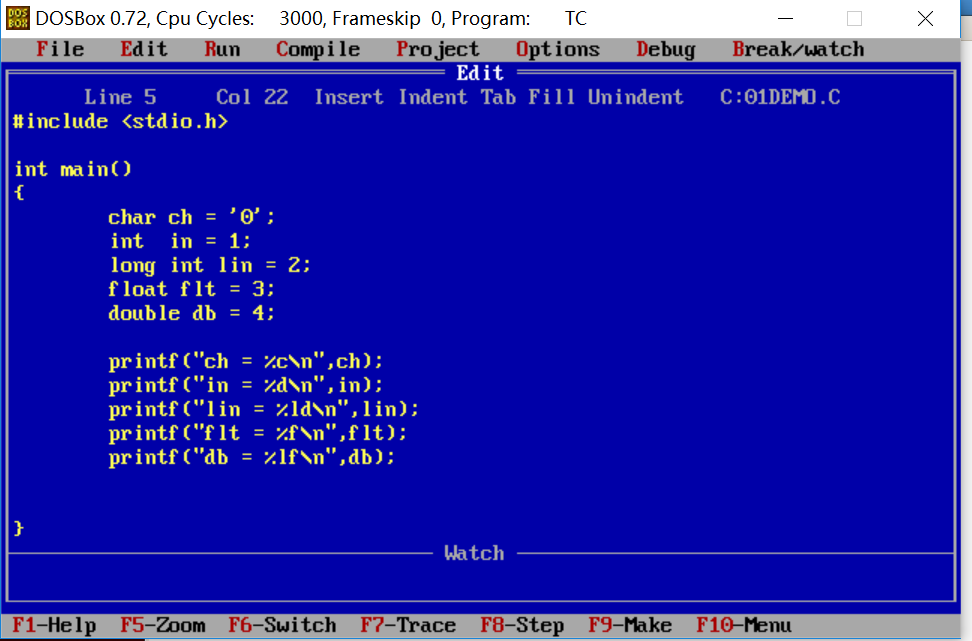
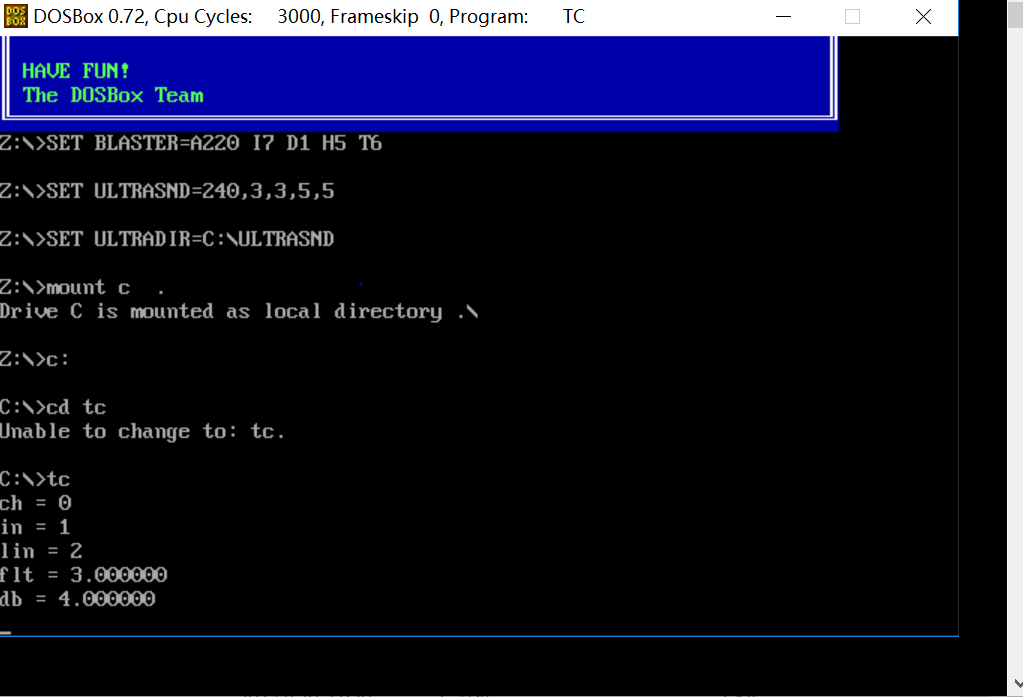
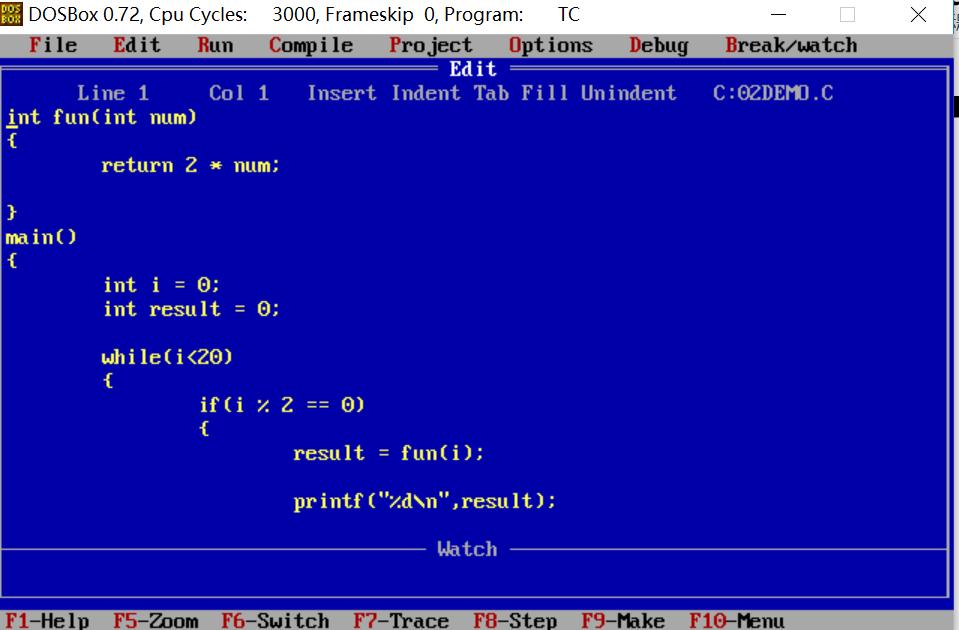
**三个一工程C语言基础复习题**

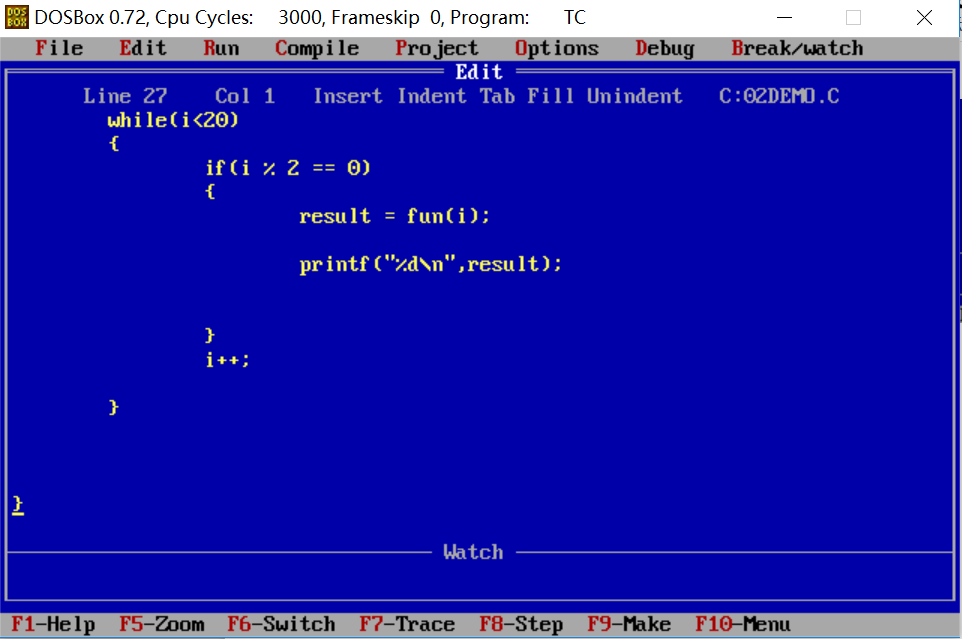
1.

