# 1 TO THE PART OF T

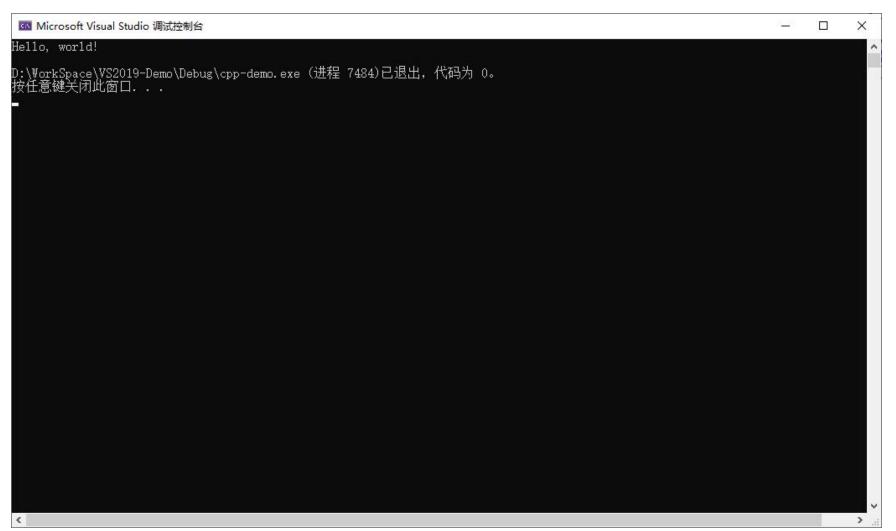
#### 要求:

- 1、完成本文档中所有的题目并写出分析、运行结果,体会字符数组输入输出时不同用法的差异
- 2、题目明确指定编译器外,缺省使用VS2022即可
  - ★ 如果要换成其他编译器,可能需要自行修改头文件适配
  - ★ 部分代码编译时有warning,不影响概念理解,可以忽略
- 3、直接在本文件上作答,写出答案/截图(不允许手写、手写拍照截图)即可;填写答案时,为适应所填内容或贴图, 允许调整页面的字体大小、颜色、文本框的位置等
  - ★ 贴图要有效部分即可,不需要全部内容
  - ★ 在保证一页一题的前提下,具体页面布局可以自行发挥,简单易读即可
  - ★ 不允许手写在纸上,再拍照贴图
  - ★ 允许在各种软件工具上完成(不含手写),再截图贴图
  - ★ 如果某题要求VS+Dev的,则如果两个编译器运行结果一致,贴VS的一张图即可,如果不一致,则两个图都要贴
- 4、转换为pdf后提交
- 5、5月9日前网上提交本次作业(在"文档作业"中提交)



贴图要求: 只需要截取输出窗口中的有效部分即可,如果全部截取/截取过大,则视为无效贴图

例:无效贴图

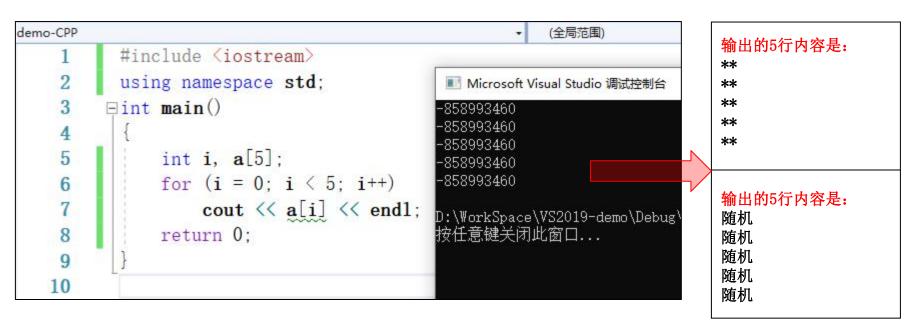


### 例:有效贴图

Microsoft Visual Studio 调试控制台
 Hello, world!

