



§ 2. 基础知识

补充2: 字符的输出

- ★ (前序)将默认输出窗口从Windows终端改为Windows控制台主机
- ★ 有效的输出
- ★ 无效的输出
- ★ 尾零的输出
- ★ 总结



§ 2. 基础知识

- ★ (前序) 将默认输出窗口从Windows终端改为Windows控制台主机
 - 设置方法
 - 新旧控制台的切换
 - 从新版控制台切换回Windows终端

操作文档见大礼包部分



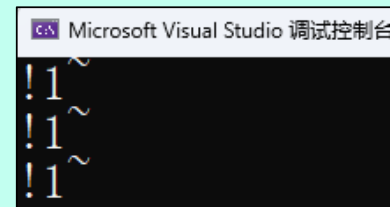
§ 2. 基础知识

★ 有效的输出

- 纯ASCII图形字符的输出(基本ASCII码 33-126之间)
- 汉字的输出(成对的扩展ASCII码)
- 汉字+纯ASCII图形字符的混合输出

```
#include <iostream>
using namespace std;
int main()
{
    cout << "\x21\x31\x7e" << endl;
    cout << "\41\061\176" << endl;
    cout << "!1~" << endl;

    return 0;
}
```





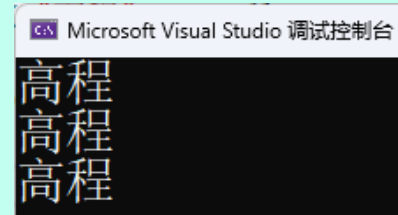
§ 2. 基础知识

★ 有效的输出

- 纯ASCII图形字符的输出(基本ASCII码 33-126之间)
- 汉字的输出(成对的扩展ASCII码)
- 汉字+纯ASCII图形字符的混合输出

```
#include <iostream>
using namespace std;
int main()
{
    cout << "\xB8\xDF\xB3\xCC" << endl;
    cout << "\270\337\263\314" << endl;
    cout << "高程" << endl;

    return 0;
}
```





§ 2. 基础知识

★ 有效的输出

- 纯ASCII图形字符的输出(基本ASCII码 33-126之间)
- 汉字的输出(成对的扩展ASCII码)
- 汉字+纯ASCII图形字符的混合输出

```
#include <iostream>
using namespace std;
int main()
{
    cout << "\x21\xB8\xDF\xB3\xCC\x31\x7e" << endl;
    cout << "\41\270\337\263\314\061\176" << endl;
    cout << "!高程1~" << endl;

    return 0;
}
```





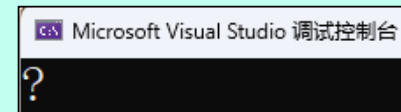
§ 2. 基础知识

★ 无效的输出生

- 单独的扩展ASCII码的输出(中文系统下非法)
- 非法的情况(不成对的扩展ASCII码、含控制字符)
- 同一程序设置不同显示字体的情况下

```
#include <iostream>
using namespace std;
int main()
{
    cout << "\xB8" << '*' << endl; //"高"的前半个

    return 0;
}
```





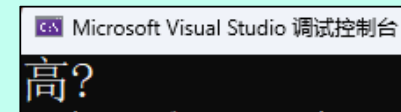
§ 2. 基础知识

★ 无效的输出生

- 单独的扩展ASCII码的输出(中文系统下非法)
- 非法的情况(不成对的扩展ASCII码、含控制字符)
- 同一程序设置不同显示字体的情况下

```
#include <iostream>
using namespace std;
int main()
{
    cout << "\xB8\xDF\xB3" << '*' << endl; //"高"+"程"的前半个

    return 0;
}
```





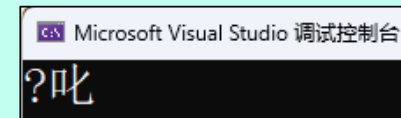
§ 2. 基础知识

★ 无效的输出生

- 单独的扩展ASCII码的输出(中文系统下非法)
- 非法的情况(不成对的扩展ASCII码、含控制字符)
- 同一程序设置不同显示字体的情况下

```
#include <iostream>
using namespace std;
int main()
{
    cout << "\xB8*\xDF\xB3" << endl; //"高"的前半个+*+"程"

    return 0;
}
```