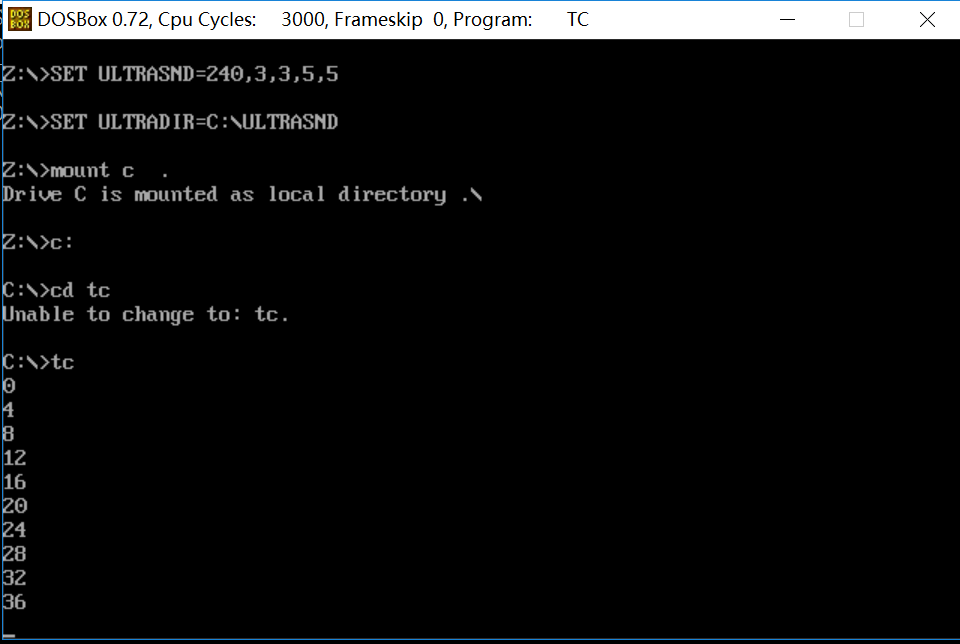


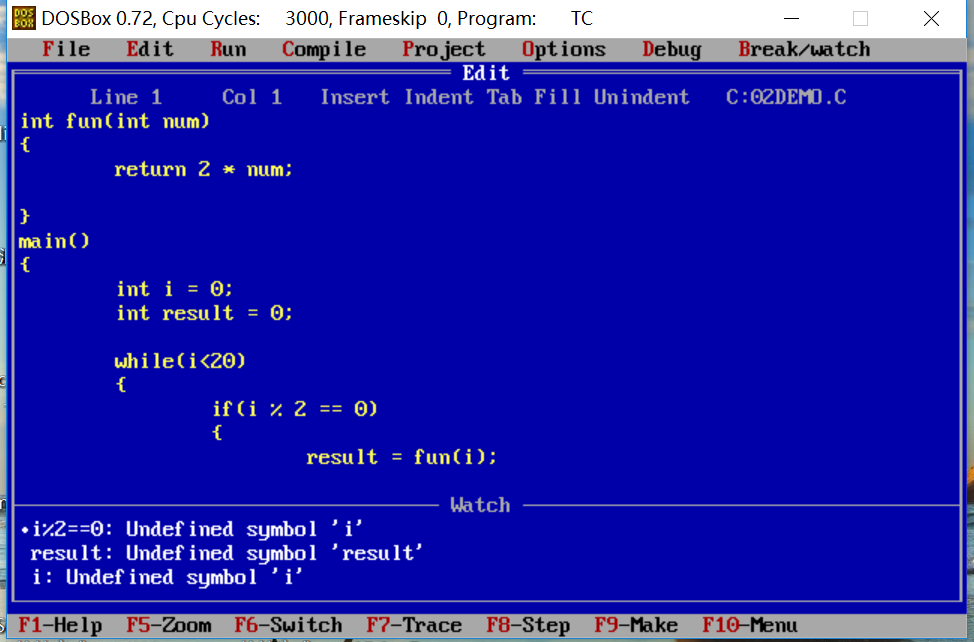


2.

