



§. 基础知识题 – 字符数组的输入与输出

学号：_1953609_ 班级：_19软件_ 姓名：_王灏廷_

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，右下角有标注，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 **(5.5前)**

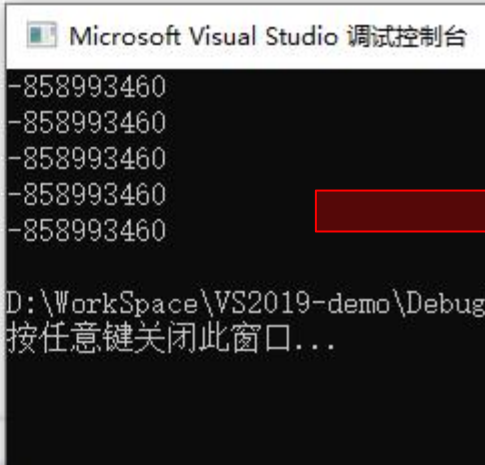


§. 基础知识题 - 字符数组的输入与输出

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

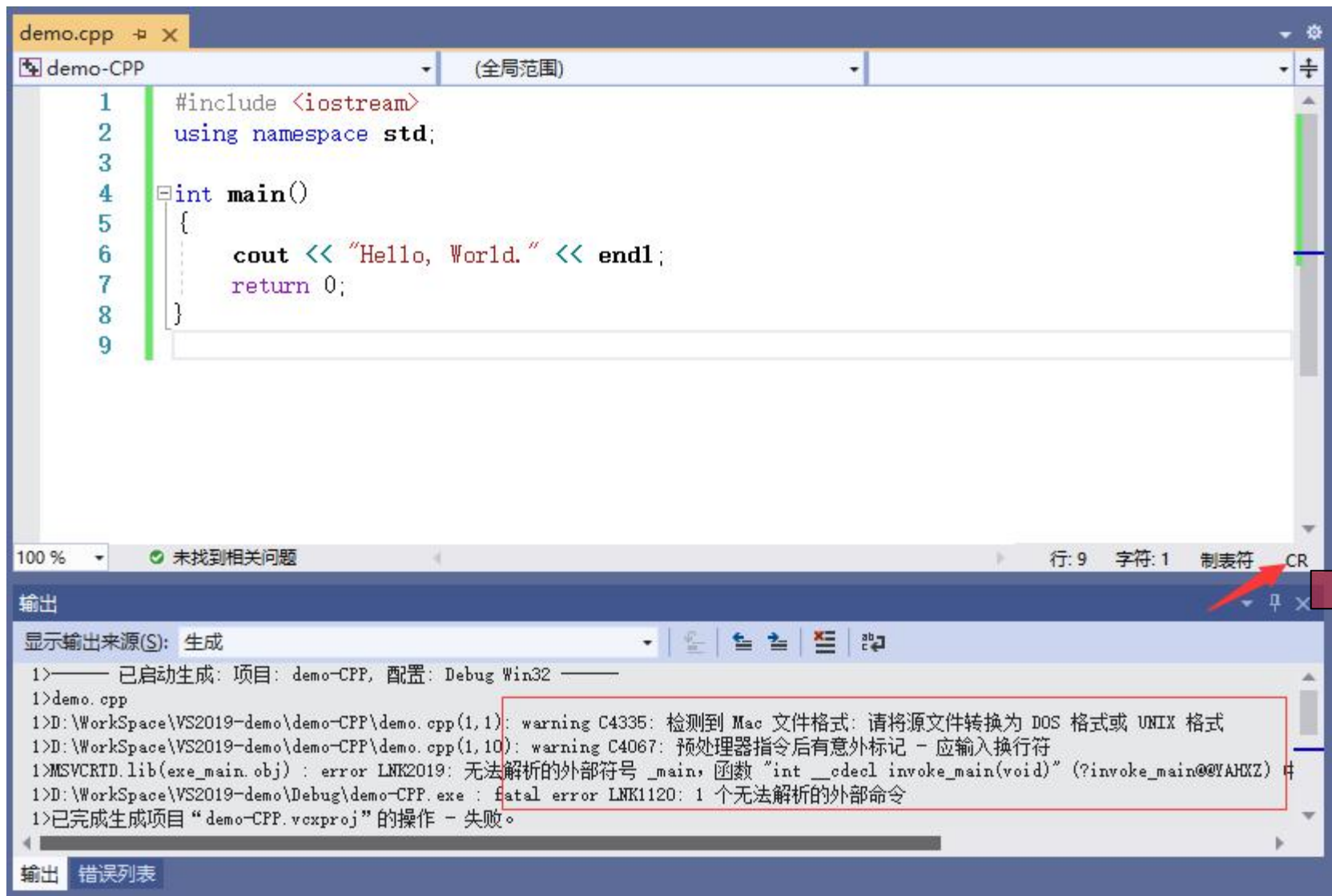


§. 基础知识题 - 字符数组的输入与输出

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

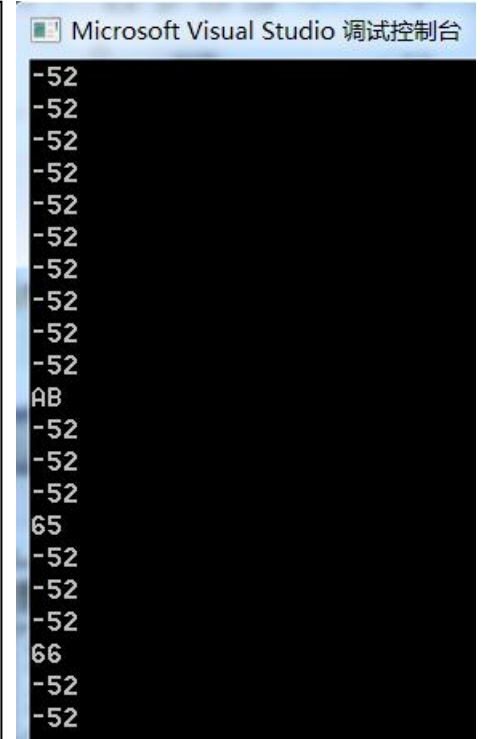
scanf前首先输出10行, 内容是:

-52
-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
