能被 i 整除 return 0;
  return 1;}

```

main()函数中输入一个整数赋给变量 n，通过 prime(n)判断其是否素数，若函数值为 1 则是素数否则不是素数。

5. 用下面的公式求  $\pi$  的近似值，直到最后一项的绝对值小于指定的数（参数 num）为止： $\pi/4 \approx 1-1/3+1/5-1/7+\dots$ ，主函数已经给出，请在函数 fun 的花括号中填入你编写的若干语句。例如，程序运行后输入 0.0001，则程序输出 3.1414。

```

#include<math.h>
#include<stdio.h>
float fun(float num)
{

}

void main()
{ float n1,n2;printf("enter a float number:");scanf("%f",&n1);
  n2=fun(n1);
  printf("%6.4f\n",n2); }

```

## 实验 10 函数及预处理命令 2

### 一、实验目的

1. 掌握定义函数的方法。
2. 掌握函数实参及形参的对应关系以及“值传递”方式。
3. 掌握函数的嵌套调用和递归调用的方法。
4. 掌握全局变量和局部变量、动态变量和静态变量的概念与使用方法。
5. 学会使用宏替换编写程序，弄清“文件包含”的作用。

### 二、实验内容和步骤

1. 用二分法求方程  $2x^3-4x^2+3x-6=0$  的一个根，并要求绝对误差不超过 0.001。例如，若给 m 输入 -100，给 n 输入 90，则函数求得的一个根值为 2.000。

请改正程序中的错误，使它能得出正确结果。

```
#include<stdio.h>
#include<math.h>
double funx(double x)
{return(2*x*x*x-4*x*x+3*x-6);}
double fun(double m,double n)
{
/*****found*****/
int r;
r=(m+n)/2;
/*****found*****/
while(fabs(n-m)<0.001)
{if(funx(r)*funx(n)<0) m=r;
else n=r;
r=(m+n)/2;}
return r;
}
void main()
{double m,n,root;
printf("enter m,n:\n");
scanf("%lf%lf",&m,&n);
root=fun(m,n);
printf("root=%6.3fn",root); }
```

2. 求  $s=aa\dots aa\dots aaaa-aa-a$  (此处  $aa\dots aa$  表示  $n$  个  $a$ ,  $a$  和  $n$  的值在 1 至 9 之间), 例如  $a=3$ ,  $n=6$ , 则以上表达式为:  $s=333333-33333-3333-333-33-3$ , 其值是: 296298。主函数已经给出, 请在函数 `fun` 的花括号中填入你编写的若干语句。

```
#include<stdio.h>
long fun(int a,int n)
{

}

void main()
{int a,n;
printf("\npleae enter a and n:");
scanf("%d%d",&a,&n);
printf("The value of function is:%ld\n",fun(a,n)); }
```

3. 用递归法将一个整数  $n$  转换成字符串。例如, 输入 483, 应输出字符串"483"。 $n$  的位数不确定, 可以是任意的整数。

提示:

```
void convert(int n)
{ int i;
if((i=n/10)!=0)
convert(i);
putchar(n%10+'0');}
```

在 `main()` 函数中输入一个整数, 然后先输出该数的符号, 然后调用函数 `convert(n)`。

4. 求两个整数的最大公约数和最小公倍数。用一个函数求最大公约数用另一个函数根据求出的最大公约数求最小公倍数，分别用下面的两种方法编程。

(1) 不用全局变量，在主函数中输入两个数和输出结果。

(2) 用全局变量，数据的传递通过全局变量的方法。

**提示：**

使用下面函数求最大公约数，其中  $v$  为最大公约数，若将  $v$  设为外部变量，则可不使用 `return` 语句。

```
int hcf(int u,int v)/*求最大公约数*/
{ int t,r;
  if(v>u){t=u;u=v;v=t;}
  while((r=u%v)!=0)/*余数 r 不为 0 时继续作辗转相除法*/
  {u=v;v=r;}
  return(v);}

```

5. 写一个函数，输入一个十六进制数，输出相应的十进制数。

**提示：**

输入时将十六进制数作为一个字符串输入，然后将其每一个字符转换成十进制数并累加，转换方法如下：