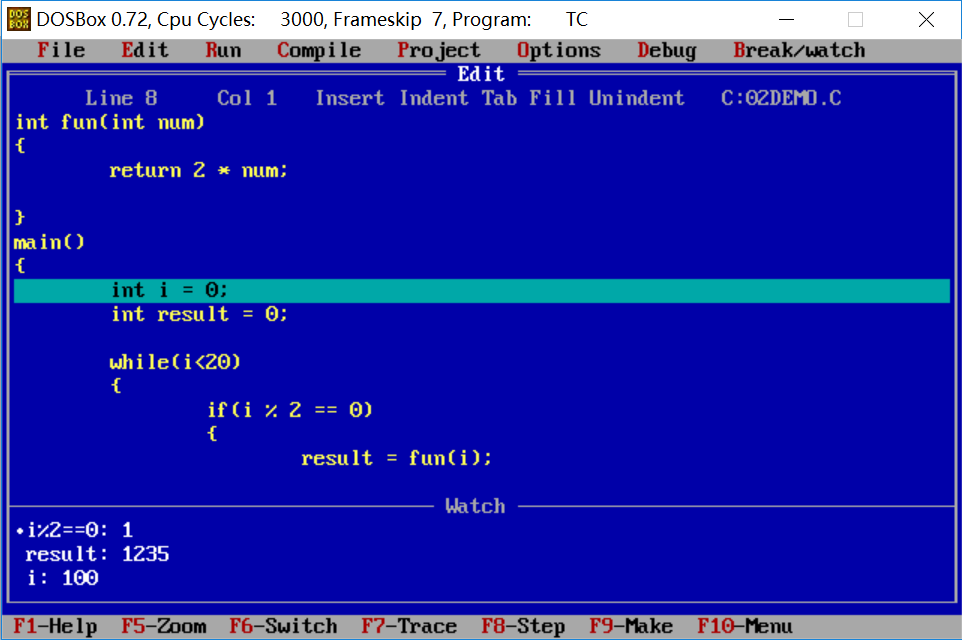




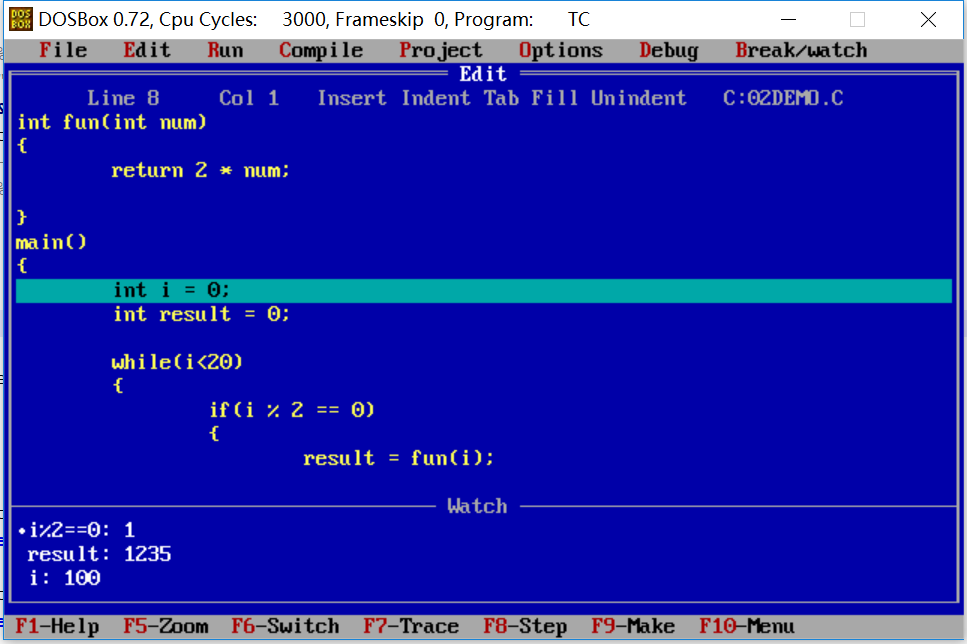




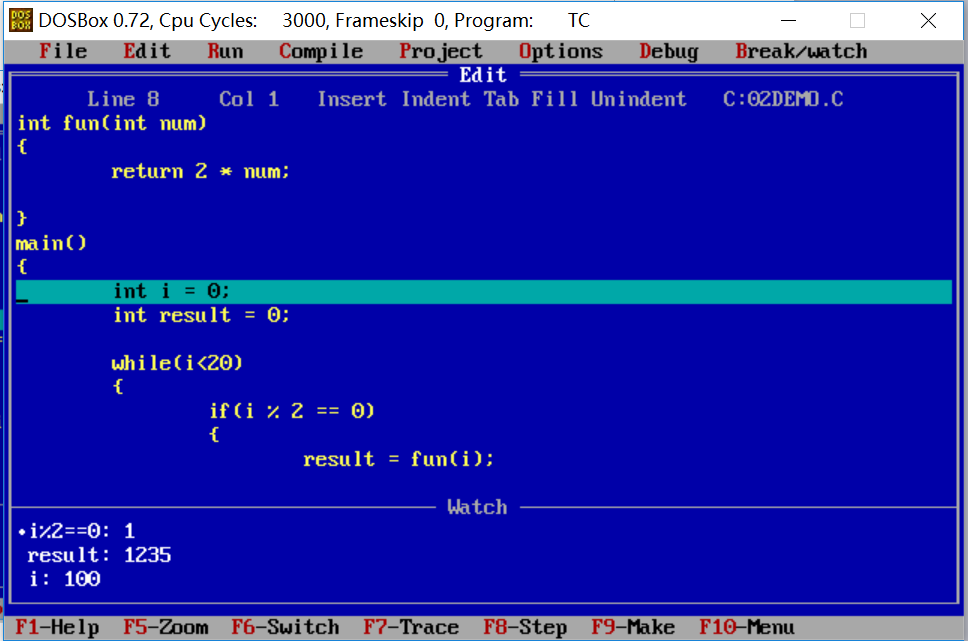
第一步就不解，明明还没赋值，这些表达式的值是如何千奇百怪地显示出来的呢？这些值是固定的吗？待会儿至少单步调试三次看看就知道了。



第二次：

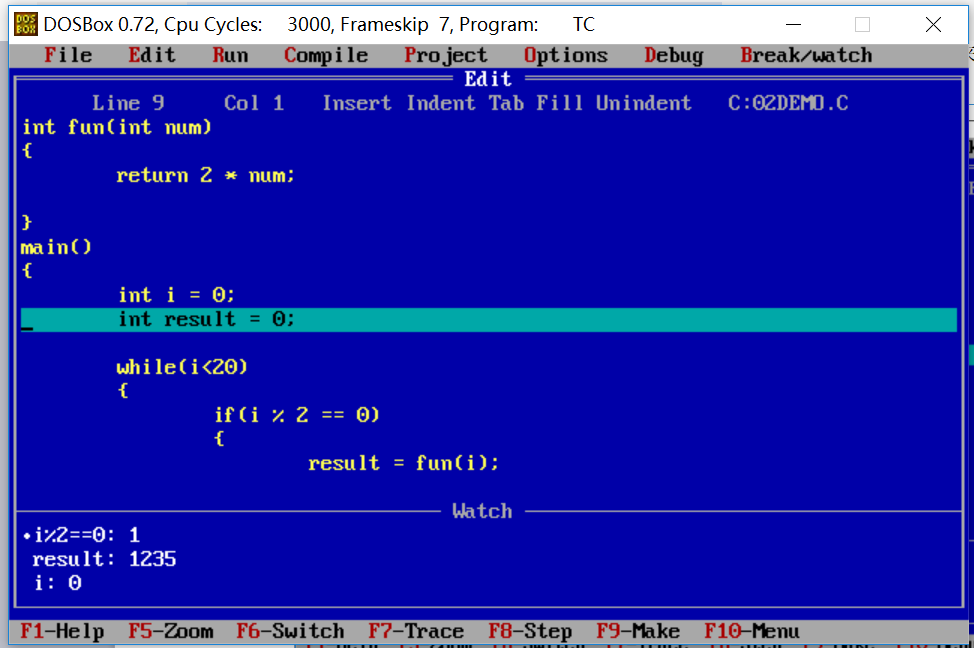


第三次：

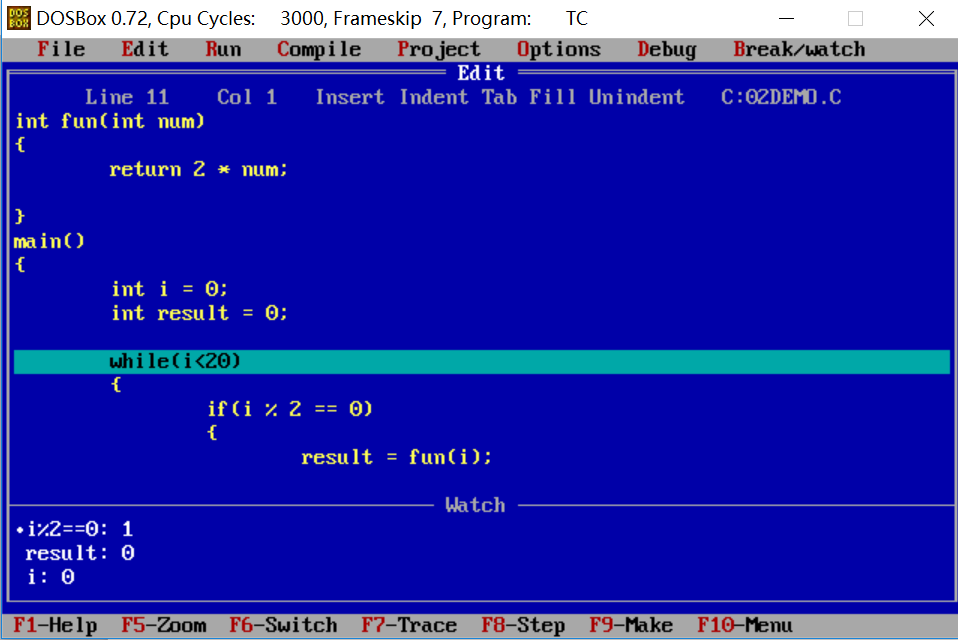


三次都一样，编译器比较固执？

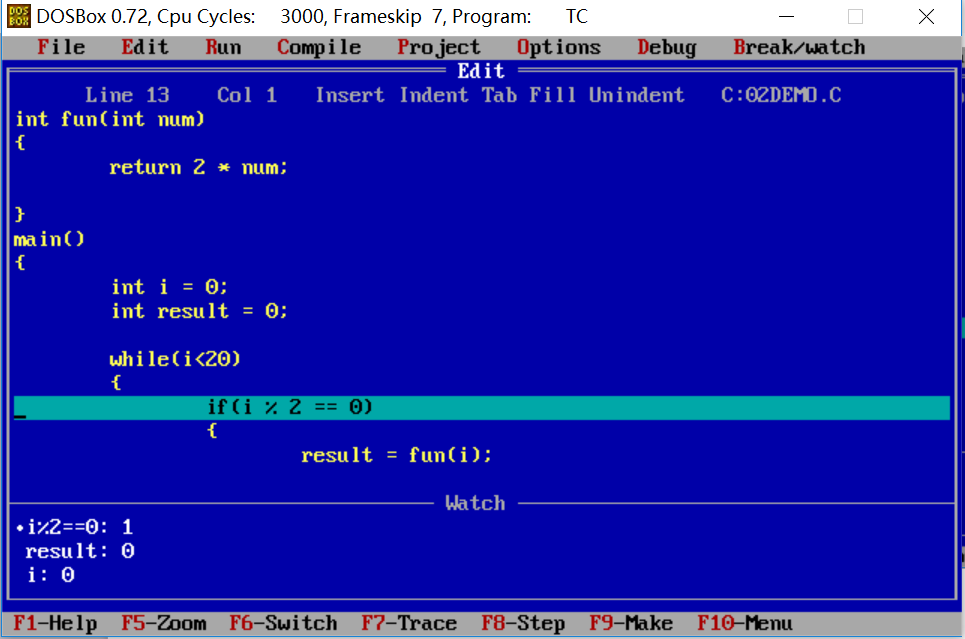
；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；



；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；



；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；



；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；；

在这个过程中表达式“i%2 == 0”的值在“0”和“1”之间交替，而值的情况就是当表达式成立，即为TRUE时，为“1”值；当表达式不成立，即为FALSE时，为“0”值。

3.

