题目：解得

#include <stdio.h>

#include <math.h> //程序中调用求平方根函数sqrt

int main()

{

double a,b,c,disc,x1,x2,p,q;

scanf("%lf%lf%lf",&a,&b,&c); //输入双精度浮点型变量的值要用格式声明“%lf”

disc=b\*b-4\*a\*c;

if(disc<0)

{

printf("this equation hasn't real roots\n");

}

else

{

p=-b/(2.0\*a);

q=sqrt(disc)/(2.0\*a);

x1=p+q;

x2=p-q;

printf("real roots:\nx1=%7.2f\nx2=%7.2f\n",x1,x2);

}

}

实际上下面这个程序更为合理：

#include <stdio.h>

#include <math.h>

int main()

{

double a,b,c,disc,x1,x2,realpart,imagpart;

scanf("%lf,%lf,%lf",&a,&b,&c); //%lf的l是字母l

printf("the equation ");

if(fabs(a)<=1e-6) //1e的1是数字1

{

printf("is not a quadratic\n");

}

else

{

disc=b\*b-4\*a\*c;

if(fabs(disc)<=1e-6)

{

printf("has two equal roots:%8.4f\n",-b/(2\*a));

}

else

{

if(disc>1e-6)

{

x1=(-b+sqrt(disc))/(2\*a);

x2=(-b-sqrt(disc))/(2\*a);

printf("has distinct real roots:%8.4f and %8.4f\n",x1,x2);

}

else

{

realpart=-b/(2\*a); // realpart是复根的实部

imagpart=sqrt(-disc)/(2\*a); // imagpart是复根的虚部

printf(" has complex roots:\n");

printf("%8.4f+%8.4fi\n",realpart,imagpart);

printf("%8.4f-%8.4fi\n",realpart,imagpart);

}

}

}

}

程序分析：

对于判断是否等于0时，要注意：由于disc（即）是实数，而实数在计算和存储时会有一些微小的误差，因此不能直接进行如下判断：“if（disc= =0）…”，因为这样可能会出现本来是零的量，由于上述误差而被判别为不等于零而导致结果错误。所以采取的办法是判别disc的绝对值（fabs（disc））是否小于一个很小的数（例如），如果小于此数，就认为disc等于0。

在输出复根时，先分别计算出其实部与虚部，在printf函数的格式字符串中输出虚部的格式声明（%8.4f）后面人为的加上一个普通字符“i”，就能输出“p+qi”这样的复数形式。

题目：任意输入三个数a，b，c，按从小到大的顺序输出。

思路：通过比较，将最小数排在前面。

#include <stdio.h>

#include <math.h>

int main()

{

float a,b,c,t;

scanf("%f,%f,%f",&a,&b,&c);

if(a>b)

{

t=a;

a=b;

b=t;

}

if(a>c)

{

t=a;

a=c;

c=t;

}

if(b>c)

{

t=b;

b=c;

c=t;

}

printf("%5.2f,%5.2f,%5.2f\n",a,b,c);

}

题目：输入一个字符，判别它是否为大写字母，如果是，将它转换成小写字母；如果不是，则不转换。然后输出最后得到的字符。

#include <stdio.h>

#include <math.h>

int main()

{

char ch;

scanf("%c",&ch);

ch=(ch>='A'&&ch<='Z')?ch+32:ch;

printf("%c",ch);

}

题目：要求按照考试成绩的等级输出百分比分数段，a等为80~100分，b等为60~80分，c等为40~60分，d等为20~40分，e等为<20分以下，成绩的等级由键盘输入。

#include <stdio.h>

int main()

{

char grade;

scanf("%c",&grade);

printf("your scoer:");

switch (grade)

{

case 'a':printf("80~100\n");break;

case 'b':printf("60~80\n");break;

case 'c':printf("40~60\n");break;

case 'd':printf("20~40\n");break;

case 'e':printf("<20\n");break;

default:printf("enter data error!\n");

}

}

改为if结构：

#include <stdio.h>

int main()

{

char grade;

scanf("%c",&grade);

printf("your scoer:");

if(grade=='a')

printf("80~100\n");

else if(grade=='b')

printf("60~80\n");

else if(grade=='c')

printf("40~60\n");

else if(grade=='d')

printf("20~40\n");

else if(grade=='e')

printf("<20\n");