#### 注意:

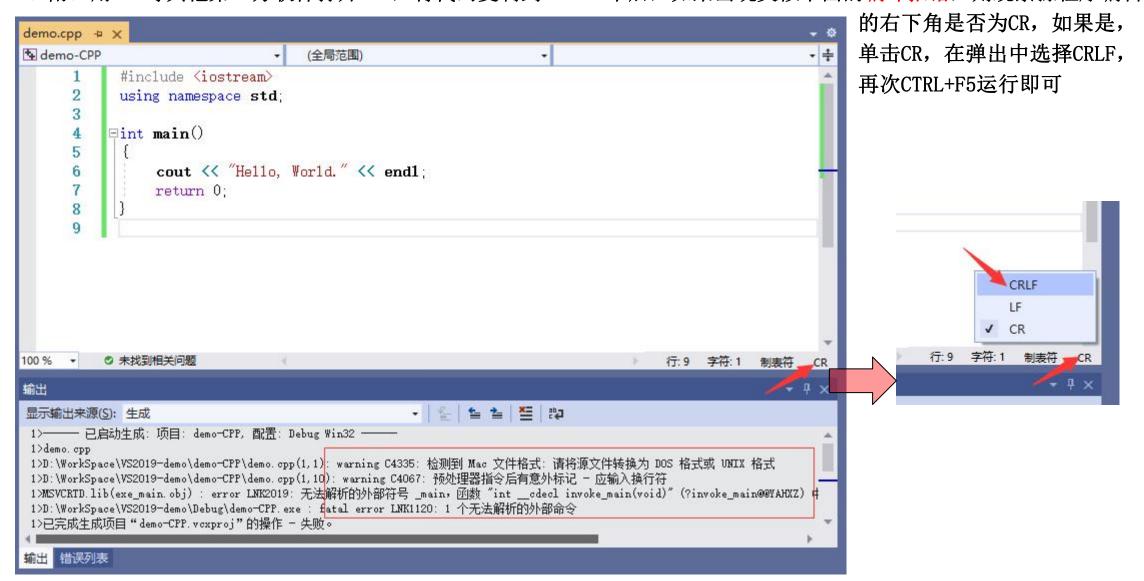
1、部分内容的填写,如果能确定是"不确定值/随机值"的,可直接填写"\*\*/随机"





注意:

2、附:用WPS等其他第三方软件打开PPT,将代码复制到VS2022中后,如果出现类似下面的编译报错,则观察源程序编辑窗



#### 1. 输入

#### 例1: C方式输入单个字符

```
#define CRT SECURE NO WARNINGS //VS需要
#include <iostream>
using namespace std;
int main()
    char a[10];
                                   数组下标表示前有
                                   取地址符号&
    int i:
                                   因为scanf规定后面
                                   必须是变量的地址
    for(i=0; i<10; i++)
        cout \langle\langle int(a[i]) \langle\langle endl;
    scanf("%c%c", &a[3], &a[7]);
    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;
   return 0:
```

```
scanf前首先输出10行,内容是:
随机
scanf时,输入AB并回车,输出是:
随机
随机
随机
65
随机
随机
随机
66
随机
随机
//用不同颜色标注出有变化的内容
```





#### 1. 输入

#### 例2: C++方式输入单个字符

```
#include <iostream>
using namespace std;
int main()
    char a[10];
    int i:
                                        数组下标表示前
                                        无取地址符号&
    for(i=0; i<10; i++)
         cout \langle\langle int(a[i]) \langle\langle endl;
    cin >> a[3] >> a[7]:
    for(i=0; i<10; i++)
         cout << int(a[i]) << endl;</pre>
   return 0;
```

```
cin前首先输出10行,内容是
随机
cin时,输入AB并回车,输出是:
随机
随机
随机
65
随机
随机
随机
66
随机
随机
//用不同颜色标注出有变化的内容
```

#### 1. 输入

例3: C方式多次逐个输入时回车的处理

```
#define CRT SECURE NO WARNINGS //VS需要
#include <iostream>
using namespace std;
int main()
    char a[10];
    int i:
    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;</pre>
    scanf("%c%c", &a[3], &a[7]);
    scanf("%c", &a[0]);
  for(i=0; i<10; i++)
      cout << int(a[i]) << endl:
  return 0:
```

```
scanf前首先输出10行,内容是
随机
随机
随机
随机
随机
随机
随机
随机
随机
随机.
scanf时,输入AB并回车,输出是:
随机
随机
65
随机
随机
随机
66
随机
随机
//用不同颜色标注出有变化的内容
```





