



§. 基础知识题 - 字符数组的输入与输出

要求:

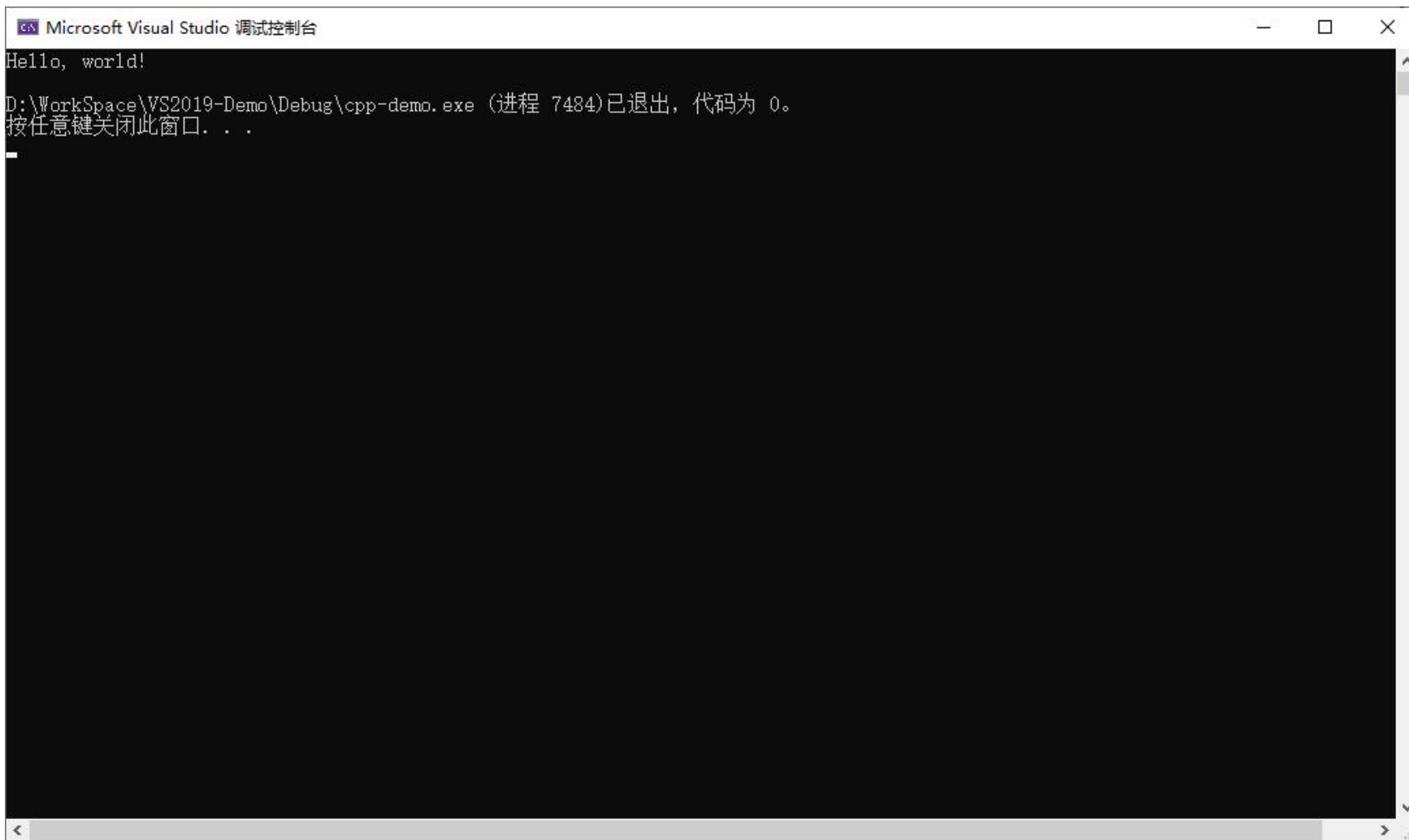
- 1、完成本文档中所有的题目并写出分析、运行结果，体会字符数组输入输出时不同用法的差异
- 2、题目明确指定编译器外，缺省使用VS2022即可
 - ★ 如果要换成其他编译器，可能需要自行修改头文件适配
 - ★ 部分代码编译时有warning，不影响概念理解，可以忽略
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
 - ★ 贴图要有效部分即可，不需要全部内容
 - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
 - ★ **不允许**手写在纸上，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、**11月23日前**网上提交本次作业（在“文档作业”中提交）