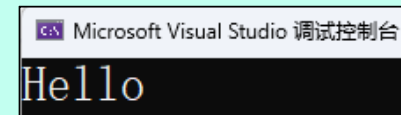


§ 2. 基础知识

★ 无效的输出生

- 单独的扩展ASCII码的输出(中文系统下非法)
- 非法的情况(不成对的扩展ASCII码、含控制字符)
- 同一程序设置不同显示字体的情况下

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello\x8" << endl; //8是退格键←
    return 0;
}
```



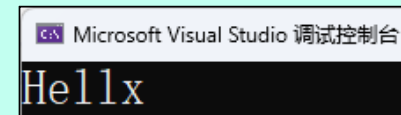


§ 2. 基础知识

★ 无效的输出生

- 单独的扩展ASCII码的输出(中文系统下非法)
- 非法的情况(不成对的扩展ASCII码、含控制字符)
- 同一程序设置不同显示字体的情况下

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello\x8x" << endl; // 8是退格键←，多一个x
    return 0;
}
```





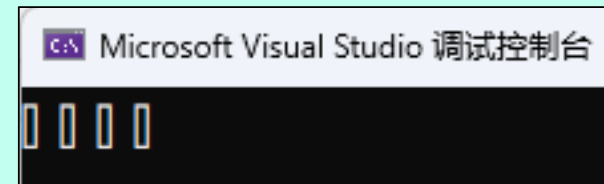
§ 2. 基础知识

★ 无效的输

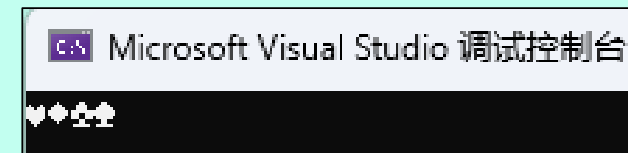
- 单独的扩展ASCII码的输出(中文系统下非法)
- 非法的情况(不成对的扩展ASCII码、含控制字符)
- 同一程序设置不同显示字体的情况下

```
#include <iostream>
using namespace std;
int main()
{
    cout << "\3\4\5\6" << endl;
    return 0;
}
```

新版控制台/新宋体



新版控制台/点阵字体





§ 2. 基础知识

★ 尾零的输出

```
#include <iostream>
using namespace std;
int main()
{
    cout << "a \0" << "a *" << endl;
    cout << "a " << ' \0' << "a *" << endl;
    return 0;
}
```

新版控制台/新宋体28

旧版控制台/新宋体28

旧版控制台/新宋体16

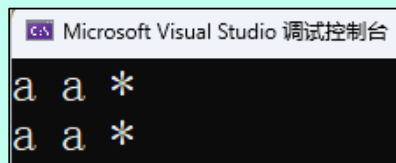


§ 2. 基础知识

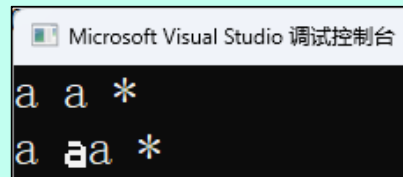
★ 尾零的输出

```
#include <iostream>
using namespace std;
int main()
{
    cout << "a \0" << "a *" << endl;
    cout << "a " << ' \0' << "a *" << endl;
    return 0;
}
```