#### 1. 输入

例4: C++方式多次逐个输入时回车的处理

```
#include <iostream>
using namespace std;
int main()
    char a[10];
    int i:
    for(i=0: i<10: i++)
        cout << int(a[i]) << endl:
    cin >> a[3] >> a[7]:
    cin >> a[0]:
    for(i=0: i<10: i++)
        cout << int(a[i]) << endl;
    return 0;
```

```
cin前首先输出10行,内容是
随机
随机
随机.
随机
随机
随机
随机
随机.
随机
随机
                     没有反应
cin时,输入AB并回车,表现如何?
多按几次回车,表现如何?
                     没有反应
最后再输入C并回车,则输出是:
67
随机
随机
65
随机.
随机
随机
66
随机
随机
//用不同颜色标注出有变化的内容
综合例3/4得到结论: 当多次逐个输入时,
C方式处理回车的方式是 直接作为一个字符处理
                               本页需填写答案
C++方式处理回车的方式是 不会读取回车
```



#### 1. 输入

字符串形式: scanf("%s",数组名) C方式 cin >> 数组名 C++方式

例5: C方式输入字符串(正确)

```
#define CRT SECURE NO WARNINGS //VS需要
#include <iostream>
using namespace std;
int main()
   char a[10];
                               直接数组名,无下标,
                               也不加&
   int i:
                               因为C/C++规定,数组名
                               代表数组的起始地址
   for(i=0; i<10; i++)
       cout << int(a[i]) << endl;</pre>
   scanf ("%s", a);
   for(i=0; i<10; i++)
       cout << int(a[i]) << end1;
   return 0;
```

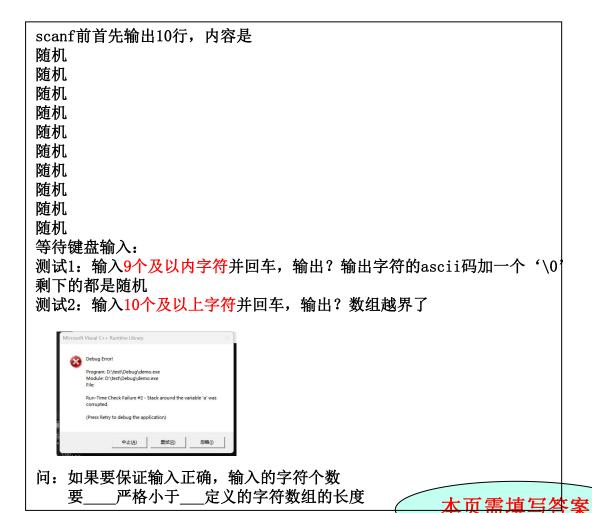
```
scanf前首先输出10行,内容是
随机
等待键盘输入,输入Hello并回车,输出为
101
108
108
111
随机
随机
随机
//用不同颜色标注出有变化的内容
问: 1、回车是否在数组中? 不在
  2、Hello后面的一个字符是什么?
                       '\0'
```



#### 1. 输入

例6: C方式输入字符串(错误)

```
#define CRT SECURE NO WARNINGS //VS需要
#include <iostream>
using namespace std;
int main()
   char a[10];
                               直接数组名, 无下标,
                               也不加&
   int i:
                               因为C/C++规定,数组名
                              代表数组的起始地址
   for(i=0; i<10; i++)
       cout << int(a[i]) << endl;
   scanf ("%s", a);
   for(i=0: i<10: i++)
       cout << int(a[i]) << end1;
   return 0;
```





#### 1. 输入

例7: C++方式输入字符串(正确)

```
#include <iostream>
using namespace std:
int main()
    char a[10];
                                  直接数组名, 无下标,
    int i:
                                      也不加&
    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;</pre>
    cin >> a;
    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;</pre>
    return 0;
```

```
cin前首先输出10行,内容是
随机
随机
随机
随机
随机
随机
随机
随机
随机
等待键盘输入,输入Hello并回车,输出为
101
108
108
111
随机
随机
随机
//用不同颜色标注出有变化的内容
问: 1、回车是否在数组中? 不在啊
  2、Hello后面的一个字符是什么?
                       '\0
```



#### 1. 输入

例8: C++方式输入字符串(错误)

```
#include <iostream>
using namespace std:
int main()
    char a[10];
                                  直接数组名, 无下标,
    int i:
                                       也不加&
    for(i=0; i<10; i+4)
        cout << int(a[i]) << endl;</pre>
    cin >> a:
    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;</pre>
    return 0;
```

```
cin前首先输出10行,内容是
随机
等待键盘输入:
测试1:输入9个及以内字符并回车,输出? ascii码加上一个'\b'
剩下的都是随机
测试2: 输入10个及以上字符并回车,输出? 数组越界了
问: 如果要保证输入正确,输入的字符个数
```