66
-52
-52

//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&

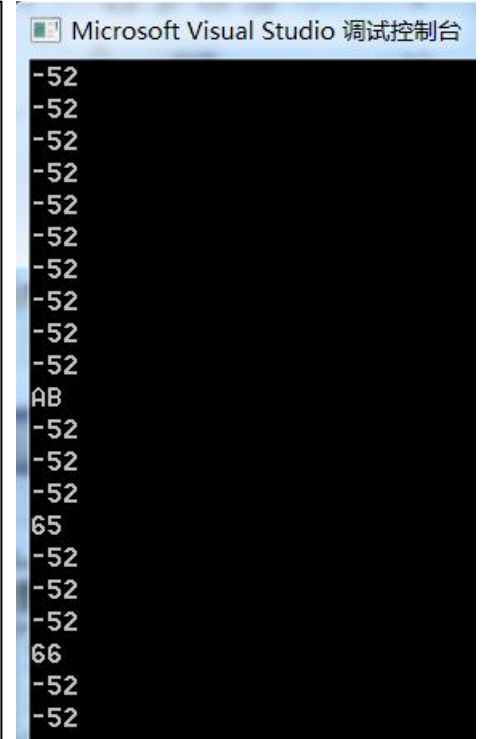
cin前首先输出10行, 内容是

-52
-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



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

-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

Microsoft Visual Studio 调试控制台

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

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

-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:

