



## §. 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、**4月2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

A screenshot of the Microsoft Visual Studio debug console window. The window title is "Microsoft Visual Studio 调试控制台". The output text is:

```
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. . .
```

The screenshot is too large, capturing the entire window and its scrollbars, which is considered an invalid screenshot according to the requirements.

例：有效贴图

A screenshot of the Microsoft Visual Studio debug console window, showing only the output text:

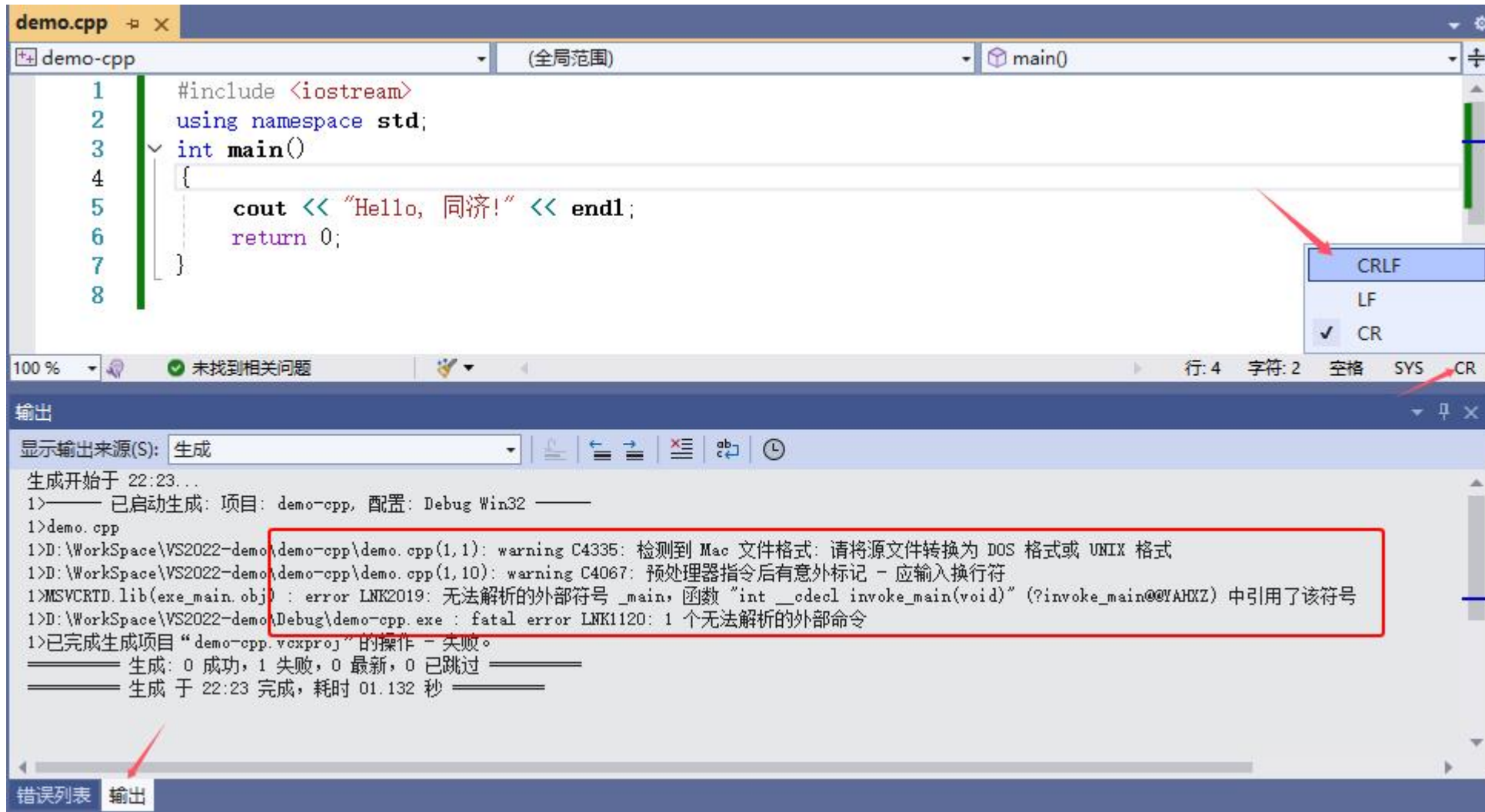
```
Hello, world!
```

The screenshot is cropped to show only the relevant output, which is considered a valid screenshot.



