

§ 5. 利用数组处理批量数据

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VS2017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VS2017即可
- 5、转换为pdf后提交

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.1. 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 //VS2017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

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

scanf时, 输入AB并回车, 输出是:

-52
-52
-52
65
-52
-52
-52
-52
66
-52
-52

//用不同颜色标注出有变化的内容

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.1. 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

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

cin时, 输入AB并回车, 输出是:

-52
-52
-52
65
-52
-52
-52
66
-52
-52

//用不同颜色标注出有变化的内容

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.1. 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 //VS2017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

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

scanf时, 输入AB并回车, 输出是:

10
-52
-52
65
-52
-52
-52
66
-52
-52

//用不同颜色标注出有变化的内容

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.1. 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

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

cin时, 输入AB并回车, 表现如何?

多按几次回车, 表现如何?

最后再输入C并回车, 则输出是:

67
-52
-52
65
-52
-52
-52
66
-52
-52

//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_将回车视为一个字符输入_,

C++方式处理回车的方式是_忽略回车_

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.1. 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 //VS2017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

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

等待键盘输入, 输入Hello并回车, 输出为

72
101
108
108
111
0
-52
-52
-52
-52

//用不同颜色标注出有变化的内容

问: 1、回车是否在数组中? 否

2、Hello后面的一个字符是什么? \0, 尾零

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.1. 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 //VS2017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

-52

-52

-52

-52

-52

-52

-52

-52

-52

-52

等待键盘输入:

测试1: 输入10个字符并回车, 输出?

测试2: 输入10个以上字符并回车, 输出?

测试1:

49

50

51

52

53

54

55

56

57

48

运行时报错:

stack around the variable 'a' was occupied.

测试2: 同上

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

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.1. 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

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

等待键盘输入, 输入Hello并回车, 输出为

72
101
108
108
111
0

-52
-52
-52
-52

//用不同颜色标注出有变化的内容

问: 1、回车是否在数组中? 否

2、Hello后面的一个字符是什么? \0, 尾零

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.1. 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

-52

-52

-52

-52

-52

-52

-52

-52

-52

-52

等待键盘输入:

测试1: 输入10个字符并回车, 输出?

测试2: 输入10个以上字符并回车, 输出?

测试1:

49

50

51

52

53

54

55

56

57

48

运行时报错:

stack around the variable 'a' was occupied.

测试2: 同上

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

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.2. 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:

8
n*
d*

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.2. 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:

Student

Student

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.2. 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*

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.2. 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:

Student

Student

问: 尾零输出了吗?

没有输出

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.2. 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:

```
14
Student*
Student*
A
```

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

本页需填写答案

5.5.5.2. 输出

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)

输出为:

China烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫
China烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫

问1：为什么会有乱字符？

字符串中没有\0，
输出语句输出到内存中某处的\0为止

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，程序逐字符输出，输出5个字符后停止

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.2. 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:

烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫
烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫烫

问1: 为什么会有乱字符?

字符数组中没有\0,

输出语句输出到内存中某处的\0为止

问2: 乱字符出现几行是正常的?

一行? 多行? 或者都正常?

无论出现多少行都正常

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

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.3. 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:

dent

dent

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.3. 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 //VS2017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

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

等待键盘输入，输入Hello并回车，输出为

-52
-52
-52
72
101
108
108
111
0
-52

//用不同颜色标注出有变化的内容

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.3. 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

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

等待键盘输入，输入Hello并回车，输出为

-52
-52
-52
72
101
108
108
111
0
-52

//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_指针(&a[i])_的形式

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

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	<code>scanf("%c", &元素名)</code>	<code>cin >> 元素名</code>
输入字符串	<code>scanf("%s", 数组名)</code>	<code>cin >> 数组名</code>
输出单个字符	<code>printf("%c", 字符变量名)</code>	<code>cout >> 字符变量名</code>
输出字符串	<code>printf("%s", 数组名)</code>	<code>cout >> 数组名</code>
任一元素开始输入串	<code>scanf("%s", &a[i]);</code>	<code>cin >> &a[i]</code>
任一元素开始输出串	<code>printf("%s\n", &a[i]);</code>	<code>cout << &a[i]</code>

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.4. 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 //VS2017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:

abc-def

2、假设输入为abc回车
def回车

则输出为:

abc-def

结论: 空格是__B__

- A. 输入串中的合法字符
- B. 输入分隔符

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.4. 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

2、假设输入为abc回车
def回车

则输出为：

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__

§ 5. 利用数组处理批量数据

5.5.5. 字符数组的输入与输出

5.5.5.4. 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

- VS2017 : gets_s, 无gets
- CodeBlocks: gets, 无gets_s
- Dev C++ : gets, 无gets_s
- Linux C++ : gets有warning, 无gets_s
- 四个编译器均能使用fgets函数, 函数原型为
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