\_定义的字符数组的长度

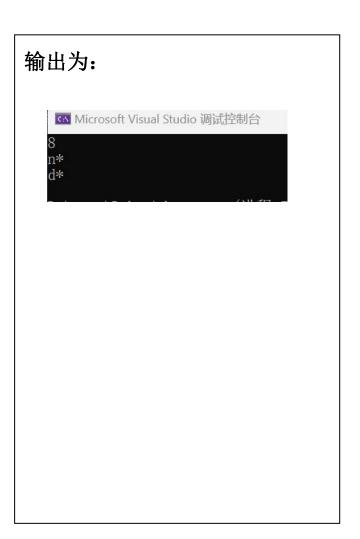
要\_\_\_严格小于\_\_

本页需填写答案

#### 2. 输出

例9: C/C++方式输出单个字符

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
    char a[]="Student": //长度缺省为8
    cout << sizeof(a) << endl:</pre>
    printf("%c*\n", a[5]);
    cout \langle\langle a[3] \langle\langle **' \langle\langle endl;
    return 0;
//输出加*是为了确认只输出了一个字符
```

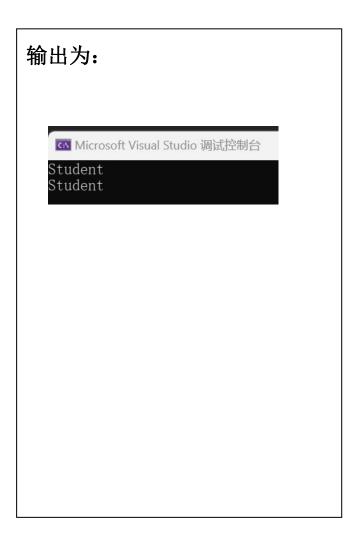






例10: C/C++方式以单个字符+循环形式输出整个数组

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
                            数组 a 缺省长度为8
                            输出[0]-[6], 尾零不输
    int i;
    char a[]="Student";
    for (i=0: i<7: i++)
        printf("%c", a[i]);
    cout << endl; //换行
    for(i=0; i<7; i++)
        cout \langle\langle a[i];
    cout << endl; //换行
    return 0;
```

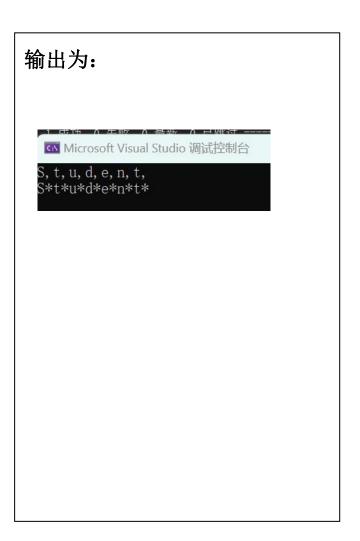




#### 2. 输出

例11: C/C++方式以单个字符+循环形式输出整个数组

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
                                 %c后面多一个,
   int i;
                                 cout方式每个字符
   char a[]="Student";
                                 后面多一个*
   for (i=0; i<7; i++)
       printf("%c,", a[i]);
   cout << endl; //换行
   for(i=0; i<7; i++)
       cout << a[i] << '*';
   cout << endl; //换行
   return 0;
```



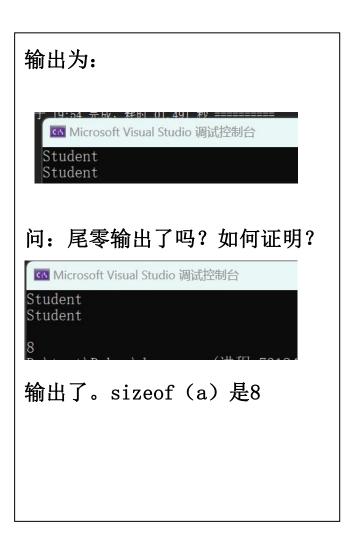


2. 输出

字符串形式: printf("%s", 数组名) C方式 cout << 数组名 C++方式

例12: C/C++以字符串方式输出字符数组

```
#include <iostream>
using namespace std;
int main()
                               跟数组名
                               不是数组元素名
   char a[]="Student"
   printf("%s\n", a);
   cout << a << endl;
   return 0;
```





#### 2. 输出

字符串形式: printf("%s", 数组名) C方式 cout << 数组名 C++方式

例13: C/C++以字符串方式输出字符数组

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
    char a[]="Student\0china";
    cout << sizeof(a) << endl:
    printf("%s*\n", a);
    cout << a << '*' << endl;
    cout \langle\langle a[12] \langle\langle end1;
    return 0;
```

