



## §. 基础知识题 – 字符数组的输入与输出

要求:

- 1、完成本文档中所有的测试程序并填写运行结果，体会字符数组输入输出时不同用法的差异
- 2、需完成的页面，右上角有标注，直接在本文件上作答，**用蓝色写出答案**即可  
★ 运行结果**允许**截图后贴在文档中，内容不要相互重叠即可
- 3、如果写答案时，字数超出了框架范围，可以缩小字体或者扩大框架，只要能够清晰识别即可
- 4、所有例子仅在VS2019下测试过，如果要换成其他编译器，可能需要自行修改头文件适配  
★ 除题目明确指定编译器外，缺省使用VS2019即可
- 5、部分代码编译时**有warning**，不影响概念理解，**可以忽略**
- 6、转换为pdf后在“实验报告”中提交（**11.22前**）

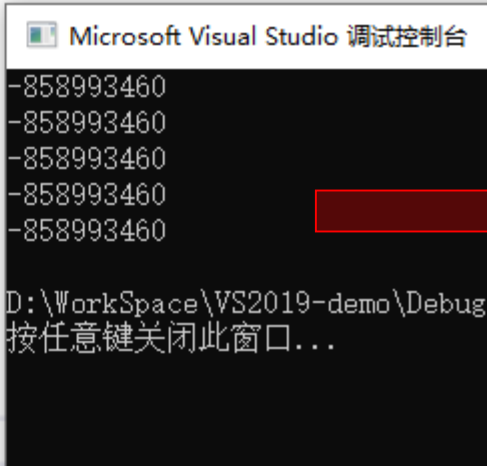


## §. 基础知识题 – 字符数组的输入与输出

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”

```
demo-CPP (全局范围)
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int i, a[5];
6      for (i = 0; i < 5; i++)
7          cout << a[i] << endl;
8      return 0;
9  }
10
```



输出的5行内容是:

\*\*  
\*\*  
\*\*  
\*\*  
\*\*

输出的5行内容是:

随机  
随机  
随机  
随机  
随机

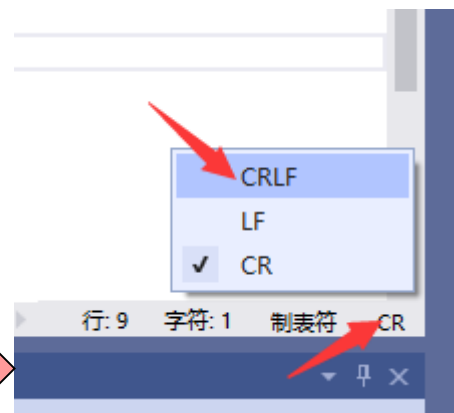
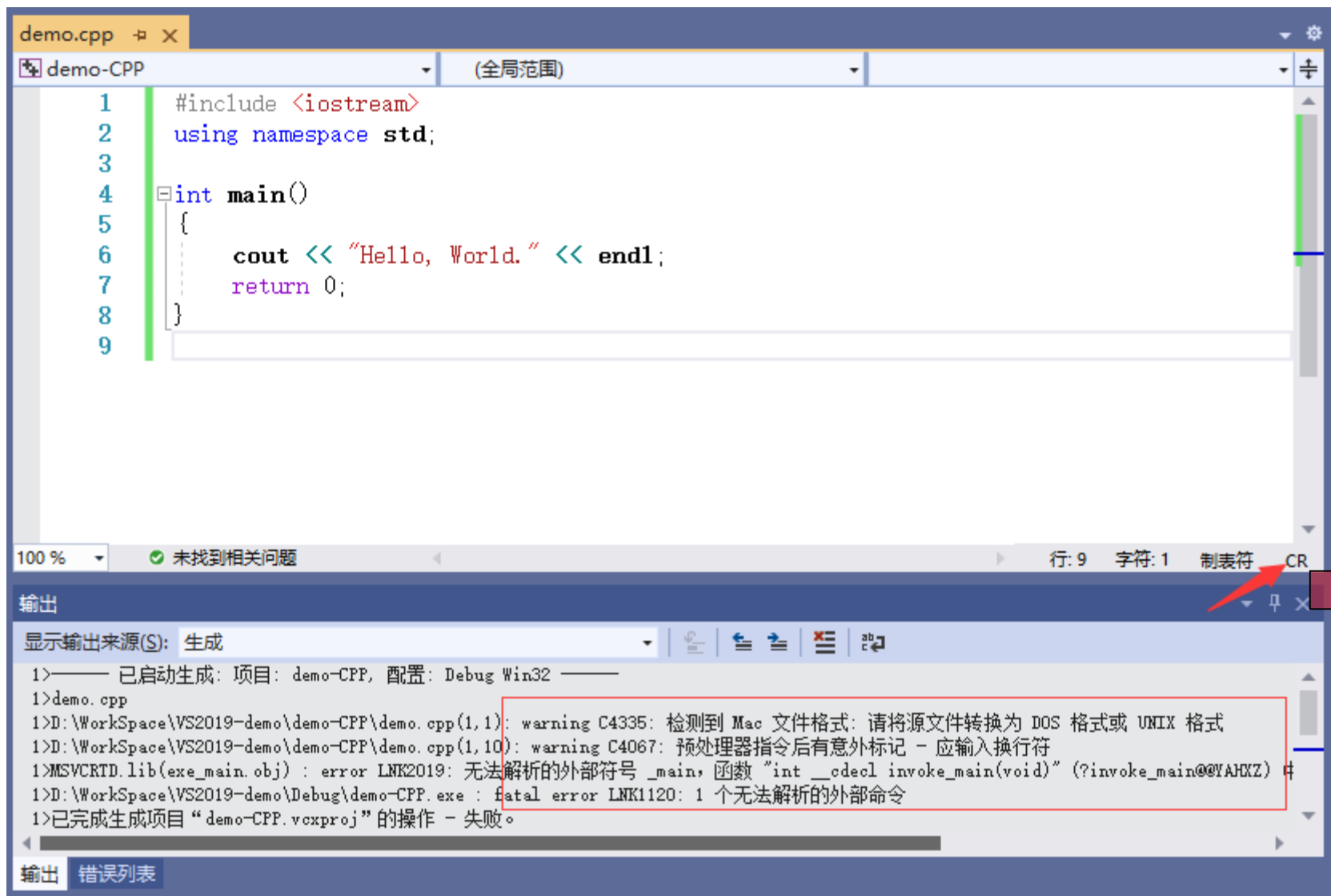