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.4. 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21：VS2017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：

abc def
xyz

2、键盘输入超过9个字符，观察
运行时报错：“Buffer is too small”

3、键盘先输入Hello并回车，
再输入超过19个字符，观察
运行时报错：“Buffer is too small”

问：为什么a最长输入只能是9？
a[10]最多只能输入9个字符，最后有一个\0
为什么b最长输入只能是19？
b[20]最多只能输入19个字符，最后有一个\0

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.4. 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22: CodeBlocks/DevC++/Linux下用gets输入含空格的字符串

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

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

    gets(a);
    gets(b);

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

    return 0;
}
```

//Linux有编译警告，但可以继续运行

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

2、键盘输入超过9个字符，观察
程序正常输出了所有输入的字符，
但有时程序的返回值为0xC0000005

3、键盘先输入Hello并回车，
再输入超过19个字符，观察
程序第一行可能会输出错误的字符，
第二行输出正确

问：为什么a最长输入只能是9？

a[10]最多只能输入9个字符，最后有一个\0

为什么b最长输入只能是19？

b[20]最多只能输入19个字符，最后有一个\0

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.4. 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23：四个编译器下均可用fgets输入含空格的字符串

```
#include <iostream>
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`

`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？

程序将回车也作为输入写入了字符数组

问2：后面两段红色代码得目的是什么？

检查字符数组中是否含有回车

2、键盘输入`9个字符`并回车，则输出为：

`123456789`

`49 50 51 52 53 54 55 56 57`

`10`

3、如果输入`28个字符`并回车，则输出为：

`123456789`

`0123456789012345678`

`49 50 51 52 53 54 55 56 57`

`48 49 50 51 52 53 54 55 56 57 48 49 50 51 52 53 54 55 56`

4、如果输入`超过28个字符`并回车，

则输出为：

`123456789`

`0123456789012345678`

`49 50 51 52 53 54 55 56 57`

`48 49 50 51 52 53 54 55 56 57 48 49 50 51 52 53 54 55 56`

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.5. 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]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
a[2]=0123456789
```

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.5. 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 //VS2017需要
#include <iostream>
using namespace std;

int main()
{
    char a[3][30]={"ABCDEFGHJKLMNOPQRSTUVWXYZ",
                  "abcdefghijklmnopqrstuvwxyz",
                  "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]=abcdefghijklmnopqrstuvwxyz@vwxyz

2、键盘输入#并回车，
输入@并回车
输出为：

a[0]=AB#DEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyz@vwxyz

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.5. 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
//VS2017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：

a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ

a[1]=1234567890

a[2]=0123456789

2、输入30-59个字符，输出为：

a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ

a[1]=1234567890123456789012345678901234567890

a[2]=1234567890

3、输入60个以上字符，输出为：

a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ

a[1]=12345678901234567890123456789012345678901234

a[2]=1234567890123456789012345678901234567890

运行时报错：

Stack around the variable

‘a’ was corrupted

将scanf换为 cin >> a[1];

再重复1、2、3，观察结果

结果与1、2、3相同

问1：输入30~59个字符为什么不

出现错误？a[2]中是什么？

31~59这些多余的字符将a[2]覆盖了，

因此a[2]是这些多余的字符

问2：简述你是怎么理解二维数组

越界的？

二维数组本质上为一维数组，越界时会覆盖其之后的字符

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.5. 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]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
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？

```
printf("c", a[i][j]);
cout << a[i][j];
```

```
printf("s", &a[i][j]);
cout << &a[i][j];
```

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][0]=%s\n", &a[0][0]);
cout << "a[2][0]=" << &a[2][0] << endl;
```

§ 5. 利用数组处理批量数据

本页需填写答案

5.5.5. 字符数组的输入与输出

5.5.5.5. 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
//VS2017需要
#include <iostream>
using namespace std;
int main()
{
    char a[3][30]={
        "ABCDEFGHJKLMNOPQRSTUVWXYZ",
        "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;
}
```

a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ

a[1]=abc*****

a[2]=0123456789

2、输入27-56个字符，输出为：

a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ

a[1]=abcdefghijklmnopqrstuvwxyz

a[2]=*****

3、输入56个以上字符，输出为：

a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ

a[1]=abcdefghijklmnopqrstuvwxyz

a[2]=*****

运行时报错：

Stack around the variable

‘a’ was corrupted

将scanf换为 cin >> &a[1][3];

再重复1、2、3，观察结果

结果与1、2、3相同

问1：输入27~56个字符为什么不

出现错误？a[2]中是什么？

28~56这些多余的字符将a[2]覆盖了，

因此a[2]是这些多余的字符

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？

本例从a[1][3]开始覆盖字符，

与例26相比少了三个字符