§. 基础知识题 - 字符数组的输入与输出

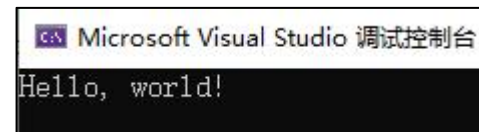


贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

例：无效贴图

A screenshot of the Microsoft Visual Studio debug console window. The window is titled "Microsoft Visual Studio 调试控制台". It contains the text "Hello, world!" followed by a newline, and then "D:\Workspace\VS2019-Demo\Debug\cpp-demo.exe (进程 7484)已退出, 代码为 0." followed by another newline and "按任意键关闭此窗口. . .". The window is large, showing a significant portion of the screen.

例：有效贴图

A screenshot of the Microsoft Visual Studio debug console window, cropped to show only the "Hello, world!" text. The window title "Microsoft Visual Studio 调试控制台" is visible at the top.



§. 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
```

Microsoft Visual Studio 调试控制台

```
-858993460
-858993460
-858993460
-858993460
-858993460
D:\Workspace\VS2019-demo\Debug\
按任意键关闭此窗口...
```

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

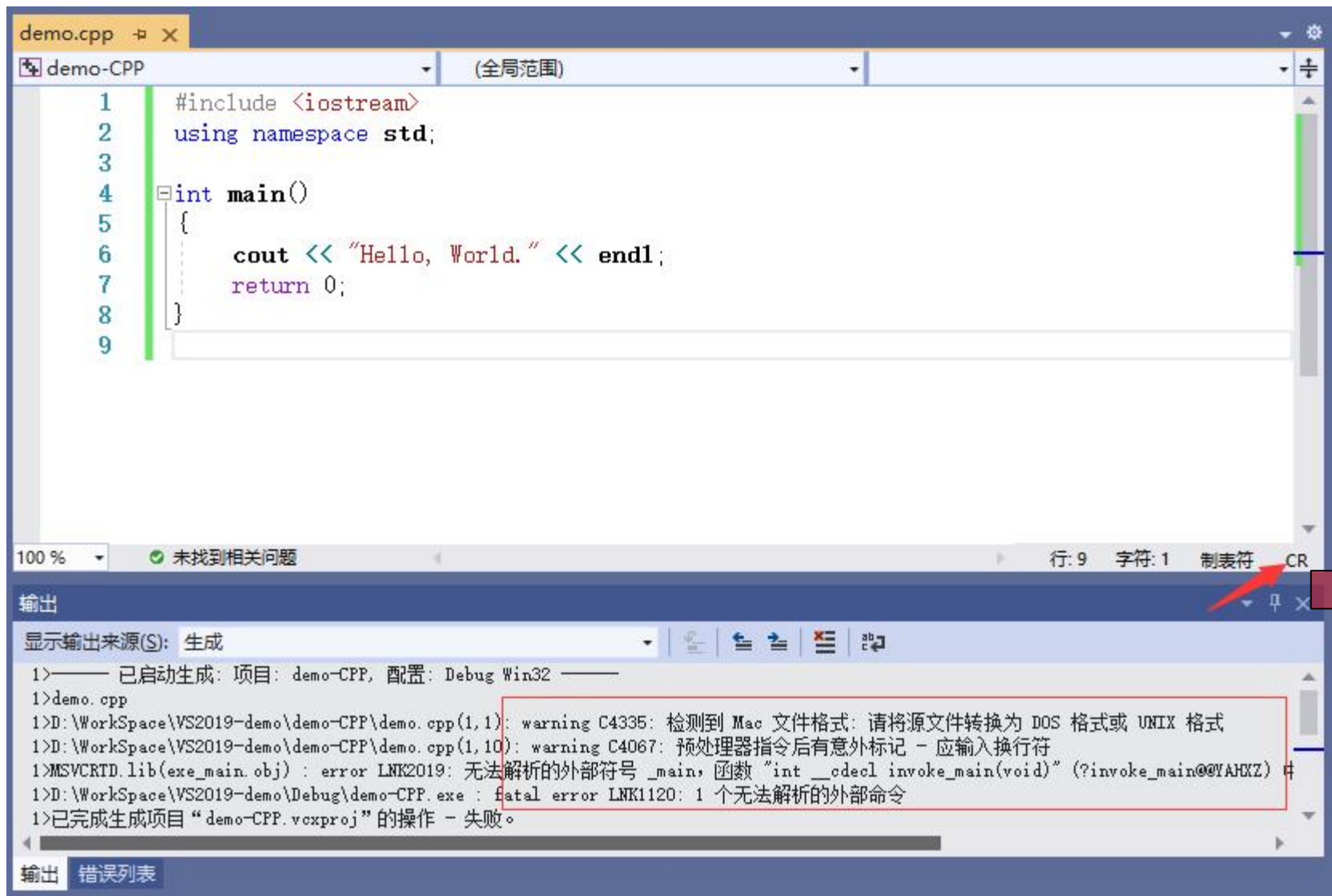


§. 基础知识题 - 字符数组的输入与输出

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

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

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 //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;

    scanf("%s", a);

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

    return 0;
}
```