```
if(s[i]>'0'&&s[i]<='9')    n=n*16+s[i]-'0';
if(s[i]>='a'&&s[i]<='f')    n=n*16+s[i]-'a'+10;
if(s[i]>='A'&&s[i]<='F')    n=n*16+s[i]-'A'+10;

```

6. 先读下面程序，分析出程序的执行结果，然后再上机运行，结果是否一致？

```
#define MA(x) x*(x-1)
#include<stdio.h>
void main()
{ int a=1,b=2;
  printf("%d\n",MA(1+a+b)); }

```

7. 先读下面程序，分析出程序的执行结果，然后再上机运行，结果是否一致？

```
#define f(x) x*x
#include <stdio.h>
void main()
{ int a=6,b=2,c;
  c=f(a)/f(b);
  printf("%d\n",c);}

```

## 实验 11 数组 1

### 一、实验目的

1. 掌握一维数组与二维数组的定义、赋值及输入输出方法。
2. 掌握字符数组和字符串函数的使用。
3. 掌握与数组有关的算法（特别是排序算法）。



## 二、实验内容和步骤

1. 有一数组，内放 10 个整数，要求找出最小的数和它的下标，然后把它和数组中最前面的元素对换位置。编写程序，上机运行，并记录下结果。

**提示：**

数组的 10 个元素可用输入函数 `scanf()` 通过键盘输入进去，找出数组中最小的元素可通过循环语句和条件语句来实现。设 `min` 是存放数组中最小元素的变量，`array[k]` 为一个暂存单元。实现最前面的元素与最小元素对换可通过下面语句实现：

```
array[k]=array[0];
array[0]=min;
```

参考程序：

```
#include<stdio.h>
void main()
{int i,array[10],min,k=0;
 printf("Please input 10 data\n");
 for(i=0;i<10;i++)
   scanf("%d",&array[i]);
 printf(" Before exchange:\n");
 for(i=0;i<10;i++)
   printf("%5d",array[i]);
 min=array[0];
 for(i=1;i<10;i++)
   if(min>array[i])
     { min=array[i];
       k=i; }
 array[k]=array[0];
 array[0]=min;
 printf("After exchange:\n");
 for(i=0;i<10;i++)
   printf("%5d",array[i]);
 printf("k=%d\t min=%d\n",k,min);}

```

2. 在一个已排好序的数列中（由小到大）再插入一个数，要求仍然有序，编程并运行。

**提示：**编程时应考虑到插入的数的各种可能性（比原有所有的数大；比原有所有的数小；在最大数和最小数之间）。

参考程序：

```
#include<stdio.h>
void main()
{ int i,n;float a,x[20],y[21];
 printf("Please input n value\n");scanf("%d",&n);
 printf("Please input value (from small to big)\n");
 for(i=0;i<n;i++) scanf("%f",&x[i]);
 printf("Insert value=?"); scanf("%f",&a);
 i=0;
 while(a>x[i]&& i<n)
   { y[i]=x[i]; i++; }

```

```

y[i]=a;
for(i=i+1;i<n+1;i++) y[i]=x[i-1];
printf("\n");
for(i=0;i<n+1;i++)
    { printf("%8.2f",y[i]);
      if((i+1)%5==0) puts("\n"); }
}

```

3. 用选择法对 10 个整数排序, 要求 10 个整数用 `scanf` 函数输入。

提示:

输入 10 个整数存放到数组 `a` 的 `a[1]` 到 `a[10]` 中

输出 10 个数

```

for(i=1;i<10;i++)
    { min=i;
      for(j=i+1;j<=10;j++)
          if(a[min]>a[j]) min=j;
      交换 a[i]与 a[min] }

```

输出排序后的 10 个数

4. 由形参给定 `n` 个实数, 输出平均值, 并统计在平均值以上 (含平均值) 的实数个数。

请改正程序中的错误, 使它能得出正确结果。例如, `n=8` 时, 输入 193.199、195.673、195.757、196.051、196.092、196.596、196.579、196.763 所得平均值为 195.838745, 在平均值以上的实数个数为 5。