输入AB并回车, 仍处于等待输入状态。
多按几次回车, 没有改变, 仍然处于等待输入状态。

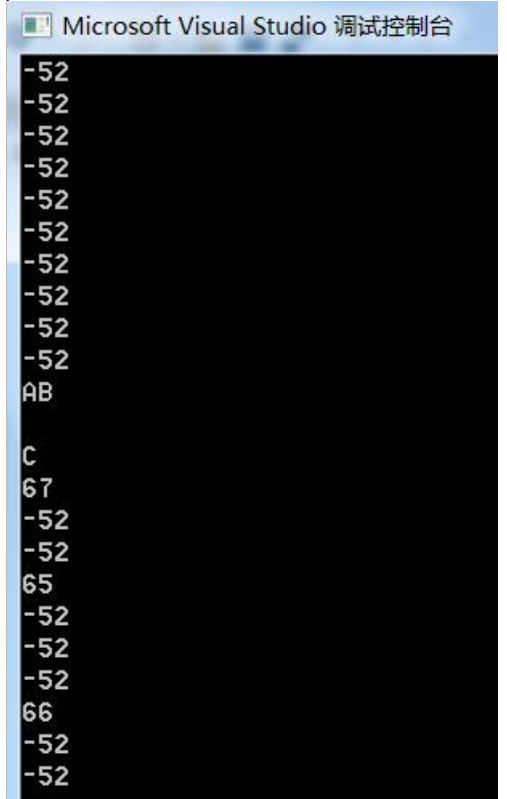
67
-52
-52
65
-52
-52
-52
66
-52
-52

//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__按照字符读入__,

C++方式处理回车的方式是__当做输入结束的标志, 不会作为字符本身读入。__



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

-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? 回车不在数组中

2、Hello后面的一个字符是什么? '\0'
即ASCII码“0”这个对应的字符

Microsoft Visual Studio 调试控制台

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

本页需填写答案

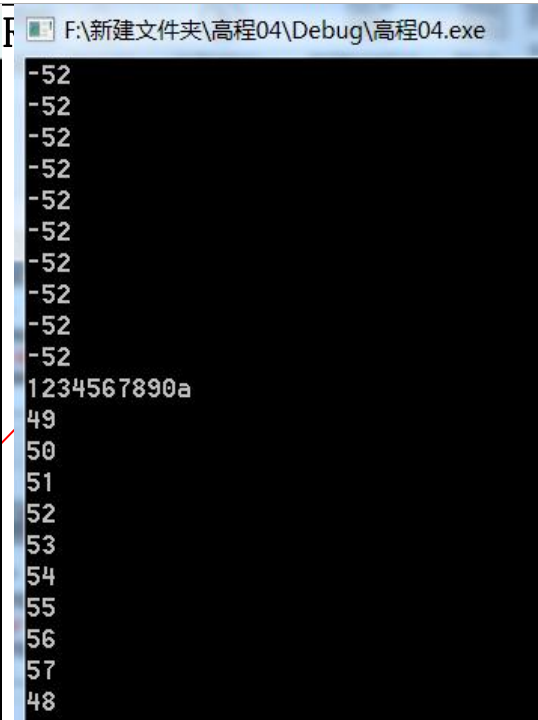
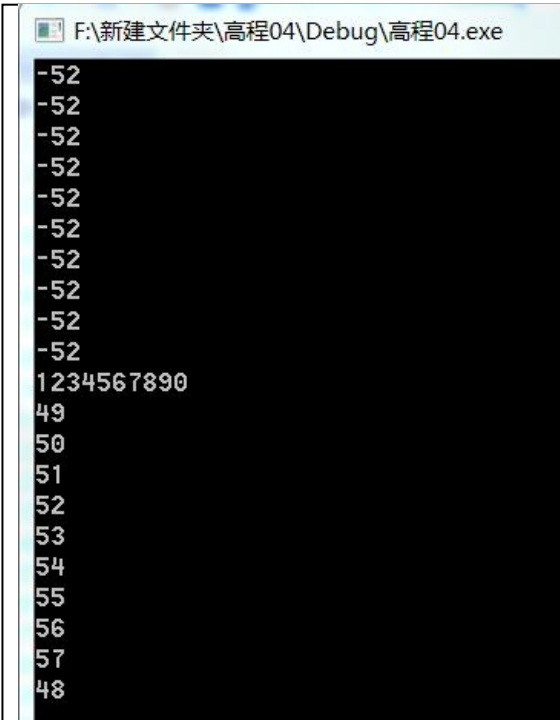


§ . 基础知识题 - 字符数组的输入与输出

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)



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

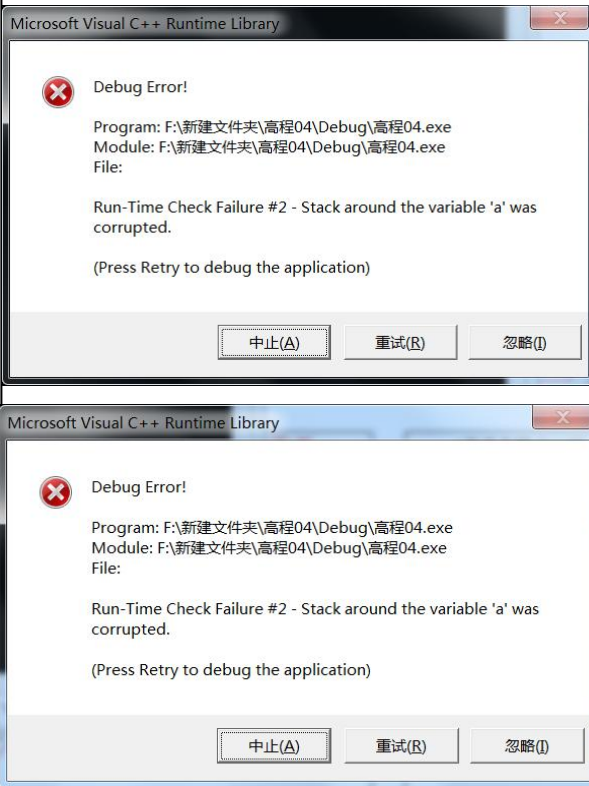
scanf前首先输出10行, 内容是

-52
-52
-52
-52
-52
-52
-52
-52
-52
-52
-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?

