

2020/04/01_第3课_C语言基本数据类型

笔记本: C

创建时间: 2020/4/1 星期三 20:05

作者: ileemi

- [课前回顾](#)
- [VC++6.0 bug](#)
- [C语言基本数据类型的使用](#)
 - [浮点数在内存中的存储方式](#)
 - [字符串](#)

课前回顾

习题讲解:

```
/*  
输出r如下七行空心菱形
```

y轴

```
6          *  
5      *      *  
4  *          *  
3*              *  
2  *          *  
1      *      *  
0          *  
  0    1    2    3    4    5    6    x轴
```

解题思路: 看成直角坐标系, 求出菱形四条边对应的方程公式, 四条直线, 满足条件打印 *, 反之打印空格

取两个点的坐标(需要是一条直线), 求出对应的方程式, 需要求出四组, 例如:

需满足方程式: $ax + b = y$;

```
*/
```

```
/*
```

$$0a + b = 3$$

$$3a + b = 6$$

$$b = 3$$

$$a = 1$$

$$x + 3 == y$$

$$3a + b = 6$$

$$6a + b = 3$$

$$3a = -3$$

$$a = -1$$

$$b = 9$$

$$-x + 9 == y$$

$$6a + b = 3$$

$$3a + b = 0$$

$$3a = 3$$

$$a = 1$$

$$b = -3$$

$$x - 3 == y$$

$$3a + b = 0$$

$$0a + b = 3$$

$$b = 3 \quad a = -1$$

$$-x + 3 == y$$

*/

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    /*
```

```
    普通版
```

```
    */
```

```
    //int nTemp;
```

```
    //int nCount = 3;
```

```
    //for (int i = 0; i < 7; i++)
```

```
    //{
```

```
    //    if (i > nCount)
```

```
    //    {
```

```
    //        nTemp = 6 - i;
```

```
    //    }
```

```
    //    else
```

```
    //    {
```

```
    //        nTemp = i;
```

```
    //    }
```

```
    //    for (int j = nCount - nTemp; j > 0; j--)
```

```
    //    {
```

```
    //        printf(" ");
```

```
    //    }
```

```
    //    printf("*");
```

```
    //    if (nTemp)
```

```
    //    {
```

```
    //        for (int k = 2 * nTemp - 1; k > 0; k--)
```

```
    //        {
```

```
    //            printf(" ");
```

```
    //        }
```

```
    //    printf("*");
```

```

        //      }
        //      printf("\r\n");
        //}
        //数学公式版
for (int x = 0; x < 7; x++)
{
    for (int y = 0; y < 7; y++)
    {
        if (x + 3 == y || -x + 9 == y || x - 3 == y || -x + 3 == y)
        {
            printf("*");
        }
        else
        {
            printf(" ");
        }
    }
    printf("\r\n");
}