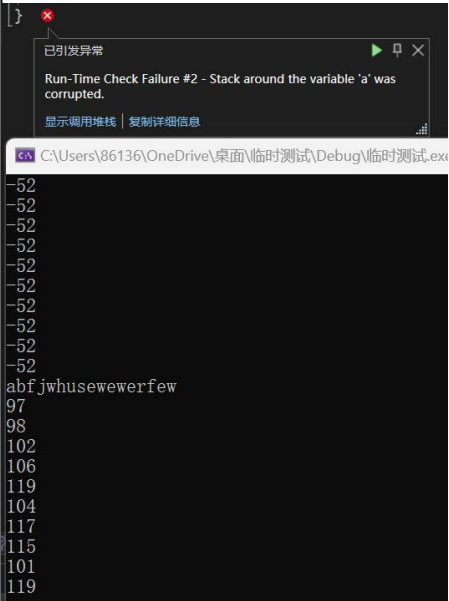
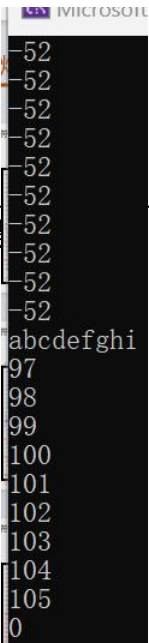
直接数组名，无下标，
也不加&
因为C/C++规定，数组名
代表数组的起始地址

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:
测试1: 输入9个及以下字符并回车，输出?
输出键盘输入的字符
测试2: 输入10个及以上字符并回车，输出?

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? 否

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

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";

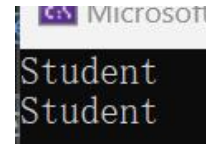
    for(i=0; i<7; i++)
        printf("%c", a[i]);
    cout << endl; //换行

    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



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

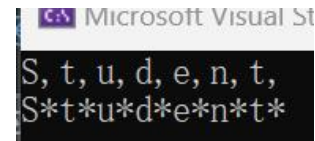
    for(i=0; i<7; i++)
        cout << a[i] << '*';
    cout << endl; //换行

    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

问: 尾零输出了吗? 如何证明?