1. 输出这10个字符的ascii码, 但会报错
2. 输出前10个字符的ascii码, 但也会报错

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

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

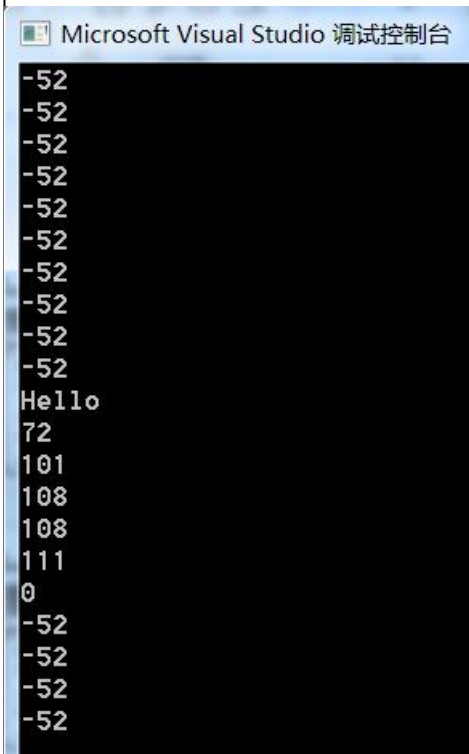
等待键盘输入, 输入Hello并回车, 输出为

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? 回车不在数组中

2、Hello后面的一个字符是什么? '\0'
即ASCII码“0”这个对应的字符



本页需填写答案

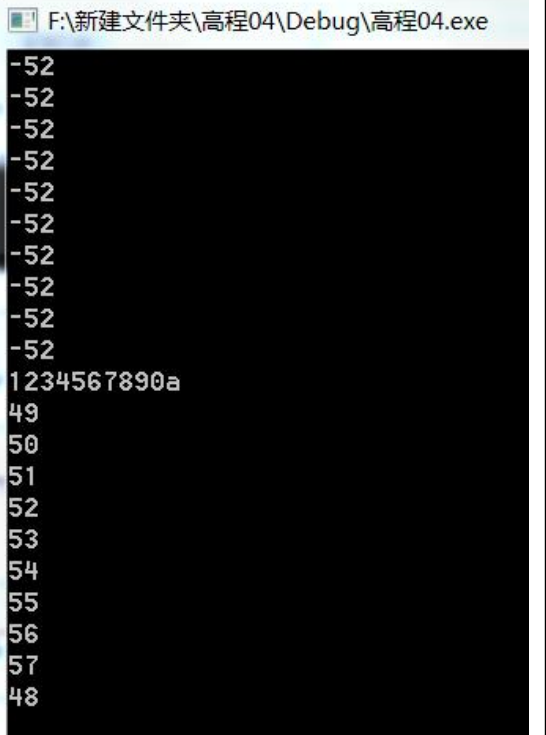
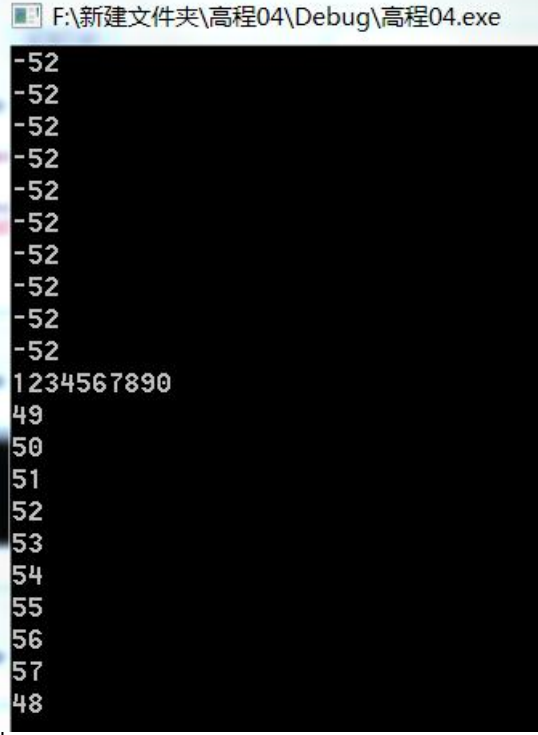


§ . 基础知识题 - 字符数组的输入与输出

1. 输入

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

例8: C++方式输入字符串(错误)



```
cout << int(a[i]) << endl;  
  
return 0;  
}
```

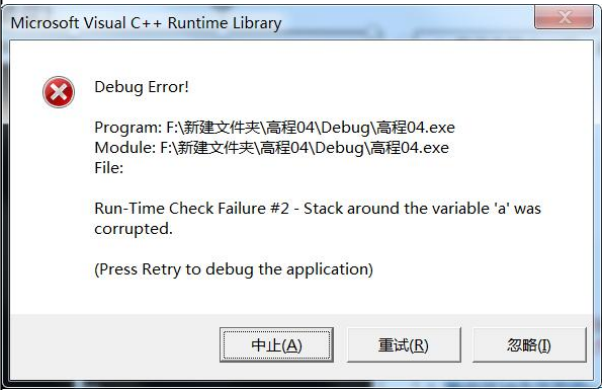
cin前首先输出10行, 内容是

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

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?