    return 0;
}

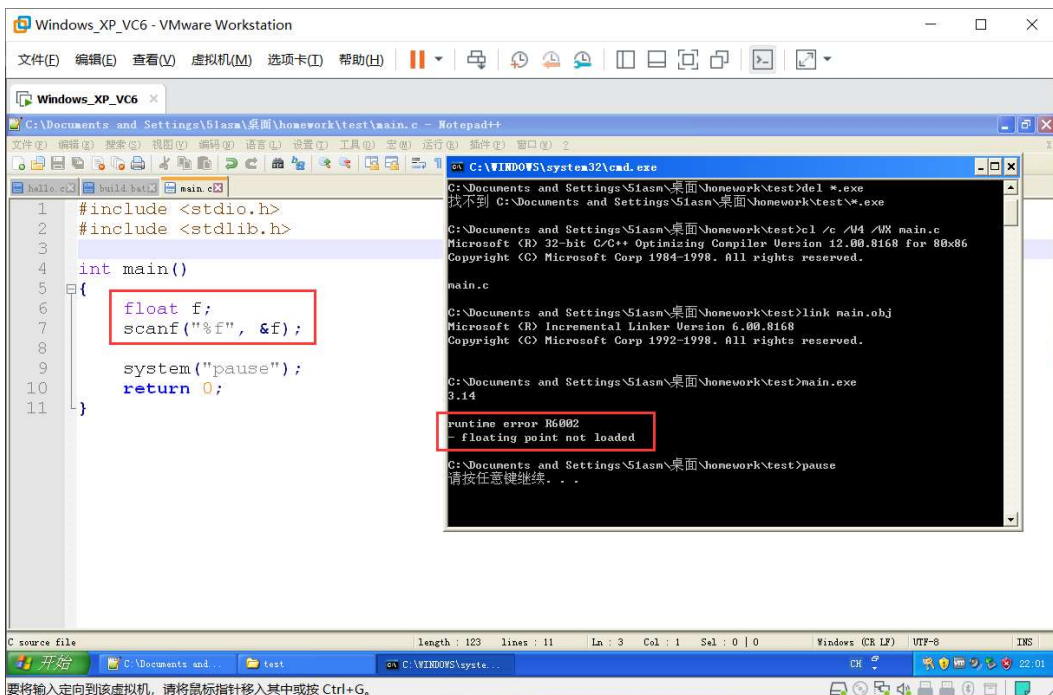
```

VC++6.0 bug

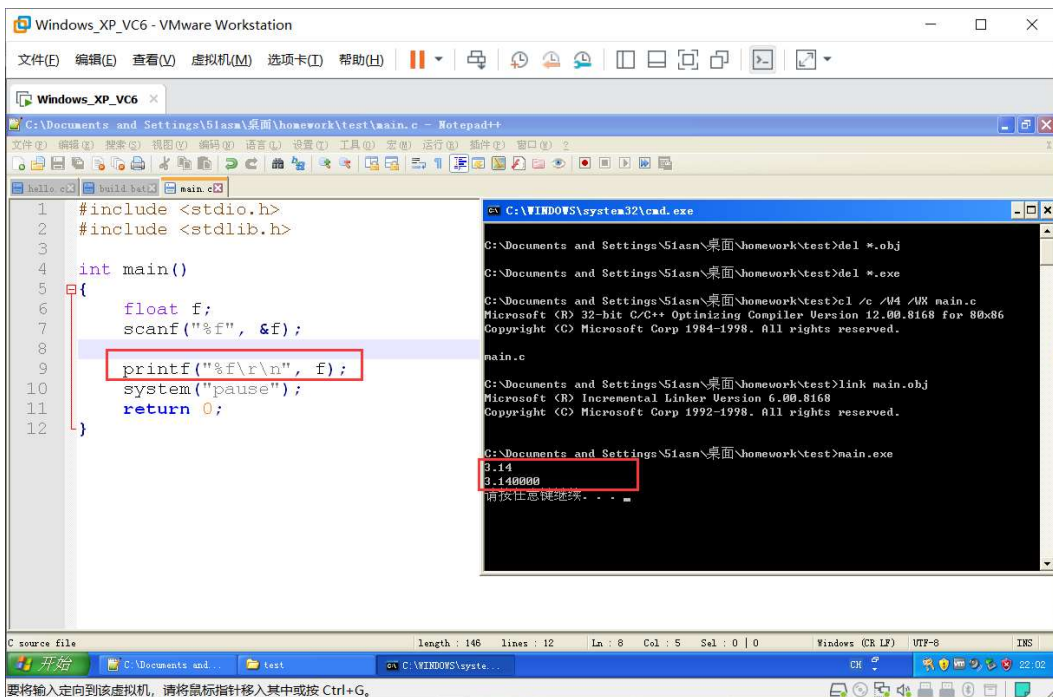
```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    float f;
    scanf("%f", &f);
    /*
    在VC++ 6.0 编译器下，浮点数声明时不初始化，链接时会报以下错误：
    runtime error R6002
    - floating point not loaded
    */
    //添加printf()函数或者声明是对其初始化
    //printf("%f\r\n", f);
    system("pause");
    return 0;
}

```



浮点型初始化的问题:

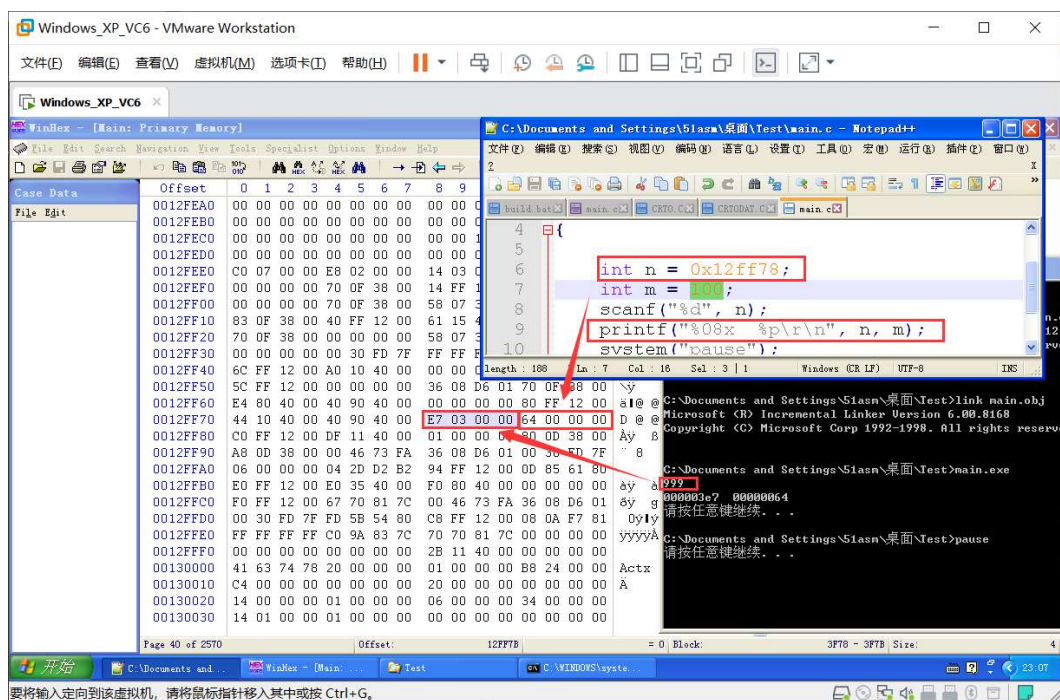