1>.”if...else”语句：

#include <stdio.h>

int main()

{

short results[] = {55,80,71,72,87,90,95,100,96,87,

76,100,50,83,80,65,70,69,80,88,

91,98,96,76,89

};

int count1 = 0;

int count2 = 0;

int count3 = 0;

int count4 = 0;

int count5 = 0;

int xu = 0;

for( xu = 0 ; xu < 25 ; xu++ )

{

if(90 <= results[xu] && results[xu] <= 100)

{

count1++;

}

else if(80 <= results[xu] && results[xu] < 90)

{

count2++;

}

else if(70 <= results[xu] && results[xu] < 80)

{

count3++;

}

else if(60 <= results[xu] && results[xu] < 70)

{

count4++;

}

else

{

count5++;

}

}

printf("A : %d\n",count1);

printf("B : %d\n",count2);

printf("C : %d\n",count3);

printf("D : %d\n",count4);

printf("E : %d\n",count5);

return 0;

}

2>.”switch...case”语句：

#include <stdio.h>

int main()

{

short results[] = {55,80,71,72,87,90,95,100,96,87,

76,100,50,83,80,65,70,69,80,88,

91,98,96,76,89

};

int count1 = 0;

int count2 = 0;

int count3 = 0;

int count4 = 0;

int count5 = 0;

int xu = 0;

for( xu = 0 ; xu < 25 ; xu++ )

{

switch( results[xu] /10 )

{

case 10:

case 9:

count1++;

break;

case 8:

count2++;

break;

case 7:

count3++;

break;

case 6:

count4++;

break;

default:

count5++;

break;

}

}

printf("A : %d\n",count1);

printf("B : %d\n",count2);

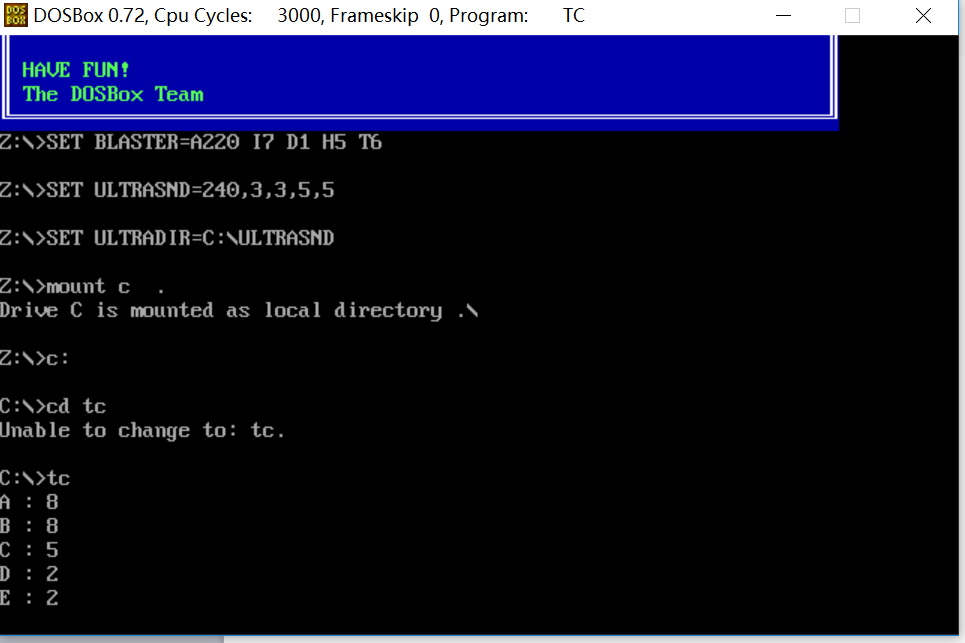
printf("C : %d\n",count3);

printf("D : %d\n",count4);

printf("E : %d\n",count5);

return 0;

}



4.

1>.”for循环”语句：

#include <stdio.h>

int main()

{

int chengshu1 = 0;

int chengshu2 = 0;

printf("九九乘法表：\n");

for( chengshu1 = 1 ; chengshu1 < 10 ; chengshu1++)

{

for(chengshu2 = 1 ; chengshu2 <= chengshu1 ; chengshu2++)

{

printf("%d x %d = %d ",chengshu1,chengshu2,chengshu1\*chengshu2);

}

printf("\n");

}

return 0;

}

2>.”while 循环”语句：

#include <stdio.h>

int main()

{

int chengshu1 = 1;

int chengshu2 = 1;

printf("九九乘法表：\n");

while(chengshu1 < 10)

{

while(chengshu2 <= chengshu1 )

{

printf("%d x %d = %d ",chengshu1,chengshu2,chengshu1\*chengshu2);

chengshu2++;

}

printf("\n");

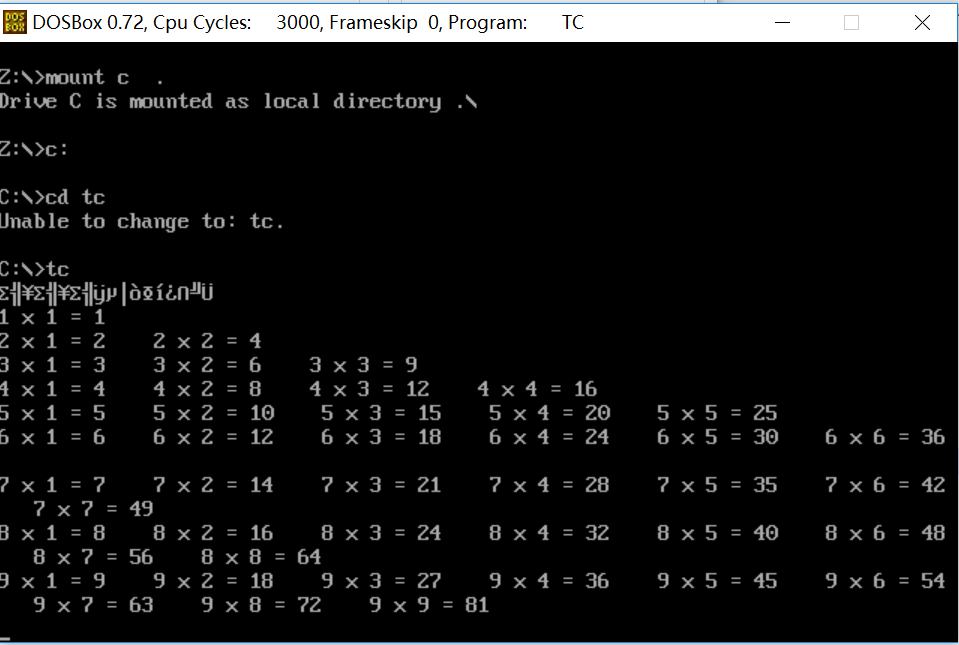
chengshu2 = 1;

chengshu1++;

}

return 0;

}



5.

1>.

int count = 0;

xunhuan:

if(count < 10)

{

printf("%d",count);

count++;

goto xunhuan;

}

2>.

int count = 0;

xunhuan:

printf("%d",count);

count++;

if(count < 10)

{

goto xunhuan;

}

6.

int fun1(int);

int fun2(int);

main()

{

int c = 0;

for( c = 1 ; c <= 9 ; c++)

{

fun1(c);

printf("%d",c);

}

printf("\n");

fun2(c);

}

int fun1(int x)

{

return x+0x30;

}

int fun2(int x)