## §. 基础知识题 - 字符数组的输入与输出

注意:

2、附：用WPS等其他第三方软件打开PPT，将代码复制到VS2019中后，如果出现类似下面的**编译报错**，则观察源程序编辑窗

的右下角是否为CR，如果是，单击CR，在弹出中选择CRLF，再次CTRL+F5运行即可





## §. 基础知识题 - 字符数组的输入与输出

### 1. 输入

逐个输入: scanf("%c",&数组元素)     **C方式**

cin >> 数组元素     **C++方式**

#### 例1: C方式输入单个字符

```
#define _CRT_SECURE_NO_WARNINGS //VS2019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;

    scanf("%c%c", &a[3], &a[7]);

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    return 0;
}
```

数组下标表示前有  
取地址符号&  
因为scanf规定后面  
必须是变量的地址

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  
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

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 1. 输入

逐个输入: scanf("%c",&数组元素)    **C方式**

cin >> 数组元素    **C++方式**

### 例2: C++方式输入单个字符

```
#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];

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    return 0;
}
```

数组下标表示前  
无取地址符号&

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

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

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 1. 输入

逐个输入: scanf("%c",&数组元素)    **C方式**  
              cin >> 数组元素        **C++方式**

### 例3: C方式多次逐个输入时回车的处理

```
#define _CRT_SECURE_NO_WARNINGS //VS2019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;

    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:

**10**  
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

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 1. 输入

逐个输入: scanf("%c",&数组元素)    **C方式**

cin >> 数组元素    **C++方式**

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

cin时, 输入AB并回车, 表现如何?

按一次回车无反应

多按几次回车, 表现如何?

依然无反应

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 将回车当作字符来处理

**C++方式**处理回车的方式是cin不读回车, 回车不读成字符

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 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 //VS2019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;

    scanf("%s", a);

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    return 0;
}
```

直接数组名, 无下标,  
也不加&  
因为C/C++规定, 数组名  
代表数组的起始地址

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

等待键盘输入, 输入Hello并回车, 输出为

72  
101  
108  
108  
111  
0

随机  
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'

本页需填写答案





## §. 基础知识题 - 字符数组的输入与输出

### 1. 输入

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

#### 例6: C方式输入字符串(错误)

```
#define _CRT_SECURE_NO_WARNINGS //VS2019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;

    scanf("%s", a);

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    return 0;
}
```

直接数组名, 无下标,  
也不加&  
因为C/C++规定, 数组名  
代表数组的起始地址

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

等待键盘输入:

测试1: 输入10个字符并回车, 输出?