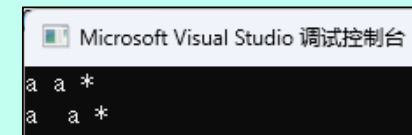
新版控制台/新宋体28



旧版控制台/新宋体28



旧版控制台/新宋体16



现象分析: \0无对应输出

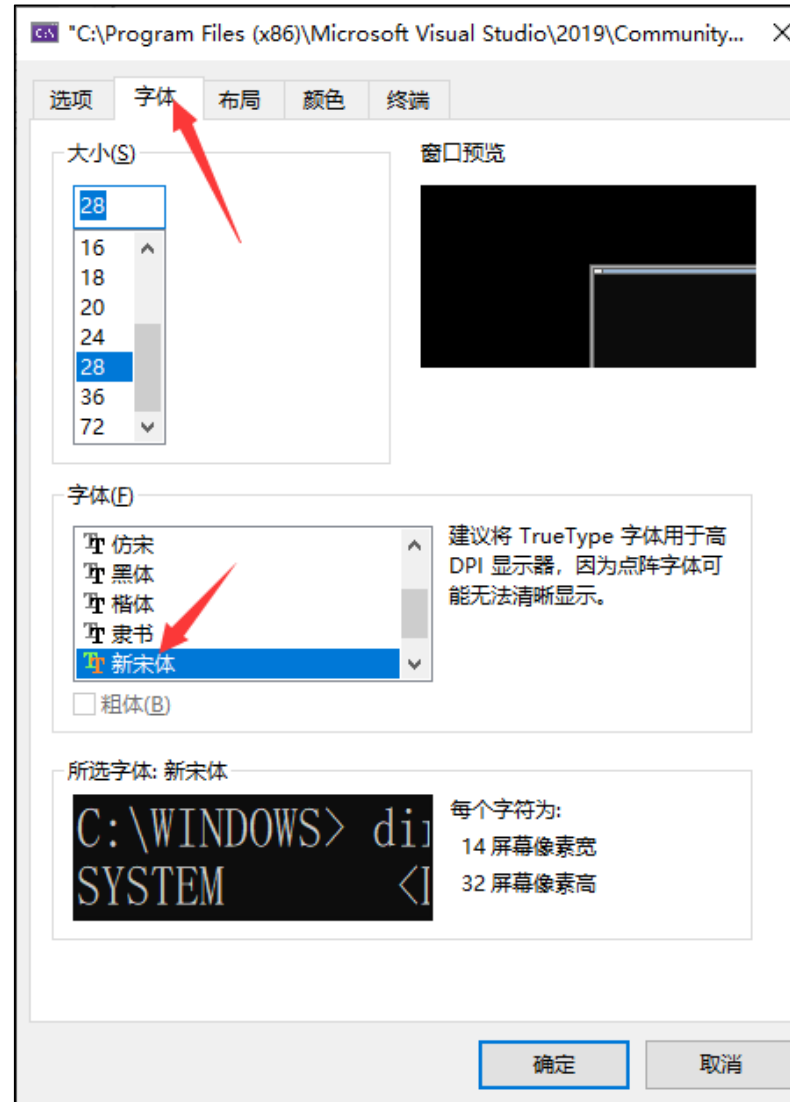
\0对应一个风格不太一样的a

\0对应一个空格

最终结论: \0不是合法的图形字符, 看到的内容不可信!!!