没有

在紧接着字符串输出后再输出一个符号,
如“#”, 该符号紧跟着先前的字符串

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char a[] = "Student";
7
8      printf("%s\n", a);
9      cout << a << '#' << endl;
10
11     return 0;
12 }
13
```

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

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? 是
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_第一个\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烫烫烫帖_x0018_联1

China烫烫烫帖_x0018_联1



问1: 为什么会有乱字符?

数组缺少\0, 导致在输出中发生了数组越界, 输出了数组合法范围外的随机数据

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, 因为不会发生数组越界, 不会输出数组

合法范围之外的随机数据

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:

烫烫烫烫烫?
愿渣/_x0001_??
烫烫烫烫烫?
愿渣/_x0001_?

烫烫烫烫烫?
愿渣/_x0001_??
烫烫烫烫烫?
愿渣/_x0001_?

问1: 为什么会有乱字符?

数组未初始化, 且缺少\0, 导致了
输出了数组内/外的随机数据

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

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：
dent
dent



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 //VS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

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

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

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

等待键盘输入，输入Hello并回车，输出为

//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_&a[n]_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" , &数组名[元素序号]) | cin >> &数组名[元素序号] |
| 任一元素开始输出串 | printf("%s" , &数组名[元素序号]) | cout << &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 //VS需要
#include <iostream>
#include <cstdio>
using namespace std;

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

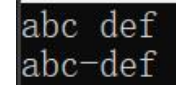
    scanf("%s%s", a, b);

    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:

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__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)

- 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



§ . 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: VS下用gets_s输入含空格的字符串

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

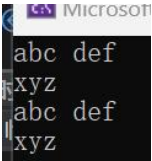
int main()
{
    char a[10], b[20];

    gets_s(a);
    gets_s(b);

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

    return 0;
}
```

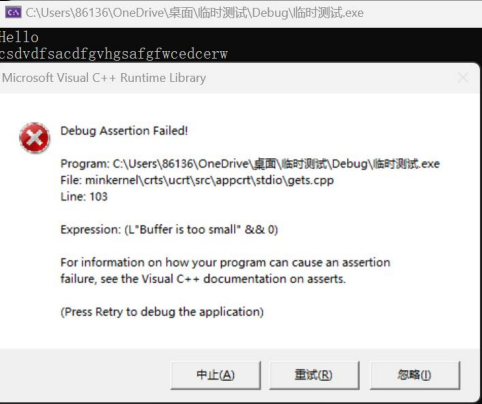
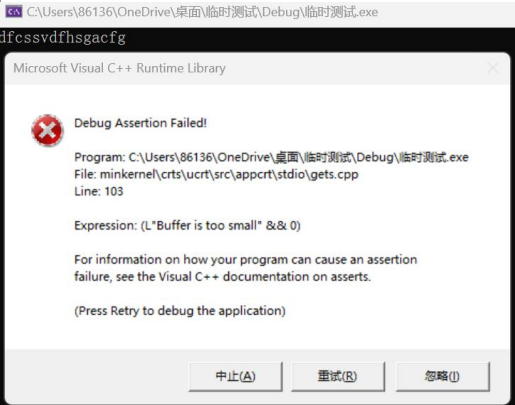
- 1、键盘输入abc空格def并回车，会继续等待输入，再输入xyz并回车，则输出为：



- 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

问：为什么a最长输入只能是9？
因为数组a的长度为10

为什么b最长输入只能是19？
因为数组b的长度为20



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;

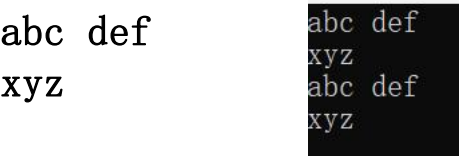
int main()
{
    char a[10], b[20];

    gets(a);
    gets(b);