# 输出为: Microsoft Visual Studio 调试控制台 Student\* Student\* 问1: 从本例的结果可知, 数组a的长度是\_\_14\_\_, 最后是否还有隐含的\0? a中的字符串的长度是 7 问2: 字符串形式输出字符数组, 如果数组中包含显式'\0', 则输出到 ′ \0′ 为止



2. 输出

字符串形式: printf("%s", 数组名) C方式 cout << 数组名 C++方式

例14: C/C++以字符串方式输出字符数组(不含尾零)

```
#include <iostream>
using namespace std;
int main()
   //注意:不能以字符串方式初始化
   char a[5]={'C', 'h', 'i', 'n', 'a'};
   printf("%s\n", a);
   cout << a << endl;
   return 0;
```

## <u>输出为:</u>

Microsoft Visual Studio 调试控制台

China烫烫烫滩qXM滮o China烫烫烫滩qXM滮o

问1:为什么会有乱字符? 数组结尾没有自带的'\0'

问2: 如果%s方式换成下面形式

int i;
for (i=0; i<5; i++)
 printf("%c", a[i]);</pre>

还会看到乱字符吗?为什么? 这样不会,这是访问数组的五个 字符





#### 2. 输出

字符串形式: printf("%s", 数组名) C方式 cout << 数组名 C++方式

例15: C/C++以字符串方式输出字符数组(不含尾零)

```
#include <iostream>
using namespace std:
int main()
   char a[5]: //不初始化
   printf("%s\n", a);
   cout << a << endl:
   return 0;
```

#### 输出为:

问1:为什么会有乱字符?没有初始化,没有'\0'

问2: 乱字符出现几行是正常的? 一行? 多行? 或者都正常? 都是两行,只是因为没有换行。 没有结束'\0',长度不会一定

结论:不能字符串形式输出不含 \_ '\0'的字符数组,否则 可能会得到不正确的结果



3. 从任一元素开始以字符串形式输入/输出



#### 例16: 从任一元素开始以字符串形式输出

```
#include <iostream>
using namespace std;
                              %s形式
int main()
   char a[]="Student";
                               &数组元素名形式
   printf("%s\n", &a[3]);
   cout << &a[3] << end1;
   return 0;
```



3. 从任一元素开始以字符串形式输入/输出



#### 例17: C方式从任一元素开始以字符串形式输入

```
#define CRT SECURE NO WARNINGS //VS需要
#include <iostream>
using namespace std;
int main()
    int i;
    char a[10];
                                    &数组元素名形式
    for(i=0; i<10; i++)
        cout \langle\langle int(a[i]) \langle\langle endl;
    scanf ("%s", &a[3]);
    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;
    return 0;
```

```
scanf先输出10行,内容是
随机
随机
随机
随机
随机
随机
随机
随机
随机
等待键盘输入,输入Hello并回车,输出为
随机
随机
随机
72
101
108
108
111
随机
//用不同颜色标注出有变化的内容
```

3. 从任一元素开始以字符串形式输入/输出



#### 例18: C++方式从任一元素开始以字符串形式输入

```
#include <iostream>
using namespace std:
int main()
    int i:
    char a[10];
                                 &数组元素名形式
    for (i=0; i<10; i++)
         cout \langle\langle int(a[i]) \langle\langle endl;
    cin >> &a[3]:
    for(i=0; i<10; i++)
         cout << int(a[i]) << endl;</pre>
    return 0;
```

```
cin先输出10行,内容是
随机
等待键盘输入,输入Hello并回车,输出为
随机
随机
随机
72
101
108
108
111
随机
//用不同颜色标注出有变化的内容
综合例16-18的结果,得出的结论是:
C/C++方式从任一元素开始以字符串形式
输入输出时,表示形式都是 地址
```

本页需填写答案

### 1-3. 总结



### 完成下表(给出了第一行的答案供参考):

	C方式	C++方式
输入单个字符	scanf("%c", &元素名)	cin >> 元素名
输入字符串	scanf("%s", &数组名)	cin>>数组名
输出单个字符	printf("%c", &数组名)	cout>>元素名
输出字符串	printf("%s", &数组名)	cout>>数组名
任一元素开始输入串	scanf("%s", &数组名)	cin>>&元素名
任一元素开始输出串	printf("%s", &数组名)	cout>>&元素名