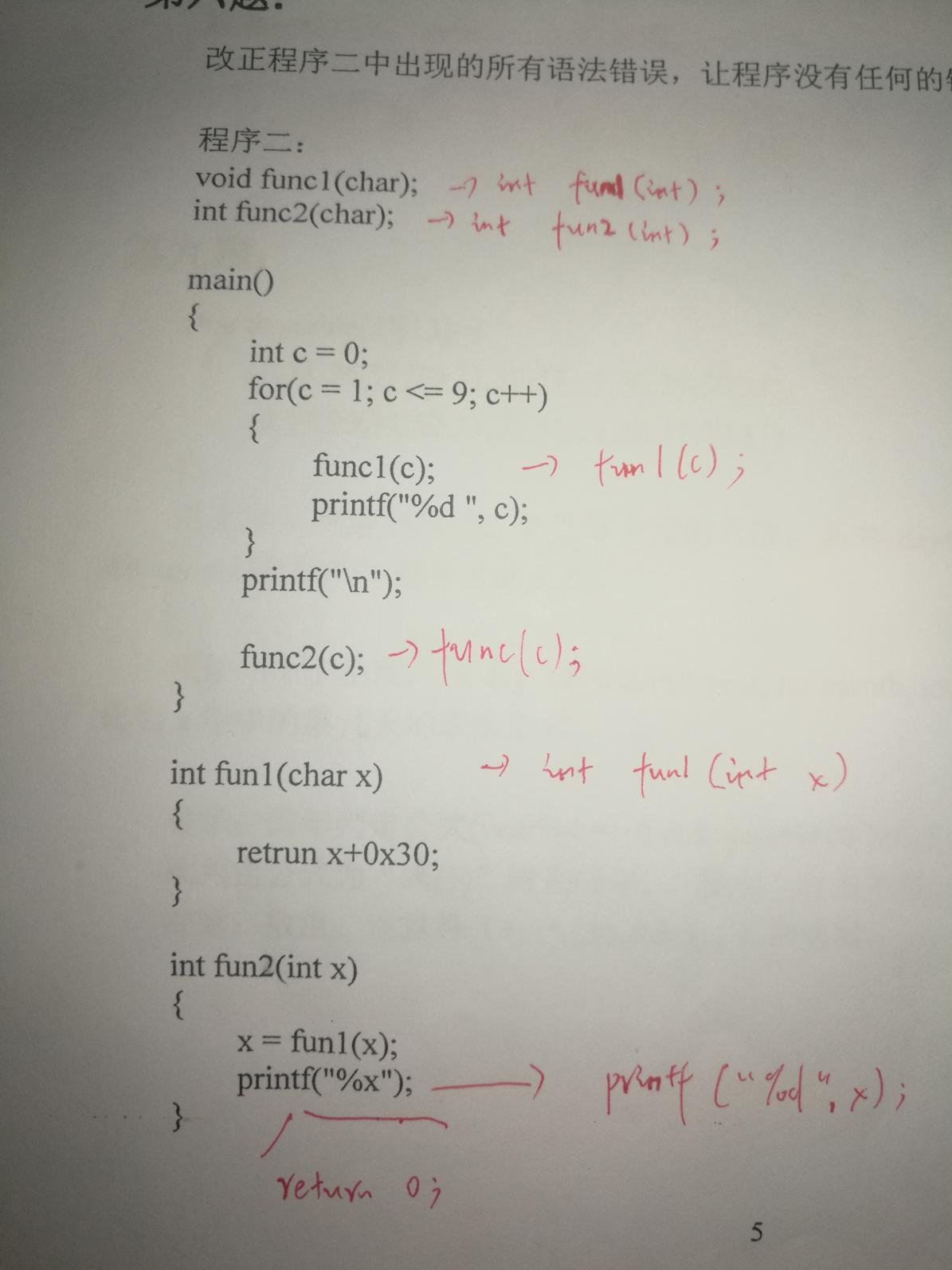
{

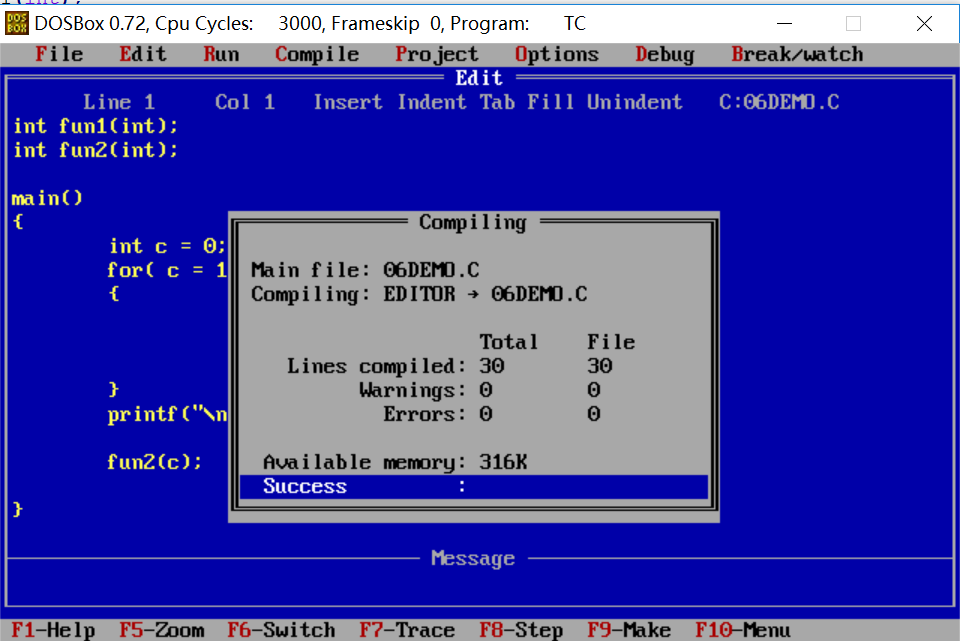
x=fun1(x);

printf("%d",x);

return 0;

}





7.

void exchange(int a,int b)

{

int temp;

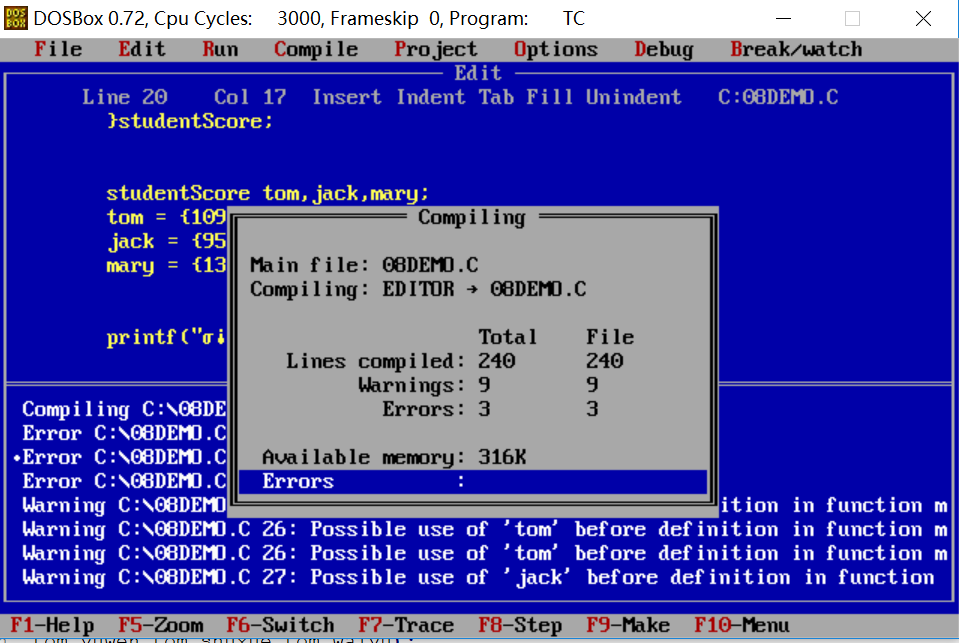
temp = a;

a = b;

b = temp;

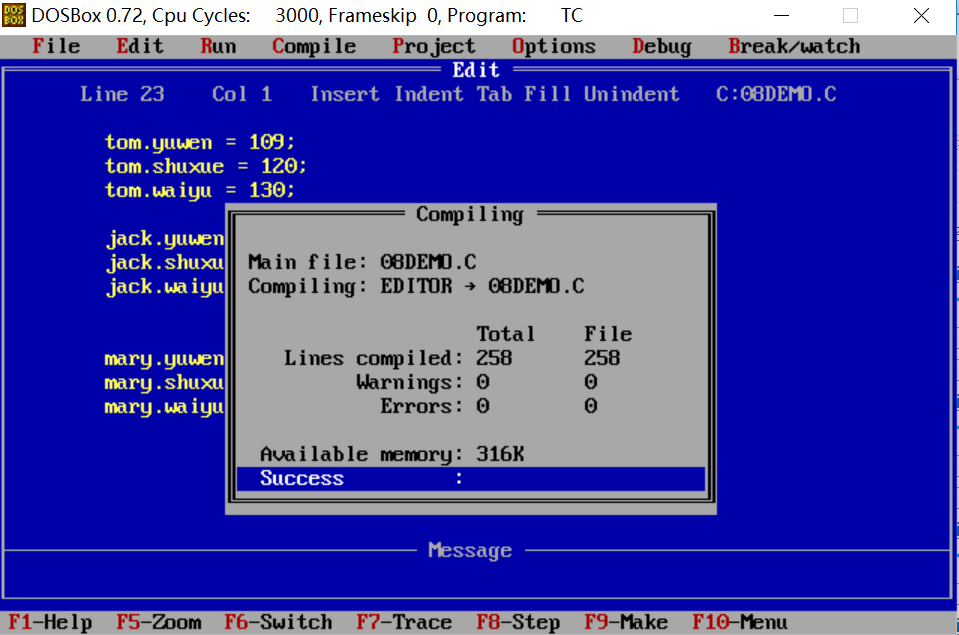
}

8.