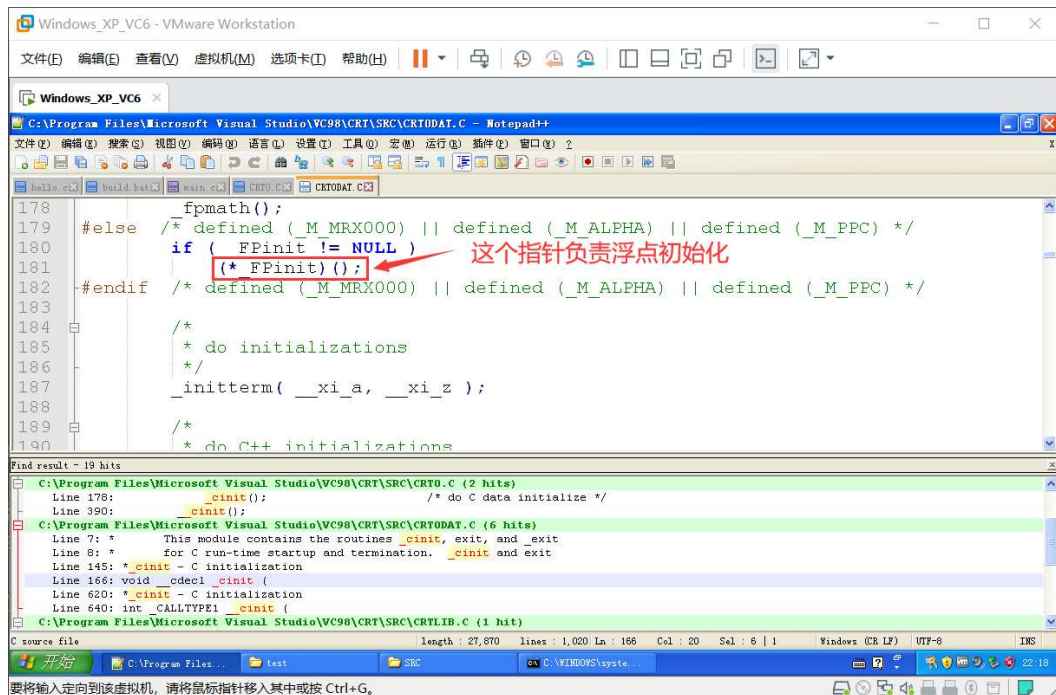


程序编译时，编译器会进行扫描，看程序是否进行浮点数运算。编译器在扫描源码的时候，如果发现源码中没有浮点运算，则认为浮点协处理器无需参与，于是就没有初始化

浮点协处理器：专本负责完成浮点数的读取，完成浮点数装载（写入，运算），指令集：8087

VC++ 6.0 安装目录下的VC98->CRT->SRC->CRTO.C 文件内的 `_cinit()`；函数的作用是初始化浮点协处理器以及初始化标准的全局对象(cin、cout等)，用户自定义的

全局对象，在 `_cinit()`；负责实现。



C语言基本数据类型的使用

不同操作系统的地址是不一样的

XP: 0x12

win2000: 0x14

win7: 0x18

win10: 0x19(随机地址)

浮点数在内存中的存储方式

定点小数的存储: 2个字节存储整数部分, 2个字节存储小数部分

优点: 读入和写出效率高, 访问快

有编码规则的字符序列，变量16进制的内存地址结尾是 0、4、8、C的，对应的十进制数都是4的倍数。