}

题目：用switch语句处理菜单命令，如从键盘输入一个“A”，“a”字符，就会执行相应的操作，输入“B”，“b”字符，就会执行另一个操作。

#include <stdio.h>

int main()

{

void action1(int,int),action2(int,int);

char ch;

int a=15,b=23;

ch=getchar();

switch(ch)

{

case'a':

case'A':action1(a,b);break; //调用action1函数，执行A、a操作

case'b':

case'B':action2(a,b);break; //调用action2函数，执行B、b操作

default:putchar('\a'); //输入其他字符，发出声音警告

}

return 0;

}

void action1(int x,int y) //执行加法函数，注：int后的字母，惯例用x，y

{

printf("x+y=%d\n",x+y);

}

void action2(int x,int y) //执行乘法函数

{

printf("x\*y=%d\n",x\*y);

}

从键盘输入一个小于1000的正数，要求输出它的平方根（如平方根不是整数，则输出其整数部分），要求在运行前先检查输入数据是否为小于1000的正数，若不是，则重新输入。

#include <stdio.h>

#include <math.h>

#define M 1000 //符号常量M

int main()

{

int i,k;

printf("请输入一个小于%d的整数i:",M);

scanf("%d",&i);

while (i>M)

{

printf("不符合，请重新输入");

scanf("%d",&i);

k=sqrt(i);

}

k=sqrt(i);

printf("%d的平方根的整数部分是%d\n",i,k);

}

程序分析：为增加灵活性，定义符号常量M为1000，如有需要，只需修改define指令即可，而不必修改主函数。

题目：给出一百分制成绩，要求输出成绩等级‘a’，‘b’，‘c’，‘d’，‘e’。90分以上为‘a’，80~89分为‘b’，70~79分为‘c’，60~69分为‘d’，60分以下为‘e’。

#include <stdio.h>

int main()

{

float score;

char grade;

printf("请输入学生成绩：");

scanf("%f",&score);

while (score>100 || score<0)

{

printf("\n不在规定范围内，请重输");

scanf("%f",&score);

}

switch ((int)(score/10)) //（int）将（score/10）的值进行强制类型转换，得到整型值。

{

case 10:

case 9:grade='a';break;

case 8:grade='b';break;

case 7:grade='c';break;

case 6:grade='d';break;

case 5:

case 4:

case 3:

case 2:

case 1:

case 0:grade='e';

}

printf("成绩是%5.1f，对应的等级是%c\n",score,grade);

}

题目：给出一个不多与5位的正整数，要求：

1. 求出它是几位数；
2. 分别输出每一位数字；
3. 按逆序输出各位数字。

#include <stdio.h>

#include <math.h>

int main()

{

int num,indiv,ten,hundred,thousand,ten\_thousand,place;

printf("请输入一个不多于5位的整数：");

scanf("%d",&num);

if(num>9999)

{

place=5;

}

else if(num>999)

{

place=4;

}

else if(num>99)

{

place=3;

}

else if(num>9)

{

place=2;

}

else

{

place=1;

}

printf("位数:%d\n",place);

printf("每位数字为：");

ten\_thousand=num/10000;

thousand=(int)(num-ten\_thousand\*10000)/1000;

hundred=(int)(num-ten\_thousand\*10000-thousand\*1000)/100;

ten=(int)(num-ten\_thousand\*10000-thousand\*1000-hundred\*100)/10;

indiv=(int)(num-ten\_thousand\*10000-thousand\*1000-hundred\*100-ten\*10);

switch (place)

{

case 5:printf("%d,%d,%d,%d,%d",ten\_thousand,thousand,hundred,ten,indiv);

printf("\n反序数字为：");

printf("%d,%d,%d,%d,%d\n",indiv,ten,hundred,thousand,ten\_thousand);

break;

case 4:printf("%d,%d,%d,%d",thousand,hundred,ten,indiv);

printf("\n反序数字为：");

printf("%d,%d,%d,%d\n",indiv,ten,hundred,thousand);

break;

case 3:printf("%d,%d,%d",hundred,ten,indiv);

printf("\n反序数字为：");

printf("%d,%d,%d\n",indiv,ten,hundred);

break;

case 2:printf("%d,%d",ten,indiv);

printf("\n反序数字为：");

printf("%d,%d\n",indiv,ten);

break;

case 1:printf("%d",indiv);

printf("\n反序数字为：");

printf("%d\n",indiv);

break;

}

}