```

#include<stdio.h>
int fun(double x[ ],int n)
{
    /******found******/
    int j,c=0,double xa=0.0;
    for(j=0;j<n;j++)
        xa+=x[j]/n;
    printf("ave=%f\n",xa);
    for(j=0;j<n;j++)
        /******found******/
        if(x[j]>=xa)
            c++;
    return c; }
void main()
{double x[100]={ 193.199,195.673,195.757,196.051,196.092,196.596,196.579,196.763};
  printf("%d\n",fun(x,8)); }

```

5. `m` 个人的成绩存放在 `score` 数组中, 将低于平均分的人数作为函数值返回, 将低于平均分的分数放在 `below` 所指的数组中。

例如, 当 `score` 数组中的数据为 10、20、30、40、50、60、70、80、90 时, 函数返回的人数应该是 4, `below` 中的数据应为 10、20、30、40。主函数已经给出, 请在函数 `fun` 的花括号中填入你编写的若干语句。

```

#include<stdio.h>
int fun(int score[ ],int m,int below[ ])
{

```

```

}
void main()
{ int i,n,below[9];int score[9]={ 10,20,30,40,50,60,70,80,90};
  n=fun(score,9,below);
  printf("\nbelow the average score are:");
  for(i=0;i<n;i++) printf("%d",below[i]); }

```

## 实验 12 数组 2

### 一、实验目的

1. 掌握一维数组与二维数组的定义、赋值及输入输出方法。
2. 掌握字符数组和字符串函数的使用。
3. 掌握与数组有关的算法（特别是排序算法）。

### 二、实验内容和步骤

1. 编写一程序，一班级有 n 名学生要求按他们姓名的顺序排列（按汉语拼音的字母顺序从小到大），并按序输出。

参考程序：

```

#include <stdio.h>
#include <string.h>
void strup(char str[]);
void main()
{ char str[20];
  char name[20][20];
  int i,j,t,n;
  printf("Please input name number of sorting\n");
  scanf("%d",&n);
  printf("Please input name\n");
  for(i=0;i<n;i++)
  { gets(name[i]);
    strup(name[i]); }
  for(i=0;i<n;i++)
  { for(j=i+1;j<n;j++)
    { for(k=0;;k++)
      if(name[i][k]<name[j][k])
        break;
      else if(name[i][k]>name[j][k])
        {strcpy(str,name[j]);
         strcpy(name[j],name[i]);
         strcpy(name[i],str);
         break;}
    }
  }
}

```

```

    }
    for(i=0;i<n;i++)
        printf("%s",name[i]); }
void strup(char str[])
{ int i;
  for(i=0;str[i]!='\0';i++)
    if(str[i]>='a'&&str[i]<='z') str[i]=str[i]+'A'-'a'; }

```

2. 将两个串连接起来，不要用 `strcat` 函数。

提示：

```

分别输入两个字符串
i 指向第一个字符串的结束标志'\0'
while(s2[j]!='\0')
    s1[i++]=s2[j++];
s1[i]='\0';
输出连接后的字符串

```

3. 输出 10 阶杨辉三角形。

程序参考：

```

#include<stdio.h>
#define N 10
void main()
{ int a[N][N],i,j;
  for(i=1;i<N;i++) /*形成主对角线和第 1 列元素*/
    { a[i][1]=1;a[i][i]=1; }
  for(i=3;i<N;i++) /*形成其余元素*/
    for(j=2;j<=i-1;j++)
      a[i][j]=a[i-1][j-1]+a[i-1][j];
  for(i=1;i<N;i++) /*输出*/
    { for(j=1;j<=30-2*i;j++) printf(" ");
      for(j=1;j<=i;j++) printf("%4d",a[i][j]);
      printf("\n"); } }

```

4. 从键盘输入一个字符串，并判断是否形成回文（即正序和逆序一样，如"abcdcba"）？  
 5. 按字符数组输入字符串，将其中的英文字母都改成“\*”，然后按字符串形式输出。

## 实验 13 数组 3

### 一、实验目的

1. 掌握一维数组与二维数组的定义、赋值及输入输出方法。
2. 掌握字符数组和字符串函数的使用。
3. 掌握与数组有关的算法（特别是排序算法）。

### 二、实验内容和步骤

1. 有 15 个数存放在一个数组中，输入一个数要求用折半查找法找出该数是数组中的第几个元素的值，如果该数不在数组中，则输出无此数，要找的数用 `scanf` 函数输入。

**提示:**

用循环语句输入 15 个数

调用排序算法对其进行排序