### 4. 多个字符串的输入



#### 例19: C方式多个字符串的输入

```
#define _CRT_SECURE_NO_WARNINGS //VS需要
#include <iostream>
#include <cstdio>
using namespace std;
int main()
   char a[10], b[20];
   scanf ("%s%s", a, b);
   printf("%s-%s\n", a, b);
   return 0;
```

1、假设输入为abc空格def并回车 则输出为: Microsoft Visual Studio 调试控制台 abc def abc-def 2、假设输入为abc回车 def回车 则输出为: Microsoft Visual Studio 调试控制台 abc abc-def 结论: 空格是\_\_\_\_b\_ A. 输入串中的合法字符 B. 输入分隔符

### 4. 多个字符串的输入



#### 例20: C++方式多个字符串的输入

```
#include <iostream>
using namespace std;
int main()
    char a[10], b[20];
    cin \gg a \gg b;
    cout << a << '-' << b << endl;
    return 0;
```

1、假设输入为abc空格def并回车则输出为:



2、假设输入为abc回车 def回车

则输出为:



结论: 空格是\_\_\_b\_\_\_\_

A. 输入串中的合法字符

B. 输入分隔符

综合例19-20可知: scanf/cin从键盘上输入的字符串 不能包含\_\_\_\_空格\_\_\_

- 4. 多个字符串的输入
- ★ 从键盘输入含空格字符串的方法(不同编译器不同)
  - VS2022 : 有gets\_s, 无gets, 有fgets
  - Dev C++ : 有gets, 无gets\_s, 有fgets
  - fgets函数的原型定义为:

fgets(字符数组名,最大长度, stdin);

但与gets/gets s的表现有不同,请自行观察

★ scanf/cin通过某些高级设置方式还是可以输入含空格的字符串的,本课程不再讨论





- 4. 多个字符串的输入
- ★ 从键盘输入含空格字符串的方法(不同编译器不同)

### 例21: VS下用gets\_s输入含空格的字符串

```
#include <iostream>
using namespace std;
int main()
    char a[10], b[20];
    gets s(a);
    gets_s(b);
    cout << a << endl:
    cout << b << endl;
    return 0;
```

1、键盘输入abc空格def并回车, 会继续等待输入, 再输入xyz并回车 则输出为: Microsoft Visual Studio 调试控制台 abc def abc def 2、键盘制入超过9个子付,观察 输入Hello并回车, 再输入超过19个字符,观察 为什么a最长输入只能是9?数组长度10,加上一个 '\0' 为什么b最长输入只能是19?数组长度20,加上一个 **'**\0'

- 4. 多个字符串的输入
- ★ 从键盘输入含空格字符串的方法(不同编译器不同)

#### 例22: DevC++下用gets输入含空格的字符串

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
    char a[10], b[20];
   gets(a);
    gets(b);
    cout << a << endl:
    cout << b << endl;
    return 0;
```

1、键盘输入abc<mark>空格def</mark>并回车, 会继续等待输入, 再输入xyz并回车 则输出为:



2、键盘输入超过9个字符,观察

中斯但是个会让程序结束输出
3. 键盘先输入Hello并同车。

3、键盘先输入Hello并回车, 再输入超过19个字符,观察

#### agirtungi test/Debug/demo.exe (进程 23584)已退出,代码为 3。 丘臺體美闲此窗口,,

中國但是不会让程序结束输出

问:为什么a最长输入只能是9?数组长度10,加上一个'\0'

为什么b最长输入只能是19?数组长度20,加上一个'\0



- 4. 多个字符串的输入
- ★ 不同编译器从键盘输入含空格字符串的方法不同

#### 例23: VS和Dev C++均可用fgets输入含空格的字符串

```
#include <iostream>
#include <cstdio>
using namespace std:
int main()
    char a[10], b[20];
    fgets (a, 10, stdin);
    fgets(b, 20, stdin):
    cout << a << endl:
    cout << b << endl:
    int i:
    for (i=0; a[i]!=' \setminus 0'; i++)
        cout << int(a[i]) << ' ';</pre>
    cout << endl:
    for (i=0; b[i]!='\0'; i++)
        cout << int(b[i]) << ' ';
    cout << endl:
    return 0:
```

1、键盘输入abc空格def abc def abc d

问1: 和例21-22的输出区别在哪里?

fgets读入回车键和换行符都看做换行。gets和gets\_s都不会读入换行

问2: 后面两段红色代码的目的是什么?

验证读入了换行,就是ascii为10

2、键盘输入9个字符并回车,则输出为:

九个字符给到了a数组,而换行给了b数组

3、如果输入28个字符并回车,则输出为:

前九个字符给a数组,后面19个字符给了b数组,此时没有换行符是因为有'\0'

4、如果输入超过28个字符并回车,

则输出为:

超出的不计入

前九个字符给a数组,后面19个字符给了b数组

- 5. 二维字符数组的输入/输出
- ★ 数组名加双下标表示元素,单下标表示一维数组



#### 例24: 二维字符数组以双下标形式输出单个字符/单下标形式输出字符串

