1. 算术表达式

#include<stdio.h>

void main()

{int x,y,z;

float a,b,c;

x=3\*2;

x=3/4;//同类型运算，结果是同类型，异种类型运算，结果以高为准

//char<<int<<float<<double

x=3.0/4 //产生新问题，关注

x=192%10;//取余符号

x=2.6%2;//编译错，计算机要求，取余的两边只能是整数

}

1. 增量表达式

#include<stdio.h>

void main()

{int a,b,c;

a=3;

b=a++;//a++，取a当前的旧值，作为表达式的值，给b,a再加，对下次有影响

printf("%d,%d\n",a,b); //4,3

a=3;

b=++a;//++a，将a加1，有新值，4，表达式值给b,a已经加过，不再加了

printf("%d,%d\n",a,b); //4,4

a=3;

b=(++a)+9;

printf("%d,%d\n",a,b); //4,13

}

特殊表达式：赋值表达式

#include<stdio.h>

void main()

{int x,y=4,z;

z=(x=4)+9;

}

#include<stdio.h>

void main()

{int x,y=4,z;

x=y=z=9;

printf("%d,%d,%d",x,y,z);

}

1. 关系运算符

>,<,<=,>=,!=,==

printf("%d",4>1);//成立，计算机用1来表达真，0表达假

#include<stdio.h>

void main()

{ int x;

printf("%d",(3>9)+8+(x=2));//10

}

补充1

#include<stdio.h>

void main()

{ int x=9;

if (8) printf("yes");//输出yes,计算机将所有的非零都能识别为真

else printf("no");

}

补充2：

#include<stdio.h>

void main()

{ int x=9;

if (x==8) printf("yes");//输出NO

else printf("no");

}

补充3：

#include<stdio.h>

void main()

{ int x=9;

if (x=8) printf("yes");//输出yes,计算机将所有的非零都能识别为真

else printf("no");

}

补充难点：

#include<stdio.h>

void main()

{ int x=9;

if (12<x<13) printf("yes");//输出yes,计算机将所有的非零都能识别为真

else printf("no");

printf("\n%d",12<9);//0

}

逻辑表达式：

情境1：

#include<stdio.h>

void main()

{ int x=9,y=16;

if (12>x&&(y=3)) printf("yes,%d",y);//输出yes,3

else printf("no,%d",y);

}

情境2：短路原理1：

#include<stdio.h>

void main()

{ int x=9,y=16;

if (12<x&&(y=3)) printf("yes,%d",y);

else printf("no,%d",y);//no,16

}

情境3：

#include<stdio.h>

void main()

{ int x=9,y=16;

if (12<x||(y=3)) printf("yes,%d",y);//yes,3

else printf("no,%d",y);

}

情境4：

#include<stdio.h>

void main()

{ int x=9,y=16;

if (12>x||(y=3)) printf("yes,%d",y);//yes,16

else printf("no,%d",y);

}

次重点的表达式：

1. 条件表达式？

#include<stdio.h>

void main()

{ int a=4,b,c;

b=(a>3)?2:1;

//如果(a>3)成立的话，取2作为右边表达式的值

//如果(a>3)不成立的话，取1作为右边表达式的值

}

补充：

#include<stdio.h>

void main()

{ char ch;

scanf("%c",&ch);

ch=(ch>='a'&&ch<='z')?ch-32:ch;

printf("%c",ch);//功能：若输入小写字母，变大写。其他符号不变

}

2.逗号表达式

#include<stdio.h>

void main()

{ int a,b,c;

a=(b=2,b++,b+3); //由左往左，取逗号表达式的最后一个，作为整个的值

printf("%d\n",a);//6

a=(b=13,b+9,17);

printf("%d\n",a);//17

}

二、语句：

1.字符输入和输出

#include<stdio.h>

void main()

{ char ch;

ch=getchar();

putchar(ch);

putchar(getchar());

}

2.格式化的输入和输出

#include<stdio.h>

void main()

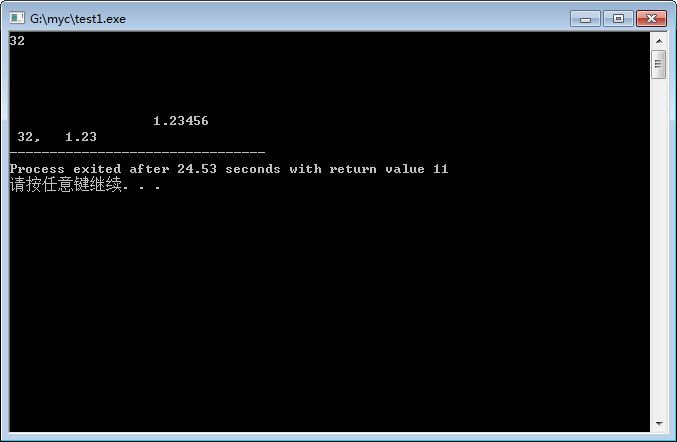
{ int a;

float b;

scanf("%d%f",&a,&b);//格式要求，必须加&取地址符号，

//强烈建议引号中不要加其他符号如,或是提示

printf("%3d,%7.2f",a,b);

}

3.if语句

情境1：

#include<stdio.h>

void main()

{ float s;

//s=64;此句多余

scanf("%f",&s);

if (s>=60)//条件，格式要求必须加括号

printf("成绩合格");

}

情境2：

#include<stdio.h>

void main()

{ float s;

//s=64;此句多余

scanf("%f",&s);

if (s>=60)//条件，格式要求必须加括号

printf("成绩合格");

else printf("成绩不合格");//else部分可以省

}

分析如下程序：

#include<stdio.h>

void main()

{ float s;

//s=64;此句多余

scanf("%f",&s);

if (s>=60)//条件，格式要求必须加括号

printf("成绩合格");

printf("成绩不合格");

}

情境3：

#include<stdio.h>

void main()

{ float s;

//s=64;此句多余

scanf("%f",&s);

if (s>=60)//条件，格式要求必须加括号

{

printf("成绩合格\n");

printf("不错，再加油\n");}

else

{

printf("成绩不合格\n");

printf("还需要努力\n");}

}

补充1：

#include<stdio.h>

void main()

{ float s;

//s=64;此句多余

scanf("%f",&s);

if (s>=60)//条件，格式要求必须加括号

printf("成绩合格\n");

printf("不错，再加油\n");**//语法错，**

else

{

printf("成绩不合格\n");

printf("还需要努力\n");}

}

补充2：

#include<stdio.h>

void main()

{ float s;

//s=64;此句多余

scanf("%f",&s);

if (s>=60)  **;//编译错，此外不能有分号**

printf("成绩合格\n");

else

printf("成绩不合格\n");

}

最后，语句格式：

#include<stdio.h>

void main()

{ int a,b,c;

a=3;b=4;c=5;

}