```
while(flag)
    {  输入要查找的数
      loca=0;
      top=0;
      bott=N-1;
      if(number<a[0]||number>a[N-1]) loca=-1;
      while(sign==1&&top<=bott&&loca>=0)
      {  mid=(bott+top)/2;
        if(number==a[mid])
          {loca=mid;
           printf("找到了,数%d 在数组的第%d 位、\n",number,loca+1);
           sign=0;}
          else if(number<a[mid])    bott=mid-1;
            else top=mid+1;  }
      if(sign==1||loca==-1) printf("\n 查无此数\n");
      printf("\n 是否继续查找? (Y/N)");
      scanf("%c",&c);getchar();
      printf("\n");
      if(c=='N'||c=='n') flag=0;    }
```

2. 实现  $B=A+A'$ , 即把矩阵 A 加上 A 的转置, 存放在矩阵 B 中。计算结果在 main 函数中输出。主函数已经给出, 请在函数 fun 的花括号中填入你编写的若干语句。

例如, 输入下面的矩阵:

1 2 3

4 5 6

7 8 9

其转置矩阵:

1 4 7

2 5 8

3 6 9

程序输出:

2 6 10

6 10 14

10 14 18

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

```
void fun(int a[3][3],int b[3][3])
```

```
{
```

```
}
```

```
void main()
```

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

```
fun(a,t);
```

```

for(i=0;i<3;i++)
{for(j=0;j<3;j++) printf("%7d",t[i][j]);
printf("\n");}
}

```

3. 计算  $N \times N$  矩阵的主对角线元素和反对角线元素之和，并作为函数值返回。注意要求先累加主对角线元素中的值，然后累加反向对角线元素中的值。例如，若  $N=3$ ，有下列矩阵：

```

1  2  3
4  5  6
7  8  9

```

fun 函数首先累加 1、5、9，然后累加 3、5、7，函数的返回值为 30。在下划线处填入正确的内容，使程序得出正确的结果。

```

#include<stdio.h>
#define N 4
fun(int t[][N],int n)
{int i,sum;
  _____;
for(i=0;i<n;i++)
sum+=_____2_____
for(i=0;i<n;i++)
sum+=t[i][n-i-_____3_____];
return sum; }
void main()
{int t [][N]={21,2,13,24,25,16,47,38,29,11,32,54,42,21,3,10},i,j;
printf("\nthe original data:\n");
for(i=0;i<N;i++)
{for(j=0;j<N;j++) printf("%4d",t[i][j]);
printf("\n");}
printf("the result is:%d",fun(t,N)); }

```

4. 把形参 a 所指数值中的奇数按原顺序依次存放到 a[0]、a[1]、a[2]、.....中，把偶数从数组中删除，奇数个数通过函数值返回。例如，若 a 所指数值中的数据最初排列为：9、1、4、2、3、6、5、8、7，删除偶数后 a 所指数值中的数据为：9、1、3、5、7，返回值为 5。在下划线处填入正确的内容，使程序得出正确的结果。

```

#include<stdio.h>
#define N 9
int fun(int a[ ],int n)
{int i,j;j=0;
for(i=0;i<n;i++)
if(a[i]%2== _____1_____) { a[j]=a[i]; _____2_____; }
return _____3_____; }
void main()
{int b[N]={9,1,4,2,3,6,5,8,7},i,n;
printf("\nThe original data:\n");
for(i=0;i<N;i++) printf("%4d",b[i]);
printf("\n");
n=fun(b,N);
}

```

```
printf("\nThe number of odd:%d\n",n);
printf("\nThe odd number:\n");
for(i=0;i<n;i++) printf("%4d",b[i]);
printf("\n"); }
```

5. 将十进制正整数  $m$  转换成  $k$  ( $2 \leq k \leq 9$ ) 进制数, 并按高位到低位顺序输出。例如, 若输入 8 和 2, 则应输出 1000 (即十进制数 8 转换成二进制表示是 1000)。请改正程序中的错误, 使它能得出正确结果。

