



## §. 基础知识题 – 循环结构

要求:

- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 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、**10月7日前**网上提交本次作业（在“文档作业”中提交）



## §. 基础知识题 – 循环结构

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

```
Microsoft Visual Studio 调试控制台
Hello, world!
D:\Workspace\VS2019-Demo\Debug\cpp-demo.exe (进程 7484)已退出, 代码为 0。
按任意键关闭此窗口...
```

例：有效贴图

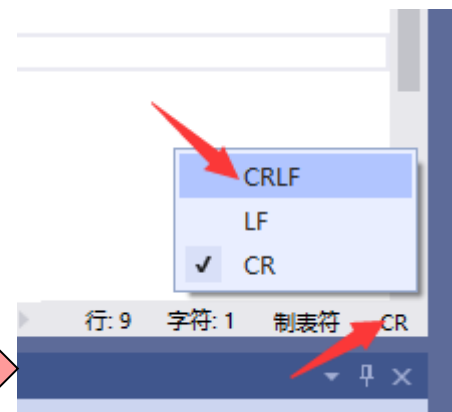
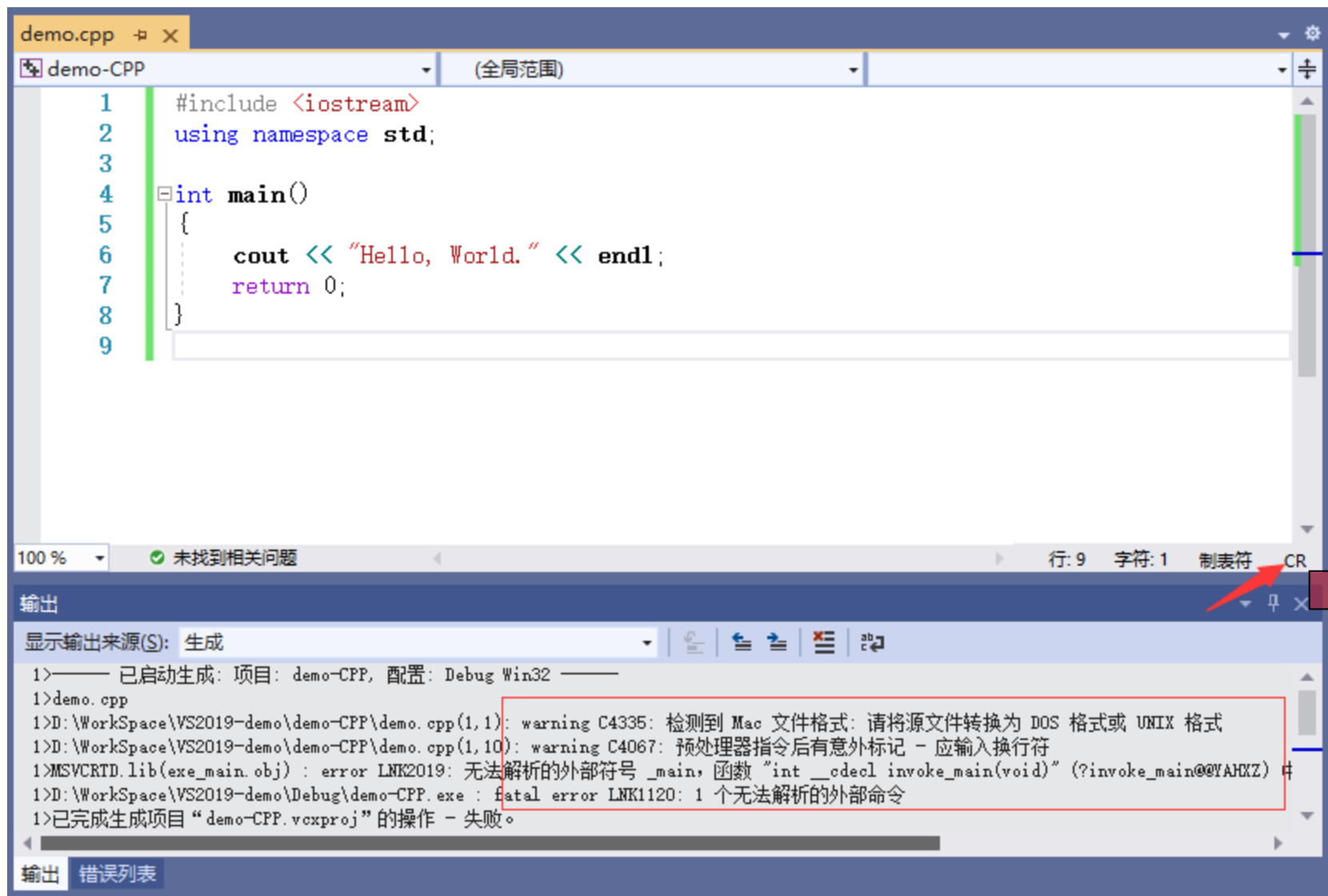
```
Microsoft Visual Studio 调试控制台
Hello, world!
```