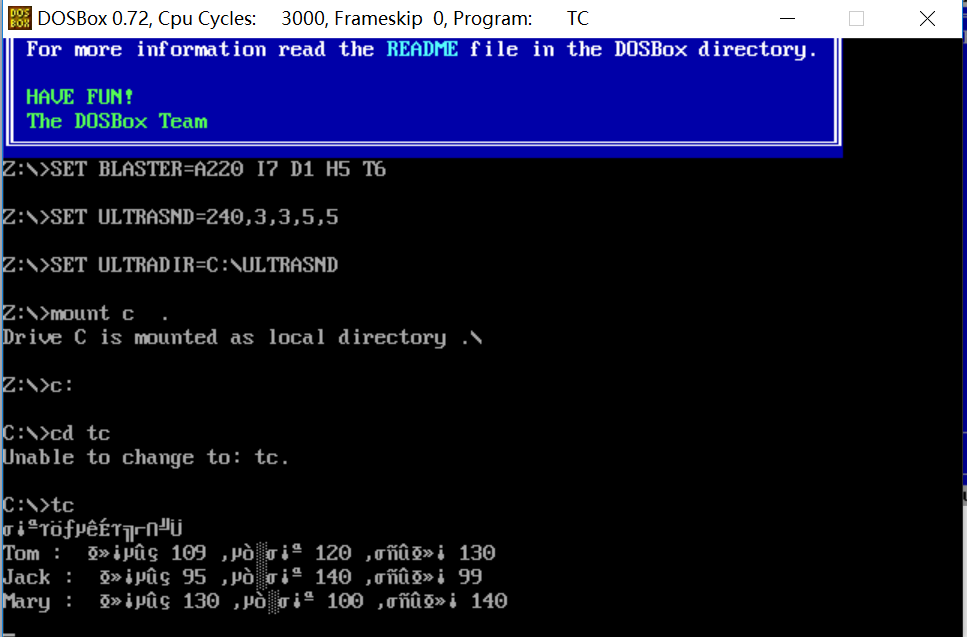


编译失败；

改写赋值结构体变量方式：



编译成功，说明turboc不支持第一种结构体变量赋值的方式。



此处，乱码现象是因为不支持汉字 。

#include <stdio.h>

int main()

{

typedef struct

{

int yuwen;

int shuxue;

int waiyu;

}studentScore;

studentScore tom,jack,mary;

tom.yuwen = 109;

tom.shuxue = 120;

tom.waiyu = 130;

jack.yuwen = 95;

jack.shuxue = 140;

jack.waiyu = 99;

mary.yuwen = 130;

mary.shuxue = 100;

mary.waiyu = 140;

printf("学生成绩： \n");

printf("Tom : 语文 %d ,数学 %d ,外语 %d\n",tom.yuwen,tom.shuxue,tom.waiyu);

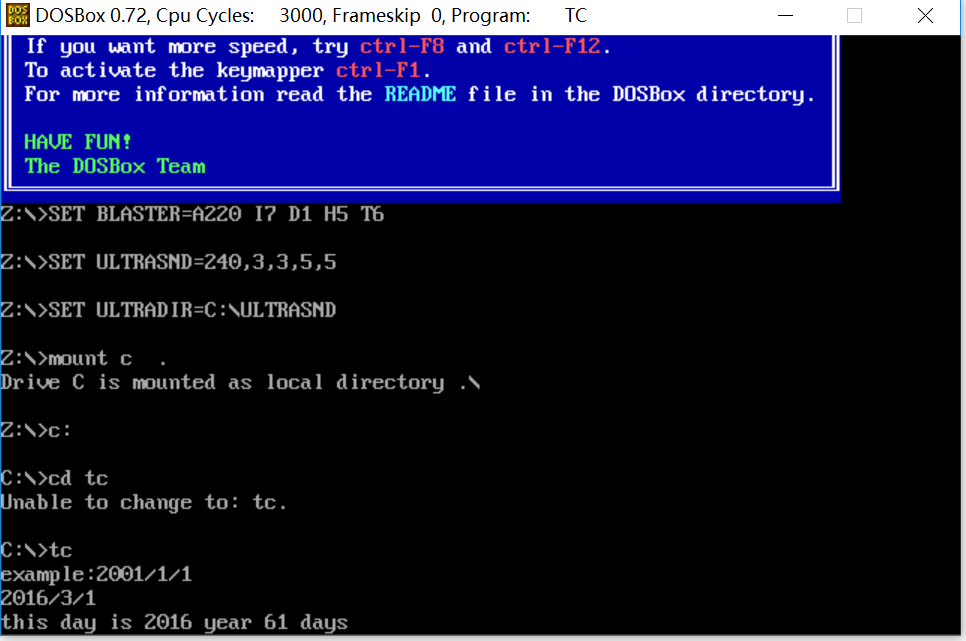
printf("Jack : 语文 %d ,数学 %d ,外语 %d\n",jack.yuwen,jack.shuxue,jack.waiyu);

printf("Mary : 语文 %d ,数学 %d ,外语 %d\n",mary.yuwen,mary.shuxue,mary.waiyu);

}

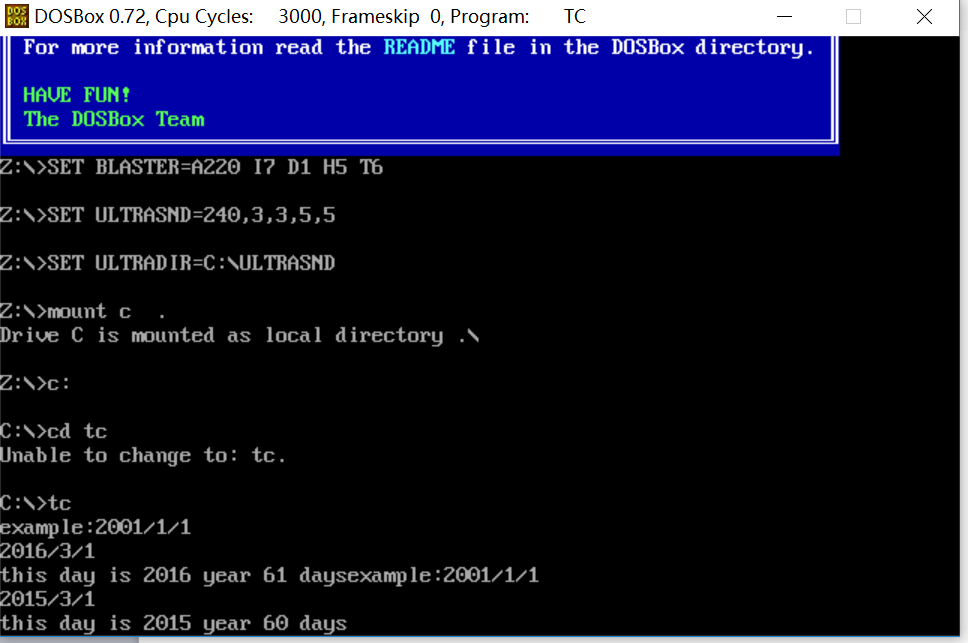
9.