```
#include<stdio.h>
void fun(int m,int k)
{int aa[20],i;
for(i=0; m; i++)
{
/*****found*****/
aa[i]=m/k;
m/=k; }
for(; i; i--)
/*****found*****/
printf("%d",aa[i]); }
void main()
{ int b,n;printf("\please enter a number and a base:\n");
scanf("%d %d",&n,&b);
fun(n,b);printf("\n"); }
```

## 实验 14 指针 1

### 一、实验目的

1. 通过实验进一步掌握指针的概念, 会定义和使用指针变量。
2. 能正确使用数组的指针和指向数组的指针变量。
3. 能正确使用字符串的指针和指向字符串的指针变量。
4. 能正确使用指向函数的指针变量。
5. 了解指向指针的指针概念及其使用方法。

### 二、实验内容和步骤

1. 交换主函数中两个变量的值。请改正程序中的错误, 使它能得出正确结果。

```
#include<stdio.h>
/*****found*****/
void fun(int x,int y)
{int t;
/*****found*****/
t=x;x=y;y=t;
}
void main()
{ int a,b;
```

```

a=8;b=3;
fun(&a,&b);
printf("%d,%d\n",a,b); }

```

2. 将两个两位的正整数  $a$ 、 $b$  合并形成一个整数放在  $c$  中。合并的方式是：将  $a$  数的十位和个位数依次放在  $c$  数的个位和百位上， $b$  数的十位和个位数依次放在  $c$  数的千位和十位上。例如：当  $a=45$ ， $b=12$  时，调用该函数后， $c=1524$ 。主函数已经给出，请在函数  $fun$  的花括号中填入你编写的若干语句。

```

#include<stdio.h>
void fun(int a,int b,long *c)
{

}

void main()
{int a,b;
long c;
printf("input a,b:");
scanf("%d,%d",&a,&b);
fun(a,b,&c);
printf("the result is:%ld\n",c); }

```

3. 把主函数中输入的 3 个数，最大的放在  $a$  中，最小的放在  $c$  中，中间的放在  $b$  中。例如，输入的数为：55 12 34，输出结果是： $a=55.0$ ， $b=34.0$ ， $c=12.0$ 。请改正程序中的错误，使它能得出正确结果。

```

#include<stdio.h>
void fun(float *a,float *b,float *c)
{
/*****found*****/
float *k;
if(*a<*b)
{k=*a;*a=*b;*b=k;}
/*****found*****/
if(*a>*c)
{k=*c;*c=*a;*a=k;}
if(*b<*c)
{k=*b;*b=*c;*c=k;}
}

void main()
{float a,b,c;
printf("input a b c:");
scanf("%f%f%f",&a,&b,&c);
printf("a=%4.1f,b=%4.1f,c=%4.1f\n",a,b,c);
fun(&a,&b,&c);
printf("a=%4.1f,b=%4.1f,c=%4.1f\n",a,b,c);}

```

4. 找出一维整型数组元素中最大的值和它所在的下标，最大的值和它所在的下标通过形参传回。数组元素中的值已经在主函数中赋予。主函数中  $x$  是数组名， $n$  是  $x$  中的数据个数， $max$  存放最大值， $index$  存放最大值所在元素的下标。主函数已经给出，请在函数  $fun$  的花括



号中填入你编写的若干语句。

```
#include<stdio.h>
#include<stdlib.h>
void fun(int a[ ],int n,int *max,int *d)
{

}

void main()
{int i,x[20],max,index,n=10;
for(i=0;i<n;i++)
{x[i]=rand()%50;
printf("%4d",x[i]);}
printf("\n");
fun(x,n,&max,&index);
printf("max=%5d,index=%4d\n",max,index);
}
```

5. 从低位开始取出长整型变量  $s$  中偶数位上的数，依次构成一个新数放在  $t$  中。高位仍在高位，低位仍在低位。例如，当  $s$  中的数为：7654321 时， $t$  中的数为：642。请改正程序中的错误，使它能得出正确结果。