## §. 基础知识题 – 循环结构

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 – 循环结构

### 1、循环的嵌套

A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

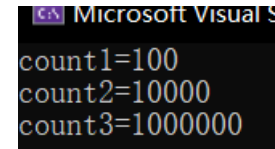
```
#include <iostream>
using namespace std;

int main()
{
    int i, j, k;
    int count1 = 0, count2 = 0, count3 = 0;

    for(i=1; i<=100; i++) {
        ++count1;
        for(j=1; j<=100; j++) {
            ++count2;
            for(k=1; k<=100; k++)
                ++count3;
        }
    }

    cout << "count1=" << count1 << endl;
    cout << "count2=" << count2 << endl;
    cout << "count3=" << count3 << endl;
    return 0;
}
```

#### 1、贴运行结果



```
Microsoft Visual S
count1=100
count2=10000
count3=1000000
```

#### 2、当循环嵌套时，内层循环的执行次数和外层循环是什么关系？

内层循环总次数等于外层循环\*内层循环  
外层循环每开始执行一次循环，都要内层循环  
执行到循环体结束，外层循环才开始执行下一  
次循环。



## §. 基础知识题 – 循环结构

### 1、循环的嵌套

B. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

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

int main()
{
    int i, j, k;
    int count1 = 0, count2 = 0, count3 = 0;

    for(i=1; i<=100; i++) {
        ++count1;
        for(j=i; j<=100; j++) {
            ++count2;
            for(k=j; k<=100; k++)
                ++count3;
        }
    }

    cout << "count1=" << count1 << endl;
    cout << "count2=" << count2 << endl;
    cout << "count3=" << count3 << endl;
    return 0;
}
```

### 1、贴运行结果

```
count1=100
count2=5050
count3=171700
```

### 2、当循环嵌套时，内层循环的执行次数和外层循环是什么关系？

$j=1+2+\dots+100;$

$k=1+(1+2)+(1+2+3)+\dots+(1+2+\dots+100);$

外层循环每开始执行一次循环，都要内层循环执行到循环体结束，外层循环才开始执行下一次循环。



## §. 基础知识题 – 循环结构

### 1、循环的嵌套

C. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
#include <conio.h>
using namespace std;

int main()
{
    int i, j, count = 0;
    for(i=1; i<=100; i++) {
        for(j=1; i<=100; j++) {
            ++count;
            if (count % 1000 == 0) {
                cout << "*";
                _getch();
            }
        }
    }

    cout << "count = " << count << endl;
    return 0;
}
```

**//注意：这个程序无法通过按CTRL+C终止**

1、贴运行结果（能表现出要表达的意思即可）  
任何按钮输进去都是\*，而且程序不结束

2、按内外for循环的执行步骤依次分析，为什么会得到这个结果？

例：第1步 – 外循环表达式1 – i=1

...