测试：2016/3/1



结果计算正确。

测试：2015/3/1



结果计算正确。

程序：

#include <stdio.h>

#define TRUE 1

#define FALSE 0

char daytable[2][13]={

{0,31,28,31,30,31,30,31,31,30,31,30,31},

{0,31,29,31,30,31,30,31,31,30,31,30,31}

};

int year=0,mon=0,day=0;

int day\_of\_year(int year,int month,int day)

{

int count = 0;

int xu=1;

if(FALSE == ((year%4 == 0&& year%100!=0)||(year%400 == 0) ) )

{

for(xu = 1;xu< month;xu++)

{

count += daytable[0][xu];

}

count += day;

}

else

{

for(xu = 1;xu< month;xu++)

{

count += daytable[1][xu];

}

count += day;

}

return count;

}

int main()

{

printf("example:2001/1/1\n");

scanf("%d/%d/%d",&year,&mon,&day);

printf("this day is %d year %d days",year,day\_of\_year(year,mon,day));

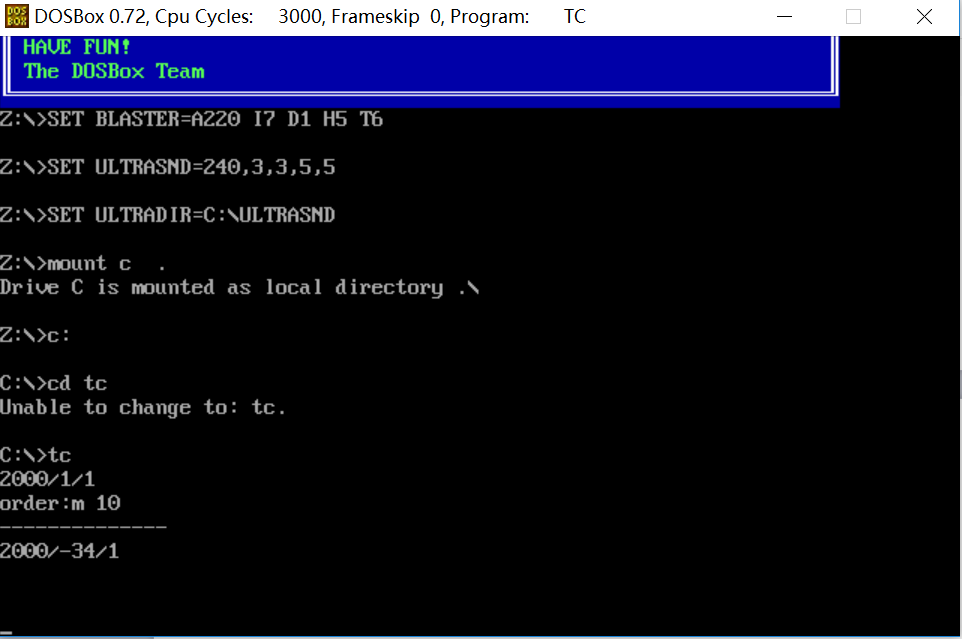
return 0;

}

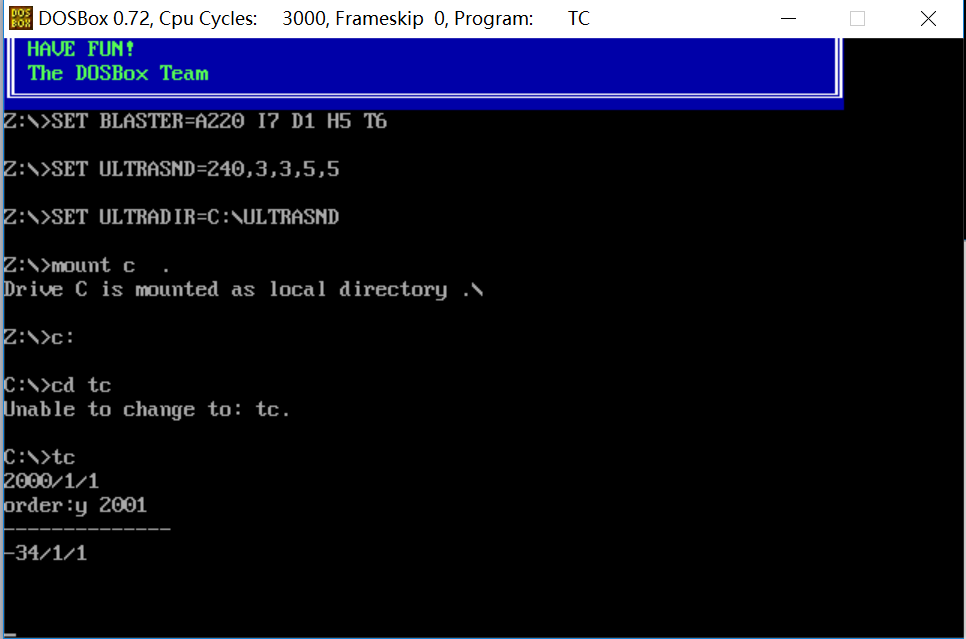
10.