## §. 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



特别提示:

- ★ 本次作业的答案，除特别提示外，上课全讲过，课件上都有!!!
- ★ 作业本质就是对上课内容及课件的review(因为读懂程序的逻辑很重要)
- ★ 对上课接受程度较好的同学，可能有点重复/多余，但还得做



## §. 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

```
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？

内层循环的总执行次数由两层循环次数的乘积决定，且内层循环会随外层循环的每次迭代重新开始一轮完整执行。



## §. 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

```
C:\> Microsoft Visual Studio 调试
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？

内层循环的执行次数=外层循环的循环次数×内层循环的循环次数



## §. 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)

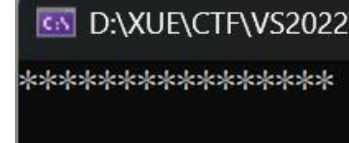
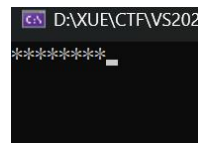
```
#include <iostream>
#include <conio.h>
using namespace std;

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，要关窗口**

1、贴运行结果（能表现出要表达的意思即可）



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 - 外循环表达式1 - i=1

第2步 - 外循环表达式2 - i<=100

第3步 - 外循环表达式3 - i++

第4步 - 内循环表达式1 - j=1

第5步 - 内循环表达式2 - i<=100

第6步 - 内循环表达式3 - j++

第7步 - 内循环表达式1 - j=2

第8步 - 内循环表达式2 - i<=100

第9步 - 内循环表达式3 - j++

.....

## §. 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\_\_④①②\_\_ (①②③④)  
当X为假时，重复执行\_\_④①②③\_\_ (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;
```

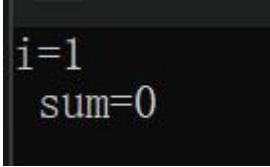
```
    cout << " sum=" << sum << endl;
```

```
    return 0;
```

```
}
```

//问题1: 循环执行了多少次? 1次

//问题2: sum=sum+i执行了多少次? 0次



```
i=1
sum=0
```

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

```
int main()
{
```

```
    int i=0, sum=0;
```

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

```
    cout << "i=" << i << endl;
```

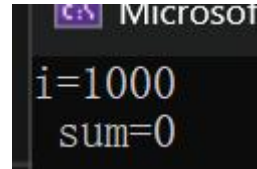
```
    cout << " sum=" << sum << endl;
```

```
    return 0;
```

```
}
```

//问题1: 循环执行了多少次? 1000次

//问题2: sum=sum+i执行了多少次? 0次



```
i=1000
sum=0
```

## §. 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=0.004634(秒)
1e-7: n=10000001	pi=3.141592454	时间=0.014489(秒)
1e-8: n=100000001	pi=3.141592634	时间=0.149422(秒)
1e-9: n=1000000001	pi=3.141592652	时间=1.474909(秒)

(因为机器配置不同, 时间值可能不同)

(2) n, t, pi为float型

精度为1e-6: n=1000001	pi=3.141593933	时间=0.017261(秒)
1e-7: n=10000001	pi=3.141596556	时间=0.144389(秒)
1e-8: n=_____	pi=_____	时间=_____ (秒)

问: 1、7项中哪个没结果? 为什么?

float型精度为1e-8没有, 因为1e-8超过了float的精度范围。

2、float和double同进度下那个时间快? (观察现象即可, 不需要解释原因)  
double快。

**本页结果不要截图, 手填即可**



## §. 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>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>是素数</p> <p>行</p>	<div data-bbox="1057 392 2484 628"><p>Microsoft Visual Studio 调试控制台</p><pre>101  103  107  109  113  127  131  137  139  149 151  157  163  167  173  179  181  191  193  197 199</pre></div> <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> <div data-bbox="1126 792 2000 963"><pre>103  107  109  113  127  131  137  139  149  151 157  163  167  173  179  181  191  193  197  199</pre></div> <p>最后一个if在上一个if之外，无论是否有素数加入，当n%10==0成立时都会换行，由于149到151是连在一起的，不会出现这种情况，而151和157相差6，也就出现了这种情况。</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