1. 输出这10个字符的askii码, 但会报错
2. 输出前10个字符的askii码, 但也会报错

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;

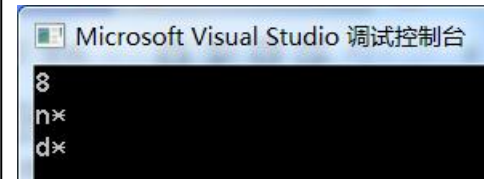
    printf("%c*\n", a[5]);

    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:

8
n*
d*





§. 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:

Student
Student

Microsoft Visual Studio 调试控制台
Student
Student

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

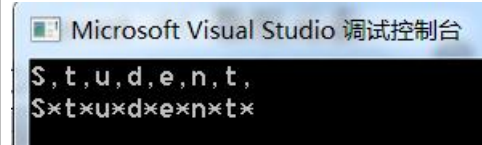
    for(i=0; i<7; i++)
        cout << a[i] << '*';
    cout << endl; //换行

    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:

Student

Student

Microsoft Visual Studio 调试控制台

Student
Student

问: 尾零输出了吗? 如何证明?

尾零没有输出

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

```
int main()
```

```
{
```

```
    char a[] = "Student";
```

```
    printf("%s\n", a);
```

```
    cout << a << endl;
```

```
    cout << a[7] << endl;
```

```
    printf("%c\n", a[7]);
```

```
    return 0; 已用时间 <= 3ms
```

```
}
```

F:\新建文件夹\高程04\Debug\高程04.exe

Student
Student
a
a
-

若输出尾零, 则“t”后应该还有字符, 但在输出整个字符数组时并没有输出, 可见尾零并没有跟随整个数组输出

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;

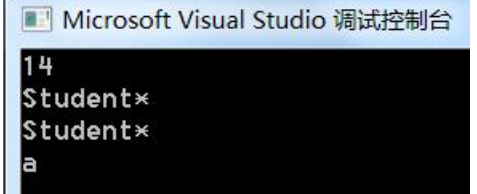
    printf("%s*\n", a);
    cout << a << '*' << endl;

    cout << a[12] << endl;

    return 0;
}
```

输出为:

```
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?
最后还有隐含的\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__遇到'\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:

China烫烫烫藁阂?鰩
China烫烫烫藁阂?鰩

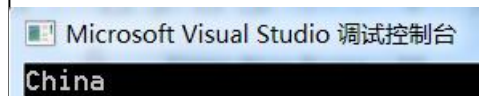
问1: 为什么会有乱字符?

在本来是尾0的位置赋予了其他字符, 在输出时遇不到尾0, 所以随机输出, 故而输出乱字符。

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?
看不到乱字符了。
因为仅仅依次输出前五项, 没有尾0的影响, 所以China后的影响不会输出出来。



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

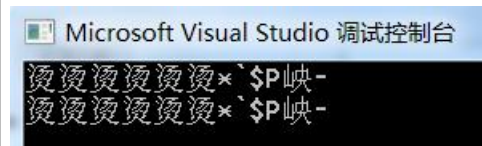
    printf("%s\n", a);
    cout << a << endl;

    return 0;
}
```

输出为:

**** (乱字符)**

**** (乱字符)**



问1: 为什么会有乱字符?

字符数组没有初始化,
故而在输出时输出随机乱字符

问2: 乱字符出现几行是正常的?
一行? 多行? 或者都正常?

都正常, 因为不含尾0
输出完全随机

结论: 不能字符串形式输出不含
__尾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:

dent
dent

Microsoft Visual Studio 调试控制台

dent
dent

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

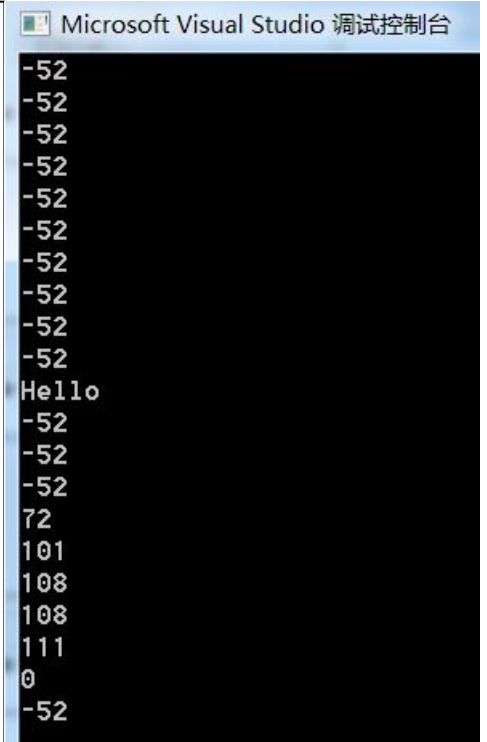
scanf先输出10行, 内容是

-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

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

//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

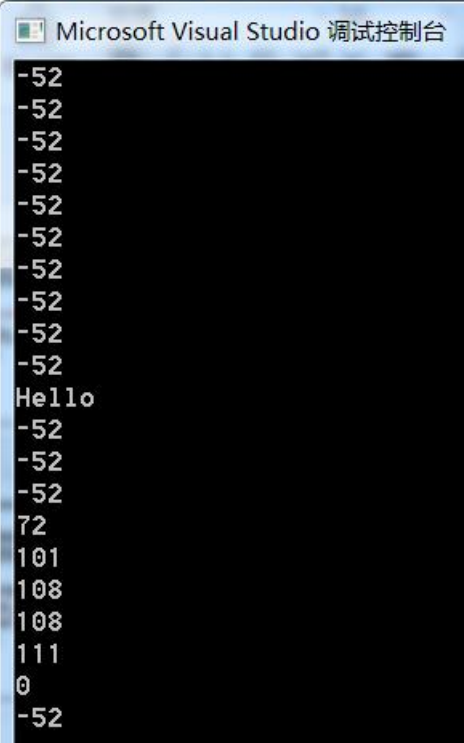
-52
-52
-52
-52
-52
-52
-52
-52
-52
-52

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__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", 字符数组名)	cin >> 字符数组名
输出单个字符	printf("%c", 元素名)	cout << 元素名
输出字符串	printf("%s", 字符数组名)	cout << 字符数组名
任一元素开始输入串	scanf("%s", &字符数组名[元素序号])	cin >> &字符数组名[元素序号];
任一元素开始输出串	printf("%s", &字符数组名[元素序号])	cout << &字符数组名[元素序号];



§. 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];

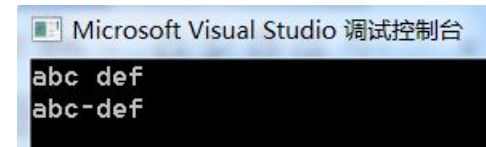
    scanf("%s%s", a, b);

    printf("%s-%s\n", a, b);

    return 0;
}
```