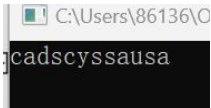
    cout << a << endl;
    cout << b << endl;

    return 0;
}
```

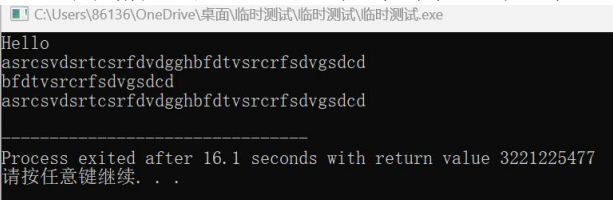
1、键盘输入abc空格def并回车，
会继续等待输入，
再输入xyz并回车
则输出为：



2、键盘输入超过9个字符，观察



3、键盘先输入Hello并回车，
再输入超过19个字符，观察



问：为什么a最长输入只能是9？因为数组a的长度为10
为什么b最长输入只能是19？因为数组b的长度为20

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: 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]!='\0'; i++)
        cout << int(a[i]) << ' ';
    cout << endl;

    for(i=0; b[i]!='\0'; i++)
        cout << int(b[i]) << ' ';
    cout << endl;

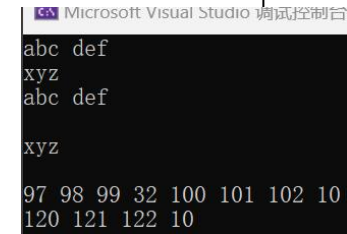
    return 0;
}
```

- 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

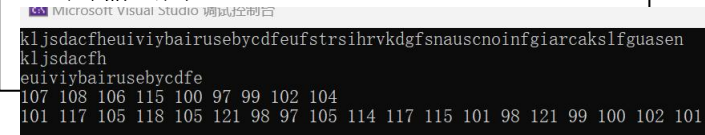
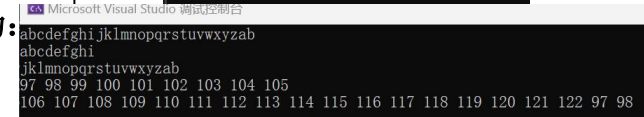
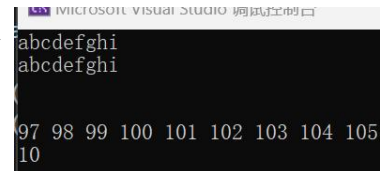


- 问1: 和例21-22的输出区别在哪里?
回车被读取并再次输出了
- 问2: 后面两段红色代码的目的是什么?
输出实际读取到的字符数据，以显示不可见的字符

- 2、键盘输入9个字符并回车，则输出为：

- 3、如果输入28个字符并回车，则输出为：

- 4、如果输入超过28个字符并回车，
则输出为：



本页需填写答案



§. 基础知识题 - 字符数组的输入与输出

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
```

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

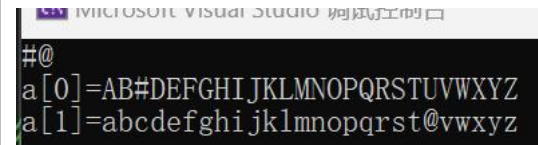
int main()
{
    char a[3][30]={"ABCDEFGHJKLMNOPQRSTUVWXYZ",
                  "abcdefghijklmnopqrstvwxyz",
                  "0123456789" };

    // 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&

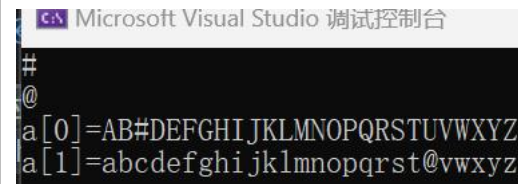
    // 字符串输出(数组名+单下标)
    printf("a[0]=%s\n", a[0]);
    cout << "a[1]=" << a[1] << endl;

    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 //VS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;
}
```