```
#include<stdio.h>
/*****found*****/
void fun(long s,long t)
{long s1=10;s/=10;*t=s%10;
/*****found*****/
while(s<0)
{s=s/100;*t=s%10*s1+*t;s1=s1*10;}
}
main()
{long s,t;
printf("\nPlease enter s:");scanf("%ld",&s);
fun(s,&t);
printf("The result is:%ld\n",t);}
```

## 实验 15 指针 2

### 一、实验目的

1. 通过实验进一步掌握指针的概念，会定义和使用指针变量。
2. 能正确使用数组的指针和指向数组的指针变量。
3. 能正确使用字符串的指针和指向字符串的指针变量。
4. 能正确使用指向函数的指针变量。
5. 了解指向指针的指针概念及其使用方法。

## 二、实验内容和步骤

1. 有四个字符串"Changhua"、"Liping"、"Chenmei"、"Gaofeng", 代表四个人的名字, 要求按字母顺序(由小到大)输出这四个字符串。编写此程序, 用数组处理, 上机调试并运行, 记录下输出结果。

参考程序:

```
#include<stdio.h>
void main()
{ void sort(char *a[ ],int n);
  static char *name[]={ "Changhua", "Liping", "Chenmei", "Gaofeng" };
  int n=4,i;
  sort(name,n);
  for(i=0;i<5;i++)
    printf("%s\n",name[i]); }
void sort(char *a[],int n)
{ char *temp;      int i,j;
  for(i=0;i<n-1;i++)
    for(j=0;j<n-i-1;j++)
      if(strcmp(a[j],a[j+1])>0)
        { temp=name[j];
          name[j]=name[j+1];
          name[j+1]=temp; }
}
```

2. 从低位开始取出长正型变量 s 中奇数位上的数, 依次构成一个新数放在 t 中。高位仍在高位, 低位仍在低位。例如, 当 s 中的数为: 7654321 时, t 中的数为: 7531。请改正程序中的错误, 使它能得出正确结果。

```
#include<stdio.h>
/*****found*****/
void fun(long s,long t)
{long s1=10;
 *t=s%10;
 while(s>0)
 {s=s/100;
 *t=s%10*s1+*t;
 /*****found*****/
 s1=s1*100; }
}
void main()
{long s,t;
 printf("\nPlease enter s:");
 scanf("%ld",&s);
 fun(s,&t);
 printf("The result is:%ld\n",t); }
```

3. 用一个函数实现两个字符串的比较, 即自己写一个 strcmp 函数, 函数的原型为:

```
int strcmp(char *p1,char *p2);
```

设 p1 指向字符串 s1, p2 指向字符串 s2。要求当两个字符相同时返回 0, 若两个字符串不相等, 则返回它们两者第一个不同字符的 ASCII 码的差值。两个字符串 s1、s2 由主函数输入, strcmp 函数的返回值也由主函数输出。

提示:

```
int strcmp(char *p1,char *p2)
{ int i=0;
  while(*(p1+i)==*(p2+i))
    if(*(p1+i++)=='\0') return 0;
  return *(p1+i)-*(p2+i); }
```

4. 判断形参 s 所指字符串是否是“回文”。若是, 函数返回值为 1; 不是, 函数返回值为 0。“回文”是正读和反读都一样的字符串(不区分大小写)。例如: LEVEL 和 Level 是“回文”, 而 LEVLEV 不是“回文”。在下划线处填入正确的内容, 使程序得出正确的结果。

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int fun(char *s)
{char *lp,*rp;lp=___1___;rp=s+strlen(s)-1;
 while((toupper(*lp)==toupper(*rp))&&(lp<rp)){lp++;rp___2___;}
 if(lp<rp)___3___;
 else return 1; }
main()
{char s[81];printf("enter a string:");scanf("%s",s);
 if(fun(s)) printf("\n\"%s\" is a palindrome.\n\n",s);
 else printf("\n\"%s\" isn't a palindrome.\n\n",s);}
```

5. 假定输入的字符串中只包含字母和\*号。请编写函数 fun, 它的功能是: 将字符串尾部的\*号全部删除, 前面和中间的\*号不删除。例如, 字符串中的内容为\*\*\*\*A\*BC\*DEF\*G\*\*\*\*\* , 删除后, 字符串中的内容应当是\*\*\*\*A\*BC\*DEF\*G。在编写函数时, 不得使用 C 语言提供的字符串函数。主函数已经给出, 请在函数 fun 的花括号中填入你编写的若干语句。

```
#include<stdio.h>
void fun(char *a)
{
}

}
void main()
{ char s[81];printf("enter a string:\n");
 gets(s);fun(s);
 printf("the string after deleted:\n");
 puts(s); }
```