显示的是10个字符的ASCII码, 但是引发异常  
因为数组越界, 结束也没有'\0'

测试2: 输入10个以上字符并回车, 输出?

显示的仍然是10个字符的ASCII码, 但是引发  
异常, 因为数组越界, 结束也没有'\0'



问: 如果要保证输入正确, 输入的字符个数  
要小于定义的字符数组的长度

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 1. 输入

字符串形式: scanf("%s", 数组名)

C方式

cin >> 数组名

C++方式

#### 例7: C++方式输入字符串(正确)

```
#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;

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    return 0;
}
```

直接数组名, 无下标,  
也不加&

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

等待键盘输入, 输入Hello并回车, 输出为

72  
101  
108  
108  
111  
0

随机  
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'

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 1. 输入

字符串形式: scanf("%s", 数组名)

C方式

cin >> 数组名

C++方式

### 例8: C++方式输入字符串(错误)

```
#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;

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    return 0;
}
```

直接数组名, 无下标,  
也不加&

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

等待键盘输入:

测试1: 输入10个字符并回车, 输出?

显示的是10个字符的ASCII码, 但是引发异常  
因为数组越界, 结束也没有'\0'

测试2: 输入10个以上字符并回车, 输出?

显示的仍然是10个字符的ASCII码, 但是引发  
异常, 因为数组越界, 结束也没有'\0'



问: 如果要保证输入正确, 输入的字符个数  
要\_\_小于\_\_定义的字符数组的长度

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 2. 输出

逐个: printf("%c", 数组元素)

C方式

cout << 数组元素

C++方式

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

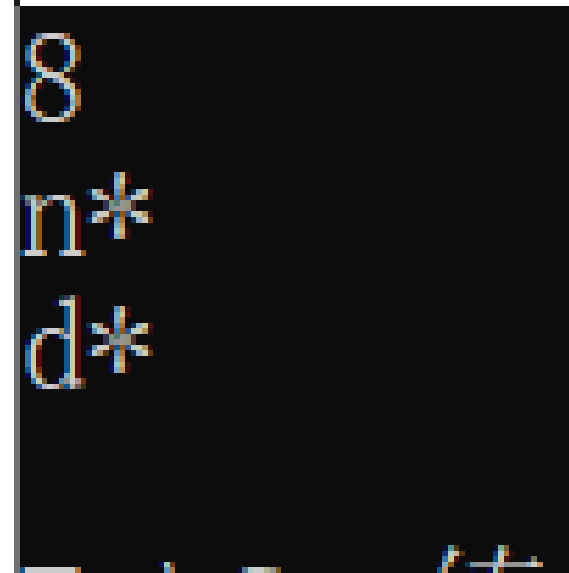
    cout << sizeof(a) << endl;

    printf("%c*\n", a[5]);

    cout << a[3] << '*' << endl;

    return 0;
}
//输出加*是为了确认只输出了一个字符
```

输出为:





## §. 基础知识题 - 字符数组的输入与输出

### 2. 输出

逐个: printf("%c", 数组元素)

C方式

cout << 数组元素

C++方式

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()
{
    int i;
    char a[]="Student";

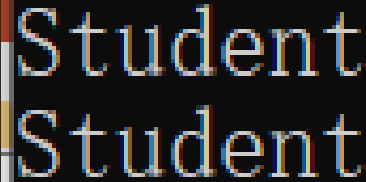
    for(i=0; i<7; i++)
        printf("%c", a[i]);
    cout << endl; //换行

    for(i=0; i<7; i++)
        cout << a[i];
    cout << endl; //换行

    return 0;
}
```

数组 a 缺省长度为8  
输出[0]-[6]，尾零不输出

输出为:



本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 2. 输出

逐个: printf("%c", 数组元素)

