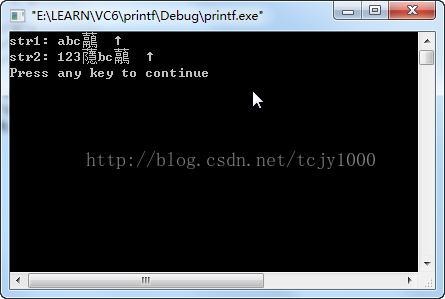
<https://blog.csdn.net/tcjy1000/article/details/46709061>

printf函数以参数"%s"输出字符串时过程为：  
  
  
（1）从首地址开始逐字节寻址，把存储单元（一个字节）内的数据转换为ASCII字符格式输出。  
（2）直到某一个字节内存的元素为字符'\0'时，输出此字符并且寻址结束。  
  
  
      字符数组里没有'\0'，因此使用printf %s 输出时，可能会因为没有结束的'\0'而多输出一些乱码或是字符串。  
如果要正确的输出字符数组，最好一位一位输出。

1. #include "stdafx.h"
2. int main(int argc, char\* argv[])
3. {
4. char str1[3];
5. char str2[3];
6. str1[0]='a';
7. str1[1]='b';
8. str1[2]='c';
9. str2[0]='1';
10. str2[1]='2';
11. str2[2]='3';
13. printf("str1: %s \r\n",str1);         //输出abc，后面有乱码，原因是str1[2]后面可能没有紧跟'\0'，多输出没有用的字符或是乱码。
14. printf("str2: %s \r\n",str2);        //输出123，后面有乱码，原因是str1[2]后面可能没有紧跟'\0'，多输出没有用的字符或是乱码。
15. return 0;
16. }

复制代码

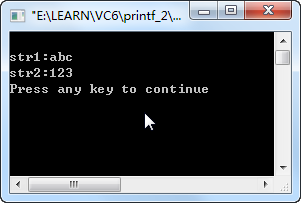
上面运行的结果：有可能不正常。  
 



1. #include "stdafx.h"
2. #include "string.h"
3. int main(int argc, char\* argv[])
4. {
5. char str1[3];
6. char str2[3];
7. unsigned int i,j;
8. str1[0]='a';
9. str1[1]='b';
10. str1[2]='c';
11. str2[0]='1';
12. str2[1]='2';
13. str2[2]='3';
14. printf("\r\nstr1:");
15. for(i=0;i<3;i++)
16. {
17. printf("%c",str1[i]);
18. }
19. printf("\r\nstr2:");
20. for(j=0;j<3;j++)
21. {
22. printf("%c",str2[j]);
23. }
24. printf("\r\n");
25. return 0;
26. }

复制代码

上面的运行结果正常：看来字符串与字符数组，还有一点区别，要认真分析后使用相应的操作函数与操作方法。



版权声明：技术开源与分享，求指教，求扩散，欢迎交流与分享。 https://blog.csdn.net/tcjy1000/article/details/46709061