6. 假定输入的字符串中只包含字母和\*号。请编写函数 fun, 它的功能是: 使字符串的前导\*号不得多于 n 个; 若多于 n 个, 则删除多余的\*号; 若少于或等于 n 个, 则什么也不做, 字符串中间和尾部的\*号不删除。例如, 字符串的内容为\*\*\*\*\*A\*BC\*DEF\*G\*\*\*\*, 若 n 的值为 4, 删除后, 字符串中的内容应当是\*\*\*\*A\*BC\*DEF\*G\*\*\*\*; 若 n 的值为 8, 则字符串中的

内容仍为\*\*\*\*\*A\*BC\*DEF\*G\*\*\*\*。n 的值在主函数中输入，在编写函数时，不得使用 C 语言提供的字符串函数。

```
#include<stdio.h>
void fun(char *a,int n)
{

}

void main()
{ char s[81];int n;printf("enter a string:\n");gets(s);
printf("enter n:");scanf("%d",&n);
fun(s,n);
printf("the string after deleted:\n");puts(s);}
```

## 实验 16 构造数据类型

### 一、实验目的

1. 掌握结构体类型变量的定义和使用。
2. 掌握结构体类型数组的概念和使用。
3. 掌握链表的概念，初步学会对链表进行操作，学会在函数之间传送链表的方法。
4. 掌握共用体的概念与使用。

### 二、实验内容和步骤

1. 程序通过定义并赋初值的方式，利用结构体变量存储了一名学生的信息。函数 fun 的功能是输出这位学生的信息。在下划线处填入正确的内容，使程序得出正确的结果。

```
#include<stdio.h>
typedef struct
{int num;char name[9];char sex;
struct(int year,month,day;)birthday;
float score[3];}STU;
void show(STU____1____)
{int i;
printf("\n%d%s%c%d-%d-%d",tt.num,tt.name,tt.sex,tt.birthday.year,tt.birthday.month,tt.birthday.day);
for(i=0;i<3;i++)
printf("%5.1f",____2____);
printf("\n");}
void main()
{STU std={1,"Zhanghua",'M',1961,10,8,76.5,78.0,82.0};
printf("\nA student data:\n");
show(____3____);}
```

2. 程序通过定义并赋初值的方式，利用结构体变量存储了一名学生的学号、姓名和三门课的成绩。函数 fun 的功能是将学生的各科成绩都乘以一个系数 a。在下划线处填入正确的内容，使程序得出正确的结果。

```

#include<stdio.h>
typedef struct
{int num;char name[9];float score[3];}STU;
void show(STU tt)
{int i;
printf("%d %s:",tt.num,tt.name);
for(i=0;i<3;i++) printf("%5.1f",tt.score[i]);
printf("\n");}
void modify(____1____ *ss,float a)
{int i;
for(i=0;i<3;i++)
ss-> ____2____ *=a;}
void main()
{STU std={1,"Zhanghua",76.5,8.0,82.0};
float a;
printf("\nThe original number and name and scores:\n");
show(std);
printf("\nInput a number:");scanf("%f",&a);
modify(____3____ ,a);
printf("\nA reault of modifying:\n");
show(std);}

```

3. 一个将指针用于结构数据动态链表操作的程序，请初步熟悉它们的使用方法。

```

#define NULL 0
#define LEN sizeof (stuct student)
stuct student
{ float score;
stuct student *next; }; /*定义结构*/
int n;float x; /*全局变量 n 和 x */
stuct student *creat() /*建立一个链表队列，返回指向链表首结点的指针（地址）*/
{ student *head;stuct student p,r;n=0;
p=r=(stuct student *)malloc(LEN);
scanf("%f",&x);}
r->next=NULL;return (read);}
void print(head) /*遍历一个 head 为指向的链表*/
stuct student *head;
{stuct student *p;p=read;
while (p!=NULL)
{printf("%8.2f",p->score);p=p->next;}
}
void main()
{stuct student *hend
printf("input score:\n");
head=creat(); /*建立单链表*/
print(head); } /*遍历单链表*/