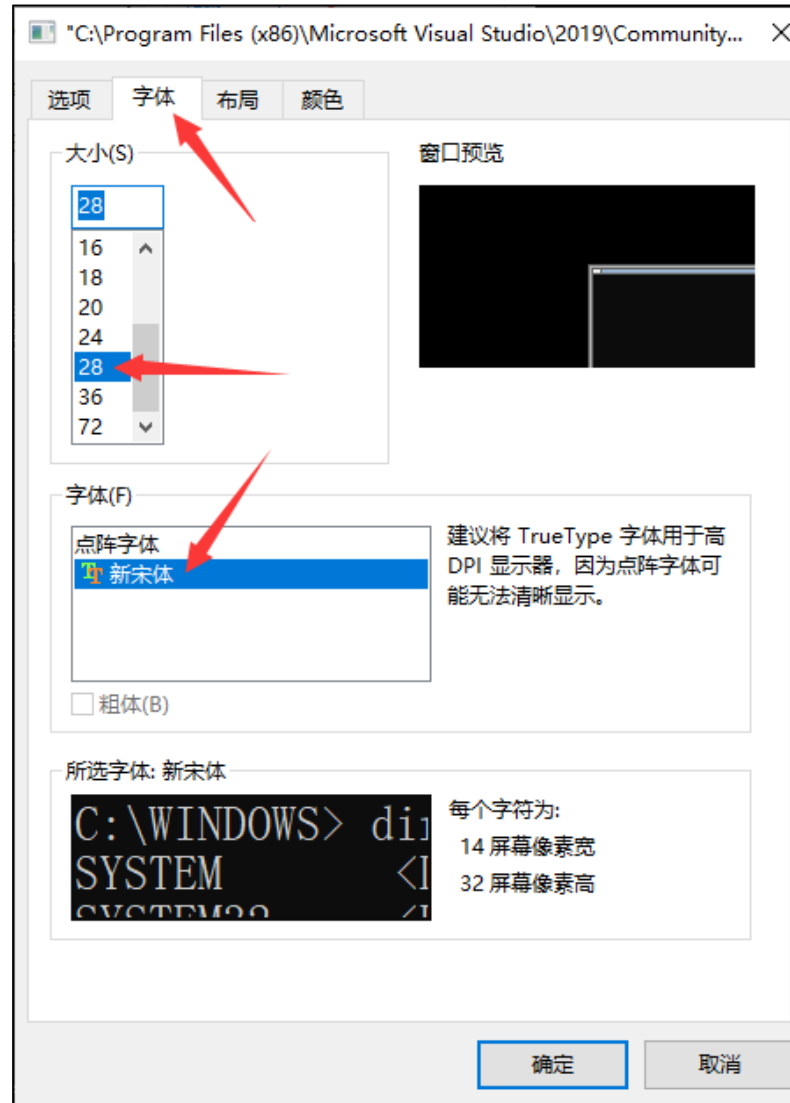


新版控制台/新宋体28



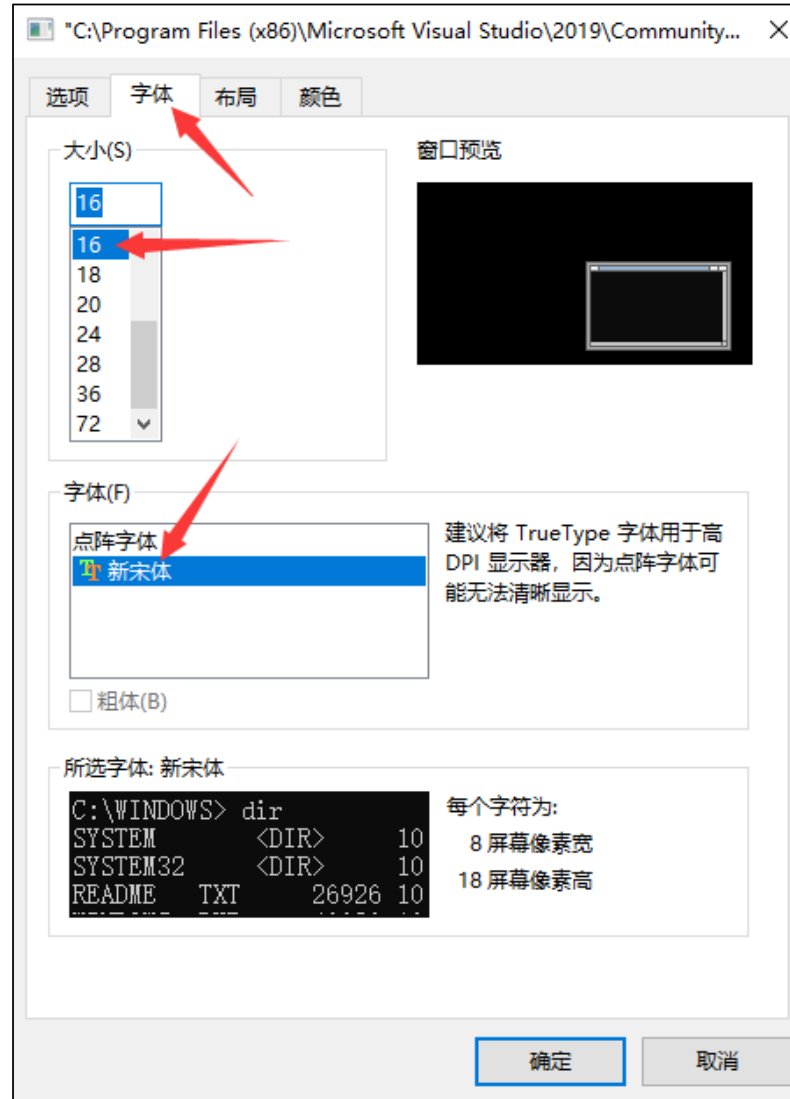


旧版控制台/新宋体28





旧版控制台/新宋体16



§ 2. 基础知识



★ 总结

- 含汉字/图形字符/空格的字符串可以正常显示，**否则输出不可信**
- 想研究长度/占内存空间的做法：strlen、sizeof
- 想得到某个字节确定值的做法：转int打印

```
#include <iostream>
using namespace std;
```

```
int main()
{
    cout << strlen("\xB8\xDF\xB3") << endl; // "高"+"程"的前半个
    cout << sizeof("Hello\x8x") << endl; // 8是退格键←，多一个x
    cout << int(' \xB3') << endl; // "程"的前半个，不能""
    return 0;
}
```