```
#include <iostream>
using namespace std;
int main()
    char a[3][30]={"ABCDEFGHIJKLMNOPQRSTUVWXYZ",
                       "abcdefghijklmnopqrstuvwxyz",
                       "0123456789" }:
    // 单个字符输出(数组名+双下标)
    printf("a[0][2]=%c\n", a[0][2]);
    cout \langle \langle "a[1][20] = " \langle \langle a[1][20] \rangle \langle \langle end1;
    // 字符串输出(数组名+单下标)
    printf((a[0]=%s\n'', a[0]);
    cout \langle \langle "a[2]=" \langle \langle a[2] \langle \langle end1;
    return 0;
```



- 5. 二维字符数组的输入/输出
- ★ 数组名加双下标表示元素,单下标表示一维数组



#### 例25: 二维字符数组以双下标形式输入单个字符

```
#define CRT SECURE NO WARNINGS //VS需要
#include <iostream>
using namespace std;
int main()
    char a[3][30]={"ABCDEFGHIJKLMNOPQRSTUVWXYZ",
                   "abcdefghijklmnopgrstuvwxyz",
                   "0123456789" }:
   // 单字符输入(数组名+双下标)
    scanf("%c\n", &a[0][2]); //格式符为%c
    cin >> a[1][20]:
                      //无&
   // 字符串输出(数组名+单下标)
    printf((a[0]=%s\n'', a[0]);
    cout \langle \langle "a[1]=" \langle \langle a[1] \langle \langle end1;
    return 0;
```



- 5. 二维字符数组的输入/输出
- ★ 数组名加双下标表示元素,单下标表示一维数组

#### 例26: 二维字符数组以单下标形式输入字符串

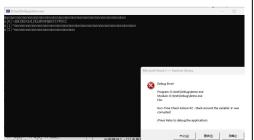
```
#define _CRT_SECURE_NO_WARNINGS //VS需要
#include <iostream>
using namespace std;
int main()
     char a[3][30]={"ABCDEFGHIJKLMNOPQRSTUVWXYZ",
                         "abcdefghijklmnopqrstuvwxyz",
                         "0123456789" }:
     scanf("%s", a[1])://a[1]是一维数组名,无&
     cout \langle\langle "a[0]=" \langle\langle a[0] \langle\langle end1;
     cout \langle \langle "a[1]=" \langle \langle a[1] \langle \langle end1;
     cout \langle \langle "a[2]=" \langle \langle a[2] \langle \langle end1;
     return 0;
```

- 1、输入≤29个字符,输出为:
- 2、输入30-59个字符,输出为:
- 3、输入60个以上字符,输出为:

将scanf换为 cin >> a[1]; 再重复1、2、3,观察结果







问1:输入30<sup>~</sup>59个字符为什么不 出现错误?a[2]中是什么? 多出来的字符通过连续的地址存放到a[2]之中

问2: 简述你是怎么理解二维数组 越界的?

当每一个一维数组的地址都被超过的时候,二维数组才会越界

本页需填写答案



- 5. 二维字符数组的输入/输出
- ★ 数组名加双下标表示元素,单下标表示一维数组

#### 例27: 二维字符数组从任一位置开始输出字符串

```
#include <iostream>
using namespace std:
int main()
     char a[3][30]={"ABCDEFGHIJKLMNOPQRSTUVWXYZ",
                       "abcdefghijklmnopgrstuvwxyz",
                       "0123456789" }:
    //(第1组)单字符输出(数组名+双下标)
    printf("a[0][2]=%c\n", a[0][2]);
    cout \langle \langle "a[1][20] = " \langle \langle a[1][20] | \langle \langle end1 \rangle \rangle
    //(第2组)字符串输出(&+数组名+双下标)
    printf("a[0][2]=%s\n", &a[0][2]);
    cout \langle \langle "a[1][20] = " \langle \langle \&a[1][20] \rangle \langle \langle end1;
    //(第3组)字符串输出(数组名+单下标)
    printf((a[0]=%s\n(a[0]):
    cout \langle \langle "a[2]=" \langle \langle a[2] \langle \langle end1:
    return 0:
```