```

4. 给定程序中，函数 fun 的功能是：计算出带有头结点的单向链表中各结点数据域中值之和作为函数值返回。在下划线处填入正确的内容，使程序得出正确的结果。

```

#include<stdio.h>
#include<stdlib.h>
#define N 8
typedef struct list
{int data;struct list *next;}SLIST;
SLIST *creatlist(int *);
void outlist(SLIST *);
int fun(SLIST *h)
{SLIST *p;int s=0;p=h->next;
while(p)
{ s+=p-> 1 ;p=p-> 2 ;}
return s;}
void main()
{SLIST *head;int a[N]={12,87,45,32,91,16,20,48};
head=creatlist(a);outlist(head);
printf("\nsum=%d\n",fun( 3 ));}
SLIST *creatlist(int a[ ])
{SLIST *h,*p,*q;int i;
h=p=(SLIST *)malloc(sizeof(SLIST));
for(i=0;i<N;i++)
{q=( SLIST *)malloc(sizeof(SLIST));
q->data=a[i];p->next=q;p=q;}
p->next=0;
return h;}
void outlist(SLIST *h)
{SLIST *p;p=h->next;
if(p==NULL) printf("The list is NULL!\n");
else {printf("\nhead");
do
{printf("->%d",p->data);p=p->next;}while(p!=NULL);
printf("->End\n");}
}

```

## 实验 17 文件

### 一、实验目的

1. 掌握文件、缓冲文件系统和文件指针的概念。
2. 学会使用文件的打开、关闭、读和写等文件操作函数。
3. 学会用缓冲文件系统对文件进行简单的操作。

### 二、实验内容和步骤

1. 下面为 5 个学生的学习成绩表，从键盘上输入这些数据，计算平均成绩，将原有数据及计算出的平均成绩存放在磁盘文件“stud”中。

学号(num)	姓名(name)	成绩 1(grad1)	成绩 2(grad2)	成绩 3(grad3)
20101	Wangming	85	88.5	96
20103	Lilin	86.5	82.5	95
20109	Zhangming	98	96.5	91.5
20111	Zhaohan	72.5	77.5	89.5
20113	Xujun	61.5	68	88.5

2. 将上题“stud”文件中的学生数据，按平均分（降序）排序处理，将已排序的学生数据存入一个新文件“studsor”中。

3. 将已经排好序的学生成绩文件进行插入处理，要插入学生数据为 20106 Liulei 82.5 91.5 96，要求插入后按平均成绩仍为有序的，插入后建立一个新文件。

4. 给定程序中，函数 fun 的功能是将形参给定的字符串、整数、浮点数写到文本文件中，再用字符方式从此文本文件中逐个读入并显示在终端屏幕上。在下划线处填入正确的内容，使程序得出正确的结果。

```
#include<stdio.h>
void fun(char *s,int a,double f)
{ 1 fp;char ch;
fp=fopen("file1.txt","w");
fprintf(fp,"%s %d %f\n",s,a,f);
fclose(fp);
fp=fopen("file1.txt","r");
printf("\nthe result:\n\n");
ch=fgetc(fp);
while(!feof(2)){
putchar(3);ch=fgetc(fp);}
putchar('\n');
fclose(fp); }
void main()
{ char a[10]="Hello!";
int b=12345;
double c=98.76;
fun(a,b,c); }
```

5. 给定程序中，函数 fun 的功能是：将自然数 1~10 以及它们的平方根写到名为 myfile3.txt 的文本文件中，然后再顺序读出显示在屏幕上。在下划线处填入正确的内容，使程序得出正确的结果。

```
#include<stdio.h>
#include<math.h>
int fun(char *fname)
{ FILE *fp;int i,n;float x;
if((fp=fopen(fname,"w"))==NULL)
return 0;
for(i=1;i<=10;i++)
```

```
fprintf(__1__ , "%d%f\n", i, sqrt((double)i));
printf("\nsucceed!!\n");
__2__ ;
printf("\nthe data in file:\n");
if((fp=fopen(__3__ , "r"))==NULL)
    return 0;
fscanf(fp, "%d%f", &n, &x);
while(!feof(fp))
{ printf("%d%f\n", n, x);
  fscanf(fp, "%d%f", &n, &x);}
fclose(fp);
return 1; }
void main()
{ char fname[ ]="myfile3.txt"
  fun(fname); }
```