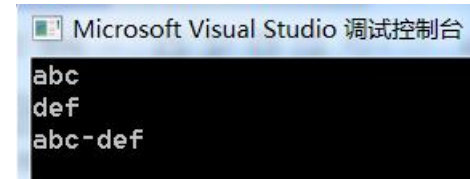
1、假设输入为abc空格def并回车
则输出为:

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

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

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];

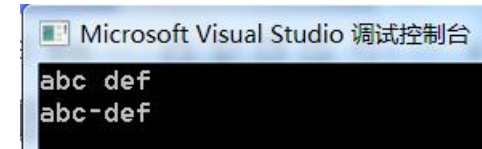
    cin >> a >> b;

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

    return 0;
}
```

- 1、假设输入为`abc空格def`并回车
则输出为:

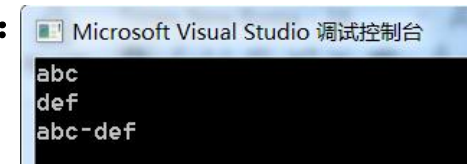
`abc-def`



- 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_输入分隔符_ (如空格、回车)

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:

```
#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;
}
```

```
Microsoft Visual Studio 调试控制台
abc def
xyz
abc def
xyz
```

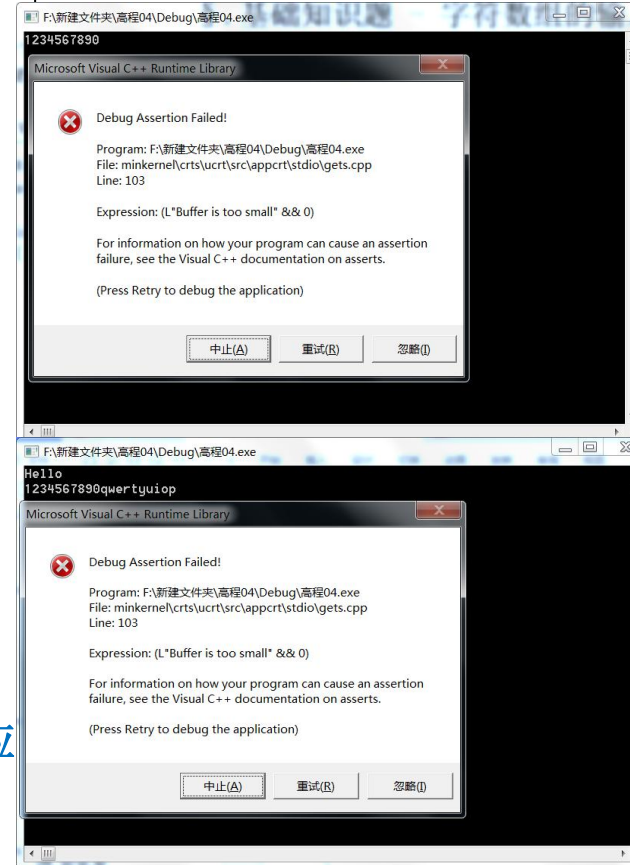
1、键盘输入`abc空格def`并回车, 会继续等待输入, 再输入`xyz`并回车 则输出为:

`abc def`
`xyz`

2、键盘输入超过9个字符, 观察报错

3、键盘先输入`Hello`并回车, 再输入超过19个字符, 观察报错

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?
a数组内存为10, 最后一个元素应是"`\0`", 所以前面最长输入只能是9。
b数组内存为20, 最后一个元素应是"`\0`" 所以前面最长输入只能是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;
}
```

```
Hello
1234567890qwertyuiopasdf
asdf
1234567890qwertyuiopasdf
```

```
Hello
acjknadkjcndajkcnhadsjkcasdncjkndaj
sjkcasdncjkndaj
acjknadkjcndajkcnhadsjkcasdncjkndaj
```