C方式

cout << 数组元素

C++方式

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()
{
    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 << a[i] << '*';
    cout << endl; //换行

    return 0;
}
```

%c后面多一个,  
cout方式每个字符  
后面多一个\*

输出为:

```
S, t, u, d, e, n, t,
S*t*u*d*e*n*t*
```

本页需填写答案



## §. 基础知识题 – 字符数组的输入与输出

### 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;

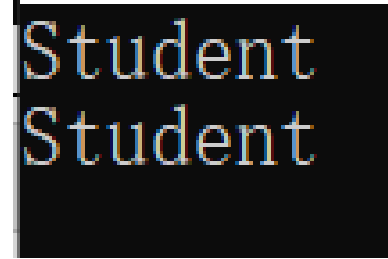
int main()
{
    char a[]="Student";

    printf("%s\n", a);
    cout << a << endl;

    return 0;
}
```

跟数组名  
不是数组元素名

输出为:



问: 尾零输出了吗? 如何证明?  
尾0是自动加的, 这是直接输出整个字符串, 一定要有尾0.

用sizeof(a)发现是8, 说明后面是有' \0' 的。  
并且输出a[7]发现是0, 也说明最后有' \0'

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 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 << a[12] << endl;

    return 0;
}
```

输出为:

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?

有  
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

本页需填写答案





## §. 基础知识题 – 字符数组的输入与输出

### 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)

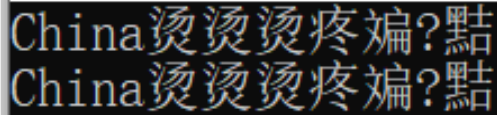
```
#include <iostream>
using namespace std;

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;
}
```

输出为:



问1: 为什么会有乱字符?

因为数组结尾没有' \0' , 并没有结束。

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]);
```

还会看到乱字符吗? 为什么?

不会, 因为这是确定的输出5个字符。不会越界。

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 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)

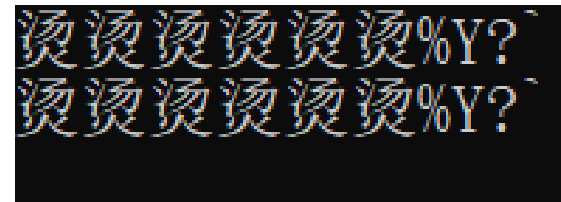
```
#include <iostream>
using namespace std;

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?

因为没有初始化, 是随机值, 并且也没有 '\0' 作为结束符。

问2: 乱字符出现几行是正常的?

一行? 多行? 或者都正常?

乱字符可能每次出现的都不一样, 但是每次都是两行, 因为没有换行, 也没有结束, 所以长度不一定, 但是只是两行

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

本页需填写答案



## §. 基础知识题 – 字符数组的输入与输出

### 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;

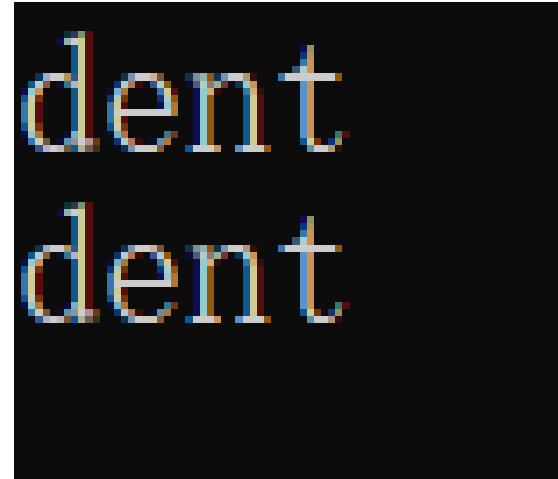
int main()
{
    char a[]="Student";
    printf("%s\n", &a[3]);
    cout << &a[3] << endl;

    return 0;
}
```

%s形式

&数组元素名形式

输出为：





## §. 基础知识题 - 字符数组的输入与输出

### 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 //VS2019需要
#include <iostream>
using namespace std;

int main()
{
    int i;
    char a[10];

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    scanf("%s", &a[3]);

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    return 0;
}
```

&数组元素名形式

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  
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  
0  
随机

//用不同颜色标注出有变化的内容

本页需填写答案



## §. 基础知识题 – 字符数组的输入与输出

### 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];

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    cin >> &a[3];

    for(i=0; i<10; i++)
        cout << int(a[i]) << endl;

    return 0;
}
```