```
Microsoft Visual Studio 调试控制台
          =CDEFGHIJKLMNOPQRSTUVWXYZ
       [0]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
输出为:
问1: 同样双下标形式(第1/2组),
    怎样输出单个字符?
    怎样输出字符串?
输出数组的双下标:
输出该元素的地址,就可以输出从该元素开始的字符串
问2: 如何修改第2组的输出
   (必须保持双下标形式不变),
    使输出结果与第3组一致?
printf("a[0]=%s\n",&a[0][0]);
cout << "a[2] = " << &a[2][0] << end1:
```

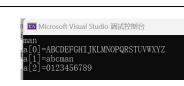
- 5. 二维字符数组的输入/输出
- ★ 数组名加双下标表示元素,单下标表示一维数组

#### 例28: 二维字符数组从任一位置开始输入字符串

```
#define _CRT_SECURE_NO_WARNINGS //VS需要
#include <iostream>
using namespace std;
int main()
     char a[3][30]={"ABCDEFGHIJKLMNOPQRSTUVWXYZ",
                         "abcdefghijklmnopqrstuvwxyz",
                         "0123456789" }:
     scanf ("%s", &a[1][3])://&+数组名+双下标
     cout \langle\langle "a[0]=" \langle\langle a[0] \langle\langle end1;
     cout \langle \langle "a[1]=" \langle \langle a[1] \langle \langle end1;
     cout \langle \langle "a[2]=" \langle \langle a[2] \langle \langle end1;
     return 0;
```

- 1、输入≤26个字符,输出为:
- 2、输入27-56个字符,输出为:
- 3、输入56个以上字符,输出为:

将scanf换为 cin >> &a[1][3]; 再重复1、2、3, 观察结果







问1: 输入27~56个字符为什么不 出现错误? a[2]中是什么?

a[2]之中还有30的空间,a[2]存从28位开始的

问2:如果想不影响a[2],

例26中是≤29个字符,

本例中是≤26个字符,

差别在哪?

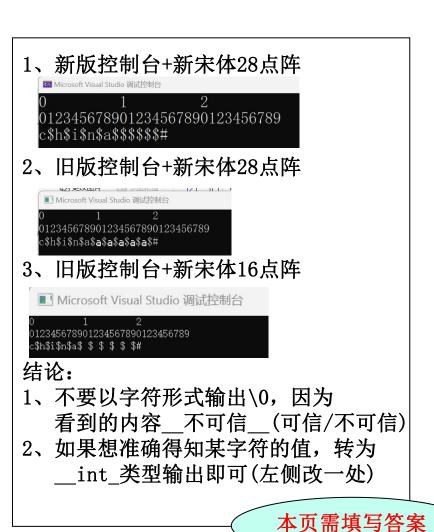
本案例中想要a[1][3]之后的值,就少了三个数据三个空间, 就是29-3=26

#### 6. 尾零的输出



#### 例29: 在不同的控制台及字体设置下尾零输出的差异

```
#include <iostream>
using namespace std:
int main()
   int i:
   char a[10] = { 'c', 'h', 'i', 'n', 'a' };
   cout << "0" 1 2 " << endl; //标尺
   cout << "012345678901234567890123456789" << endl: //标尺
   for (i = 0; i < 10; i++)
       cout << a[i] << '$'; //确认a[i]是否输出
   cout << '#' << endl: //加行尾识别符
   return 0;
```



#### 6. 尾零的输出



本页需填写答案

例30: 在不同的控制台及字体设置下其它非图形字符输出的差异

(去ASCII码表中查表示扑克牌四种花色的字符,用测试程序打印含这4个字符的字符串,然后贴图)

```
#include <iostream>
using namespace std;
int main()
    char a = 3, b = 4, c = 5, d = 6;
    cout << a << b << c << d:
    return 0;
```

1、某版控制台+某字体/某点阵 (此处找到一种可显示的) ■ Microsoft Visual Studio 调试控制台 2、某版控制台+某字体/某点阵 (此处随便找到一种不显示的即可) Microsoft Visual Studio 调试控制台 结论: 上页的结论1也\_\_适用\_(适用/不适用) 于其它非图形字符