第x步 – 内循环表达式3 – j=4

**注：具体内容瞎写的，不要信；步骤写到能得到结论即可**

第1步 – 外层循环for(i=1; i<=100; i++)，第一个i=1

第2步 – 内层循环for(j=1; i<=100; j++)，i=1，一直循环下去

第3步 if (count % 1000 == 0) {

cout << "\*";

\_getch();

反复执行里面的循环，就是一直打印\*\*\*\*\*，不管输入任何内容

## §. 基础知识题 – 循环结构



此页不要删除，也没有意义，仅仅为了分隔题目



## §. 基础知识题 – 循环结构

### 2、break与continue

A. 已知代码如下，回答问题

```
while(1) {  
    ①  
    ②  
    if (X)  
        continue;  
    ③  
    ④  
}
```

当X为真时，重复执行\_12\_\_\_\_\_ (①②③④)  
当X为假时，重复执行\_1234\_\_\_\_\_ (①②③④)

```
for(1; 1; ④) {  
    ①  
    ②  
    if (X)  
        continue;  
    ③  
}
```

当X为真时，重复执行\_124\_\_\_\_\_ (①②③④)  
当X为假时，重复执行\_1234\_\_\_\_\_ (①②③④)





## §. 基础知识题 – 循环结构

### 2、break与continue

B. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

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

int main()
{
    int i=0, sum=0;

    while(i<1000) {
        i++;
        break;
        sum=sum+i;
    }

    cout << "i=" << i << endl;
    cout << " sum=" << sum << endl;

    return 0;}
```

i=1  
sum=0

//问题1: 循环执行了多少次?

执行了一次, 到break停止

//问题2: sum=sum+i执行了多少次?

执行了0次, 被跳过了

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

int main()
{
    int i=0, sum=0;

    while(i<1000) {
        i++;
        continue;
        sum=sum+i;
    }

    cout << "i=" << i << endl;
    cout << " sum=" << sum << endl;
    return 0;
}
```

i=1000  
sum=0

//问题1: 循环执行了多少次?

执行了1000次

//问题2: sum=sum+i执行了多少次?

执行了0次, 被continue跳过了

## §. 基础知识题 – 循环结构



此页不要删除，也没有意义，仅仅为了分隔题目



## §. 基础知识题 – 循环结构

### 3、观察程序运行结果

A. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

```
#include <iostream>
#include <iomanip>    //格式输出
#include <cmath>      //fabs
#include <windows.h>  //取系统时间
using namespace std;
```

```
int main()
{
```

```
    int s=1;
    double n=1, t=1, pi=0;
```

```
    LARGE_INTEGER tick, begin, end;
    QueryPerformanceFrequency(&tick);    //取计数器频率
    QueryPerformanceCounter(&begin);      //取初始硬件定时器计数
```

```
    while(fabs(t)>1e-6) {
        pi=pi+t;
        n=n+2;
        s=-s;
        t=s/n;
    }
```

```
    QueryPerformanceCounter(&end); //获得终止硬件定时器计数
```

```
    pi=pi*4;
    cout << "n=" << setprecision(10) << n << endl;
    cout<<"pi="<<setiosflags(ios::fixed)<<setprecision(9)<<pi<< endl;
```

```
    cout << "计数器频率: " << tick.QuadPart << "Hz" << endl;
    cout << "时钟计数 : " << end.QuadPart - begin.QuadPart << endl;
    cout << setprecision(6) << (end.QuadPart - begin.QuadPart)/double(tick.QuadPart) << "秒" <<endl;
```

```
    return 0;
}
```

用下面的迭代公式求Pi的值

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

(1) n, t, pi为double型

精度为1e-6: n=1000001 pi=3.141590654 时间=0.003245(秒)

1e-7: n=10000001 pi=3.141592454 时间=0.033691(秒)

1e-8: n=100000001 pi=3.141592634 时间=0.277597(秒)

1e-9: n=1000000001 pi=30141592652 时间=2.735825(秒)

(因为机器配置不同，时间值可能不同)