&数组元素名形式

cin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

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

0

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\_的形式

本页需填写答案



# § . 基础知识题 – 字符数组的输入与输出

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", &元素名)	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", &元素名)	cin >> &元素名
任一元素开始输出串	printf ("%s", &元素名)	cout >> &元素名



## §. 基础知识题 - 字符数组的输入与输出

### 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 //VS2019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:

- 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



## §. 基础知识题 - 字符数组的输入与输出

### 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 >> a >> b;

    cout << a << '-' << b << endl;

    return 0;
}
```

1、假设输入为abc空格def并回车  
则输出为:

```
abc def
abc-def
```

2、假设输入为abc回车  
def回车  
则输出为:

```
abc
def
abc-def
```

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\_\_

本页需填写答案





## §. 基础知识题 – 字符数组的输入与输出

### 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)

- VS2019 : 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



## §. 基础知识题 - 字符数组的输入与输出

### 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: VS2019下用gets\_s输入含空格的字符串

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

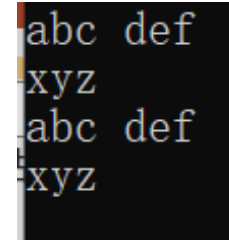
int main()
{
    char a[10], b[20];

    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：



- 2、键盘输入超过9个字符，观察



- 3、键盘先输入Hello并回车，  
再输入超过19个字符，观察  
与2一样中断

问：为什么a最长输入只能是9？  
为什么b最长输入只能是19？  
因为数组长度是10和20，结尾需  
要有'\0'，所以是9和19.

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 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;

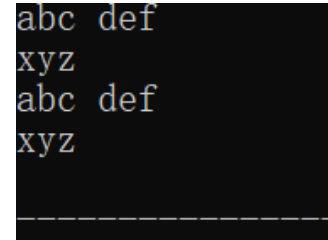
int main()
{
    char a[10], b[20];

    gets(a);
    gets(b);

    cout << a << endl;
    cout << b << endl;

    return 0;
}
```

- 1、键盘输入`abc空格def`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  
也会中断，不会输出结束
- 3、键盘先输入`Hello`并回车，  
再输入超过19个字符，观察  
也会中断，不会输出结束

问：为什么a最长输入只能是9？  
为什么b最长输入只能是19？  
因为数组长度是10和20，结尾需  
要有' \0'，所以是9和19.

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 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: VS2019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]!='\0'; i++)
        cout << int(a[i]) << ' ';
    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并回车，  
会继续等待输入，  
再输入xyz并回车  
则输出为：

```
abc def
xyz
abc def

xyz

97 98 99 32 100 101 102 10
120 121 122 10
```

问1: 和例21-22的输出区别在哪里?  
fgets读入换行符，回车键读入进换行符  
gets和gets\_s都不读入换行

问2: 后面两段红色代码的目的是什么?  
为了验证确实读入进换行，因为有ASCII码10的值出现

2、键盘输入9个字符并回车，则输出为：  
9个字符的给了a数组，a数组显示的是9个字符的ASCII码  
换行符给了b数组，b数组显示的是换行符的ASCII码10

3、如果输入28个字符并回车，则输出为：  
前9个字符给a数组，后19个字符给b数组  
这时回车不算做b的字符，因为都要补'\0'  
补完之后数组满了。

4、如果输入超过28个字符并回车，  
则输出为：依然是前9个字符给a数组  
后面19个字符给b数组，超出数量的字符不计入  
每个数组结尾都补上'\0'。

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 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]={"ABCDEFGHJKLMNOPQRSTUVWXYZ",
                  "abcdefghijklmnopqrstuvwxyz",
                  "0123456789" };
    // 单个字符输出(数组名+双下标)
    printf("a[0][2]=%c\n", a[0][2]);
    cout << "a[1][20]=" << a[1][20] << endl;

    // 字符串输出(数组名+单下标)
    printf("a[0]=%s\n", a[0]);
    cout << "a[2]=" << a[2] << endl;

    return 0;
}
```

输出为：

```
a[0][2]=C
a[1][20]=u
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ
a[2]=0123456789
```

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 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 //VS2019需要
#include <iostream>
using namespace std;

int main()
{
    char a[3][30]={"ABCDEFGHJKLMNOPQRSTUVWXYZ",
                  "abcdefghijklmnopqrstvwxyz",
                  "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 << "a[1]=" << a[1] << endl;

    return 0;
}
```