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
```

`abc def`
`xyz`

2、键盘输入超过9个字符，观察
仍在等待输入

```
1234567890
_
```

3、键盘先输入`Hello`并回车，
再输入超过19个字符，观察
不能正确输出`Hello`
但第二行所有字符被完整输出了
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？
a数组内存为10，最后一个元素应
是"`\0`"，
所以前面最长输入只能是9。
b数组内存为20，最后一个元素应
是"`\0`"
所以前面最长输入只能是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
    for(
    cout
    for(i=0; b[i]!='\0'; i++)
    cout
    ret
}
```



- 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?
本题输出了换行符和空格的askii码而例21-22没有
问2: 后面两段红色代码的目的是什么?
输出a和b中除" \0" 之外的所有字符的askii码便于观察输入情况
- 2、键盘输入`9个字符`并回车，则输出为:
9个字符和换行符的askii码
- 3、如果输入`28个字符`并回车，则输出为:
28个字符的askii码
第一行9个，代表数组a元素
第二行19个，代表数组b元素
- 4、如果输入`超过28个字符`并回车，则输出为:
前28个字符的askii码
第一行9个，代表数组a元素
第二行19个，代表数组b元素，
多余的未输出



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
```

```
选定 Microsoft Visual Studio 调试控制台
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",
                  "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]=abcdefghijklmnopqrst@vwxyz

```
Microsoft Visual Studio 调试控制台
#@
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

```
Microsoft Visual Studio 调试控制台
#
@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

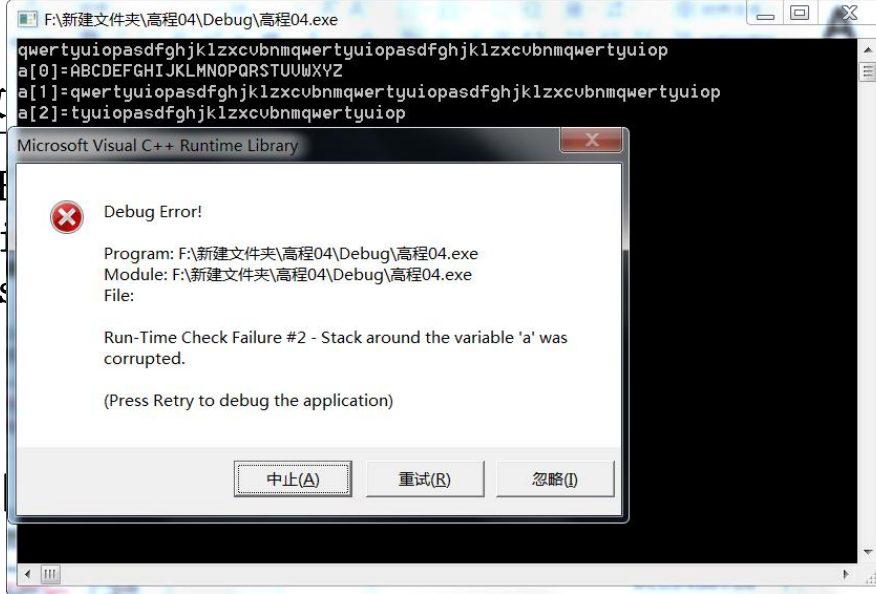
```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
using namespace std;
```

```
int main()
{
    char a[3][100];
```

```
    scanf("%s", a[0]);
```