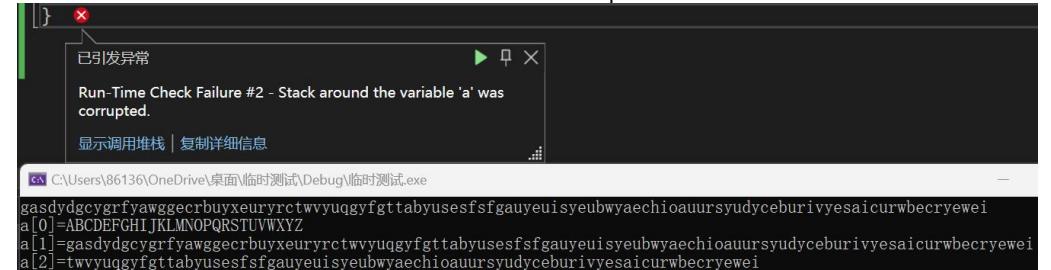
1、输入≤29个字符，输出为：

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

2、输入30-59个字符，输出为：

```
abcDEFGHIJKLMNOPQRSTUVWXYZ1234567890fsxda
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyz1234567890fsxda
a[2]=567890fsxda
```

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

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

```
abcdefghijklmnopqrstuvwxyz1425656233
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ
a[1]=abcdefghijklmnopqrstuvwxyz1425656233
a[2]=656233
```

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a[1]和a[2]在内存中连续放置，超过30但是少于60的字符串超过a[1]的部分实际上落在了a[2]的区域内

问2：简述你是怎么理解二维数组

越界的？超过的部分存于数组的下一行

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？

不加&输出单个字符

加&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？

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

```
a[0][2]=C
a[1][20]=u
a[0][2]=CDEFGHIJKLMNOPQRSTUVWXYZ
a[1][20]=vwxyz
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]={"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;
```

```
}
```

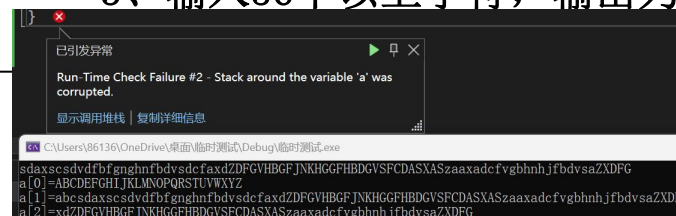
1、输入 ≤ 26 个字符，输出为：

2、输入27-56个字符，输出为：

3、输入56个以上字符，输出为：

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

```
abcdefghijklmnopqrstuvwxyz12345  
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ  
a[1]=abcabcdefghijklmnopqrstuvwxyz12345  
a[2]=2345
```



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



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

```
abcdefghijklmnopqrstuvwxyz12345  
a[0]=ABCDEFGHJKLMNOPQRSTUVWXYZ  
a[1]=abcabcdefghijklmnopqrstuvwxyz12345  
a[2]=2345
```

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a[1]和a[2]在内存中连续放置，超过30但是少于60的字符串，超过a[1]的部分实际上落在了a[2]的区域内

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，

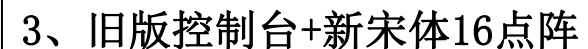
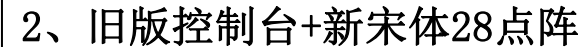
差别在哪？

区别在例28的输入地址是跳过了前三个字符元素，从第四个字符元素位置开始

本页需填写答案

例29：在不同的控制台及字体设置下尾零输出的差异

1、新版控制台+新宋体28点阵



- 本页需填写答案



§. 基础知识题 - 字符数组的输入与输出

6. 尾零的输出

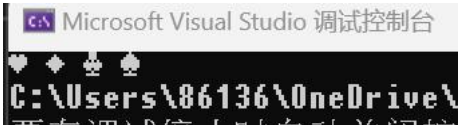
例30：在不同的控制台及字体设置下其它非图形字符输出的差异

（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）

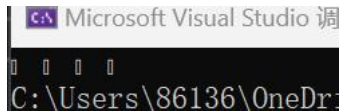
```
#include <iostream>
using namespace std;

int main()
{
    char s[]="\003 \004 \005 \006";
    cout << s;
    return 0;
}
```

1、某版控制台+某字体/某点阵
（此处找到一种可显示的）



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）



结论：
上页的结论1也__适用__（适用/不适用）
于其它非图形字符

本页需填写答案