1、键盘输入#@并回车，输出为：

```
#@
a[0]=AB#DEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrst@vwxyz
```

2、键盘输入#并回车，  
输入@并回车  
输出为：

```
#
@a[0]=AB#DEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrst@vwxyz
```

本页需填写答案



## §. 基础知识题 - 字符数组的输入与输出

### 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 //VS2019需要
#include <iostream>
using namespace std;
```

```
int main()
{
    char a[3][30]={"ABCDEFGHJKLMNOPQRSTUVWXYZ",
                  "abcdefghijklmnopqrstuvwxyz",
                  "0123456789" };
```

```
    scanf("%s", a[1]); //a[1]是一维数组名, 无&
```

```
    cout << "a[0]=" << a[0] << endl;
    cout << "a[1]=" << a[1] << endl;
    cout << "a[2]=" << a[2] << endl;
```

```
    return 0;
```

```
}
```



1、输入≤29个字符，输出为：

```
abc
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ
a[1]=abc
a[2]=0123456789
```

2、输入30-59个字符，输出为

```
abcdefghijklmnopqrstuvwxyzabcdefghijkl
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyzabcdefghijkl
a[2]=efghijkl
```

输入60个以上字符，输出为：

```
abcdefghijklmnopqrstuvwxyzabcdefghijklmnpqrstuvwxyz
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyzabcdefghijklmnpqrstuvwxyz
a[2]=efghijklmnpqrstuvwxyz
```

将scanf换为 cin >> a[1];  
再重复1、2、3，观察结果

问1：输入30~59个字符为什么不出现错误？a[2]中是什么？

因为多出的字符可以有空间存放在a[2]中，a[2]中存放的是a[1]多出的字符

问2：简述你是怎么理解二维数组越界的？

因为数组存放地址是连续的，所以越界就放在下一个数组中，改变下一个一维数组的值，只有超过所有一维数组才报错

本页需填写答案





## §. 基础知识题 – 字符数组的输入与输出

### 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]={ "ABCDEFGHJKLMNOPQRSTUVWXYZ",
                    "abcdefghijklmnopqrstuvwxyz",
                    "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 << "a[1][20]=" << a[1][20] << endl;

    // (第2组) 字符串输出(&+数组名+双下标)
    printf("a[0][2]=%s\n", &a[0][2]);
    cout << "a[1][20]=" << &a[1][20] << endl;

    // (第3组) 字符串输出(数组名+单下标)
    printf("a[0]=%s\n", a[0]);
    cout << "a[2]=" << a[2] << endl;

    return 0;
}
```

输出为:

```
a[0][2]=C
a[1][20]=u
a[0][2]=CDEFGHIJKLMNOPQRSTUVWXYZ
a[1][20]=vwxyz
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ
a[2]=0123456789
```

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？

直接输出双下标的元素，就是对应的该字符  
输出该元素的地址，就是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] << endl;
```

本页需填写答案





## §. 基础知识题 - 字符数组的输入与输出

### 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 //VS2019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 << "a[0]=" << a[0] << endl;
    cout << "a[1]=" << a[1] << endl;
    cout << "a[2]=" << a[2] << endl;

    return 0;
}
```

1、输入≤26个字符，输出为：

```
abc
a[0]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcabc
a[2]=0123456789
```

2、输入27-56个字符，输出为：

```
abcdefghijklmnopqrstuvwxyzabcdefghijklmn
a[0]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyzabcdefghijklmn
a[2]=bcdefghijklmn
```

3、输入56个以上字符，输出为：

```
abcdefghijklmnopqrstuvwxyzabcdefghijkl
a[0]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyzabcdefghijkl
a[2]=bcdefghijklmnopqrstuvwxyzabcde
```

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？

因为还有a[2]中30个空间用来存储数据

a[2]中是存的第28个字符开始向后的字符串，前面30个字符填满了a[1]，'\0'被挤到最后去。

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？

因为本例中改变的是a[1][3]开始之后的值，所以前面多出了三个数据，总体在26基础上加3为29。

本页需填写答案