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

```
    return 0;
}
```



1、输入 ≤ 29 个字符，输出为：

$a[0]$ 和 $a[2]$ 不变

$a[1]$ 为输入的字符

2、输入30-59个字符，输出为：

$a[0]$ 不变

$a[1]$ 为输入的字符

$a[2]$ 为输入字符与30的差值部分

3、输入60个以上字符，输出为：

报错

将scanf换为 `cin >> a[1];`

再重复1、2、3，观察结果

结果与上相同

问1：输入30~59个字符为什么不

出现错误？ $a[2]$ 中是什么？

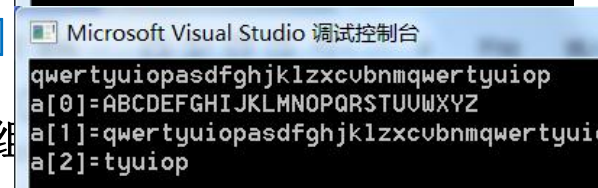
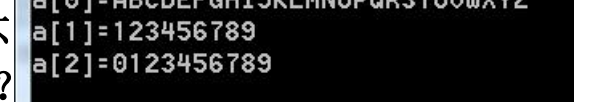
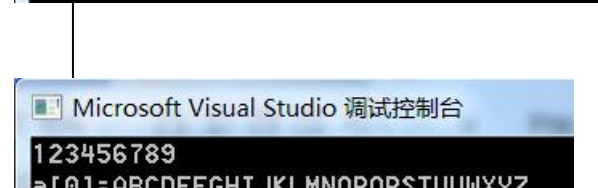
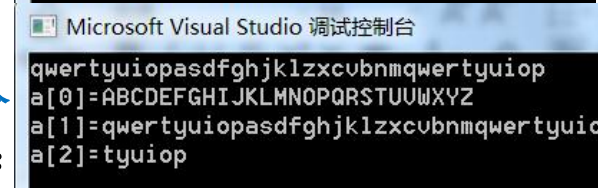
超过30个的部分被输入进了 $a[2]$

$a[2]$ 中是超过30个之后的字符

问2：简述你是怎么理解二维数组

越界的？

在输入元素大于列数时，多余部分会输入进下一行的列元素中，直到其后所有行元素都被输入后，此时若仍有多余部分，则二维数组越界



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;
}
```

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

输出为:

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

使用数组名+双下标输出单个字符
使用&+数组名+双下标输出字符串

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？

将第2组输出的列标改为0，即可得到一致结果

本页需填写答案

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



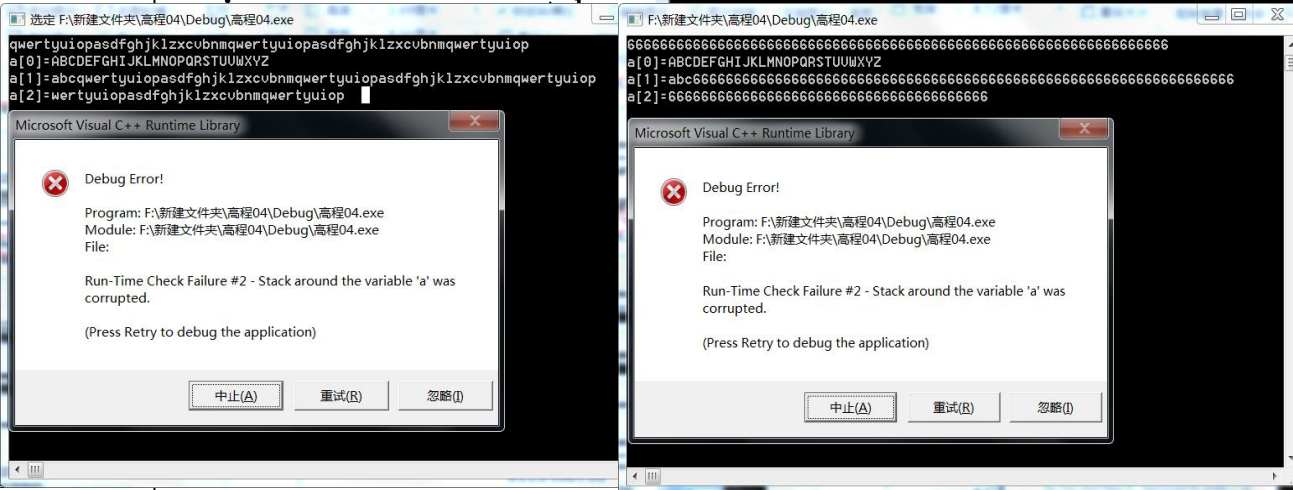

§. 基础知识题 - 字符数组的输入与输出

5. 二维字符数组的输入/输出

★ 数组名加双下标表示元素，单下标表示行

例28：二维字符数组从任一位置

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
```



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

return 0;
}
```

```
Microsoft Visual Studio 调试控制台
qwertyuiopasdfghjklzxcvbnm
a[0]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyz
a[2]=0123456789
```

```
Microsoft Visual Studio 调试控制台
qwertyuiopasdfghjklzxcvbnmqwer
a[0]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyz
a[2]=wer
```

1、输入≤26个字符，输出为：

a[1]前三项元素不变

a[1][3]之后的a[1]元素变为输入

2、输入27-56个字符，输出为：

a[1]前三项元素不变

a[1][3]之后的a[1]元素变为输入的字符

a[2]输出多余的输入字符。

3、输入56个以上字符，输出为：
报错

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

结果与上相同

问1：输入27~56个字符为什么不出现错误？a[2]中是什么？

超过27个的部分被输入进了a[2]
a[2]中是超过27个之后的字符

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？a[1]总空间为30

例26是从a[1][0]开始，a[1]中列元素还有29个
本例是从a[1][3]开始，a[1]中列元素只有26个

```
Microsoft Visual Studio 调试控制台
qwertyuiop
a[0]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyz
a[2]=0123456789
```

```
Microsoft Visual Studio 调试控制台
qwertyuiopasdfghjklzxcvbnmqwer
a[0]=ABCDEFGHIJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyz
a[2]=wer
```

本页需填写答案