本页结果手填

(2) n, t, pi为float型

精度为1e-6: n=1000001 pi=3.141593933 时间=0.023315(秒)

1e-7: n=10000001 pi=3.141596556 时间=0.206788(秒)

1e-8: n=\_\_无结果\_\_ pi=\_\_ \_\_ 时间=\_\_ \_\_ (秒)

问：7项中哪个没结果？为什么？

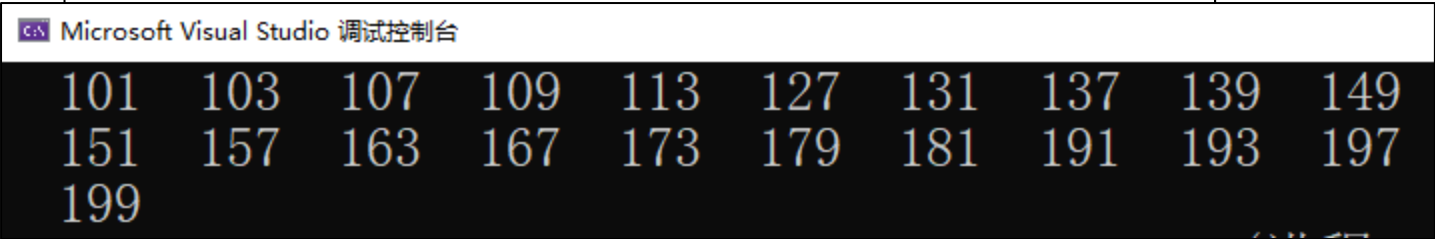
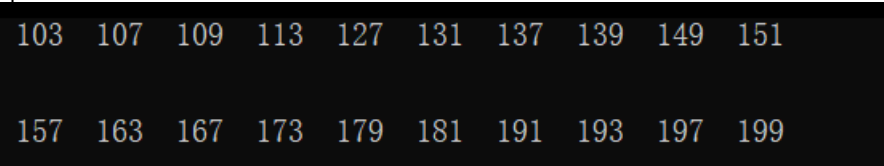
1e-8没有结果，因为float的精度（六位）无法达到1e-8，程序陷入死循环



## §. 基础知识题 – 循环结构

### 3、观察程序运行结果

B. 观察下列程序的运行结果，回答问题并将程序的运行结果截图贴上(如果有错则贴错误信息截图)

<pre>#include &lt;iostream&gt; #include &lt;iomanip&gt;</pre>	<h3>打印100-200之间的素数</h3>	(1) 目前输出结果：一共21个，每10个一行
<pre>#include &lt;iostream&gt; #include &lt;iomanip&gt; using namespace std; int main() {     int n = 0, i, m, k;     bool prime;     for (m = 101; m &lt;= 200; m += 2) { //偶数没必要判断         prime = true;                //对每个数，先认为是素数         k = int(sqrt(m));         for (i = 2; i &lt;= k; i++)             if (m % i == 0) {                 prime = false;                 break;             }          if (prime) {             cout &lt;&lt; setw(5) &lt;&lt; m;             n = n + 1;                //计数器，只为了加输出换行              if (n % 10 == 0)          //每10个数输出一行                 cout &lt;&lt; endl;         }     } //end of for      return 0; }</pre>	<p>数</p>  <p>(2) 将m的初值从101改为103，应该是20个，共2行实际呢？为什么？</p>  <p>实际上是共四行，其中第一和第四行是数据，第二和第三行是空行。原因：原来的程序中执行完m=151后，n=10，执行cout&lt;&lt;endl;后面执行m=153，m=155时，n的值未改变，语句cout&lt;&lt;endl;依然每次都执行，所以会出现两个空行。</p>	
<p>return 0; }把if放到里面就行了</p>	<p>(3) 将左侧程序改正确 (正确程序贴图在左侧，覆盖现有内容即可)</p>	

## §. 基础知识题 – 循环结构



此页不要删除，也没有意义，仅仅为了分隔题目