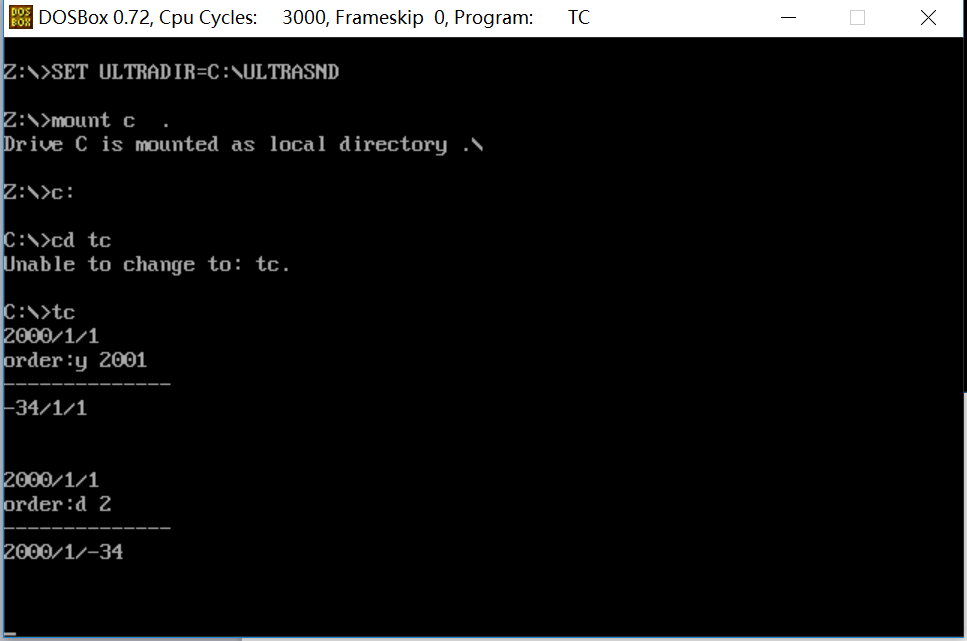
getch();函数常用于中途暂停程序方便调试和查看。

先整体运行一遍：



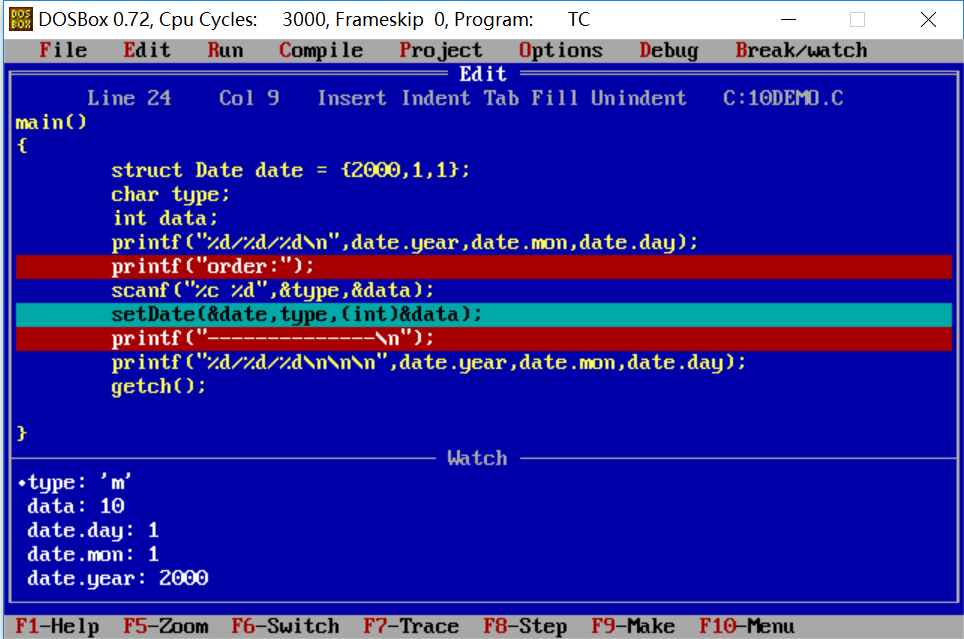
可得，月份修改不符合预期，分别试着修改年与日：





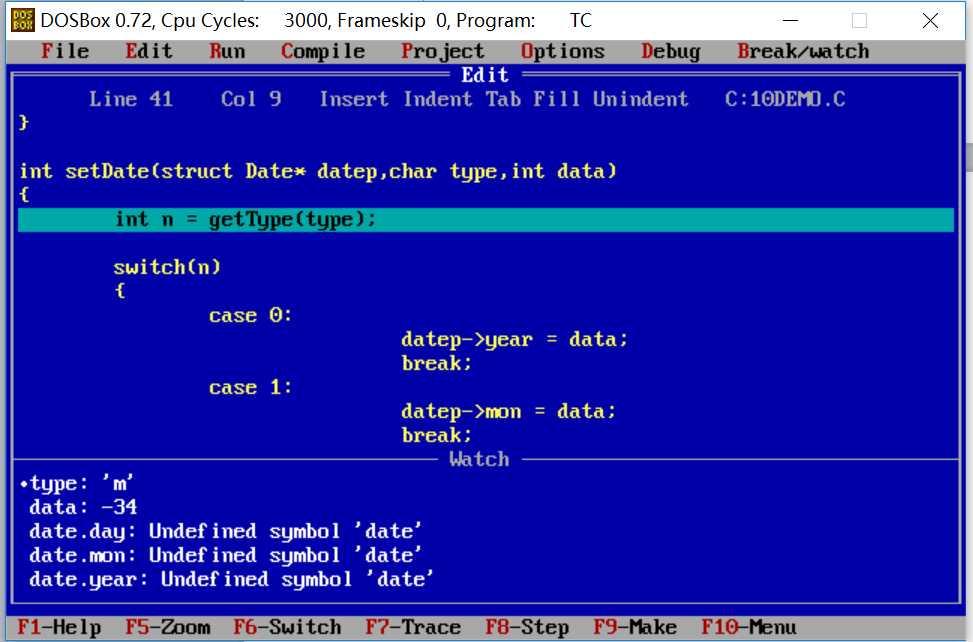
额。。。看来与“-34”很有缘分啊。

但是，观察之后，发现“更改相对应的位置”这一功能是可以实现的，但是具体的更改的数却是显示不正确的固定的数“-34”。



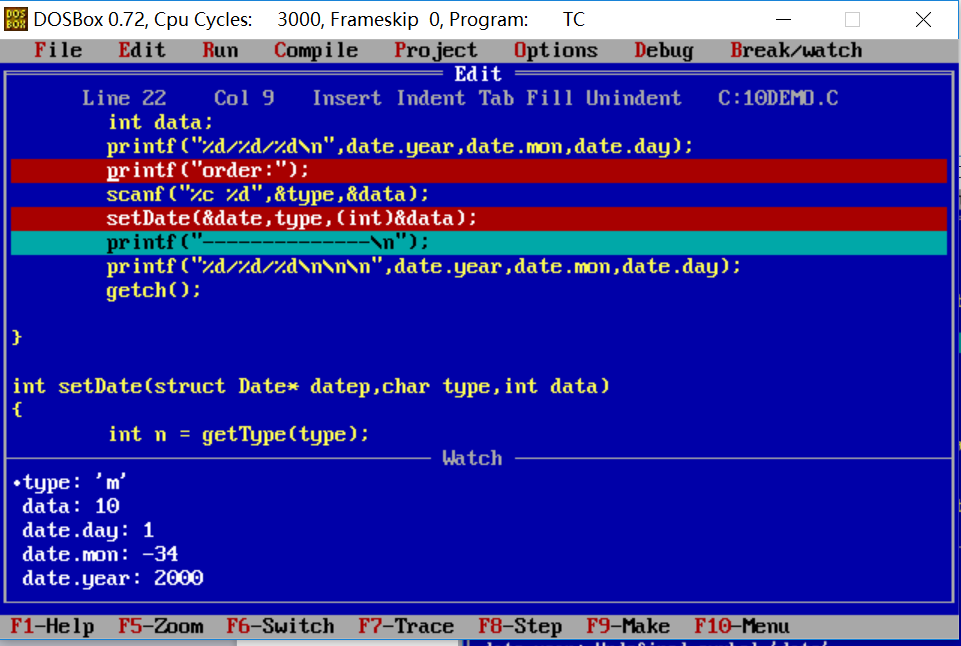
由此可见，字符‘m’顺利传入变量type中，整形数“10”顺利传入变量data中；

进入setDate（）看一下：



注意，这里data的值发生了“变化”，变为了熟悉的“-34”；

本来这里如果还不出现错误现象的话，我会决定先不进getType()继续单步下去，整体运行一下getType()看是否出现了问题,查看变量值的变化是否符合我的预期。但是这里出现“错误”了，所以再回看一下。



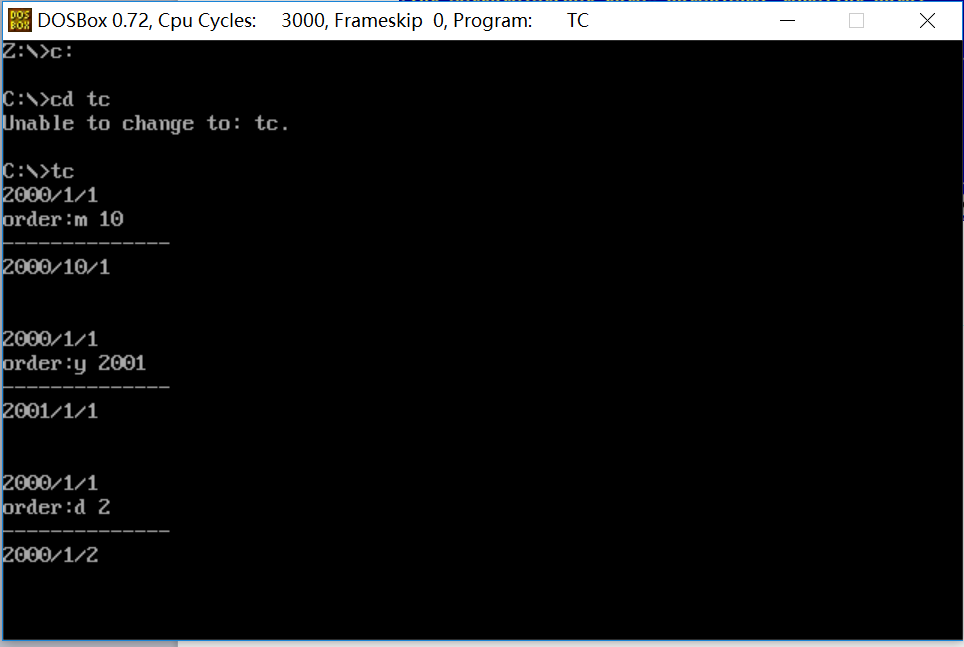
这里，可以看出data变量在setDate()的外面都是“10”，而一到了里面，就变为了“-34”。

那么问题来了，为什么这里会变，怎么让它不变呢？

仔细观察一下这一句“setDate(&date,type,(int)&data);”

额，第三个传递变量和程序原意图不一样呀，在这里，我改了一下：

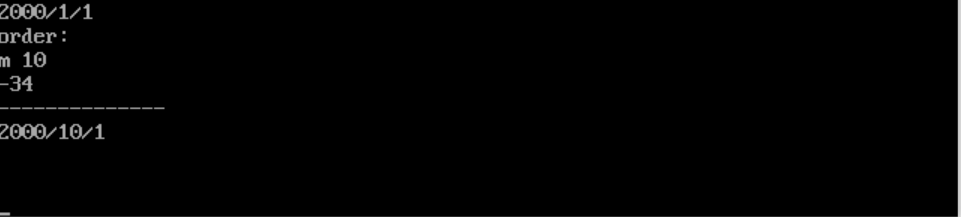
“setDate(&date,type,data);”



年月日都可以正经修改了。

于是，额。。。就这样结束了。

最后单独查看一下这个熟悉的“-34”到底是不是调试推倒出来的“（int）&data”：



结果，真的是这样。