

§. 基础知识题



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
- 4、转换为pdf后提交
- 5、**9月15日前**网上提交本次作业（在“文档作业”中提交）



§. 基础知识题

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 captures the entire window, including the title bar and the full output text, 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:

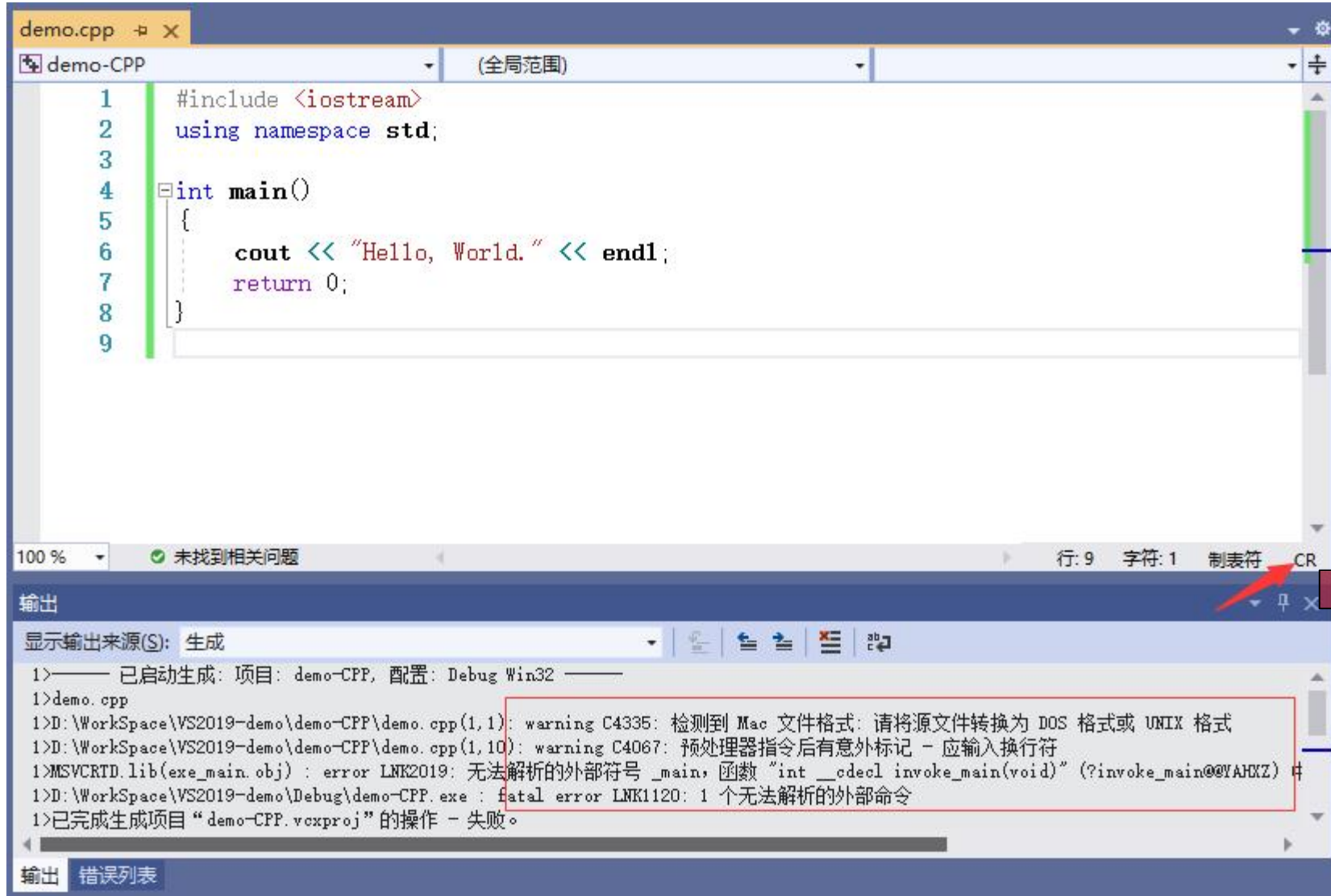
```
Hello, world!
```

The screenshot is cropped to focus on the output text, which is considered a valid screenshot according to the requirements.



§. 基础知识题

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

1、十进制整数转二进制补码（仿照课件PDF的P. 22，写出具体步骤，包括绝对值、取反、+1）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001”或“11010100-00110001”）

A. -106 （假设为1字节整数，其中进制互转部分，直接写答案即可，不需要竖式除法/按权展开相加，下同）

原数	绝对值原码	取反	补码
-106	01101010	10010101	<div>10010101</div> <div>+)</div> <div>1</div> <div>10010110</div>



§. 基础知识题

1、十进制整数转二进制补码（仿照课件PDF的P. 22，写出具体步骤，包括绝对值、取反、+1）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001”或“11010100-00110001”）

B. -206 （假设为2字节整数）

原数	绝对值原码	取反	补码
-206	00000000 11001110	11111111 00110001	<div>11111111 00110001 +) 1 ----- 11111111 00110010</div>



§ . 基础知识题

1、十进制整数转二进制补码（仿照课件PDF的P. 22，写出具体步骤，包括绝对值、取反、+1）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001” 或 “11010100-00110001”）

C. -206 （假设为4字节整数）

原数	绝对值原码
-206	00000000 00000000 00000000 11001110
取反	补码
11111111 11111111 11111111 00110001	11111111 11111111 11111111 00110001
	+) 1
	11111111 11111111 11111111 00110010



§ . 基础知识题

1、十进制整数转二进制补码（仿照课件PDF的P. 22，写出具体步骤，包括绝对值、取反、+1）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001” 或 “11010100-00110001”）

D. -4095 （假设为4字节整数）

原数	绝对值原码
-4095	00000000 00000000 00001111 11111111
取反	补码
11111111 11111111 11110000 00000000	11111111 11111111 11110000 00000000 +) 1 11111111 11111111 11110000 00000001



§ . 基础知识题

1、十进制整数转二进制补码（仿照课件PDF的P. 22，写出具体步骤，包括绝对值、取反、+1）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001” 或 “11010100-00110001”）

E. 你的学号对应的int型十进制负数（例1：1234567 => -7654321 / 1234000 => -4321）

原数	绝对值原码
2152118 => -8112512	00000000 01111011 11001001 10000000
取反	补码
11111111 10000100 00110110 01111111	11111111 10000100 00110110 01111111 +) 1 <hr/> 11111111 10000100 00110110 10000000



§. 基础知识题

2、二进制补码转十进制整数（只考虑有符号数，写出具体步骤，包括-1、取反、绝对值、加负号）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001”或“11010100-00110001”）

A. 1011 0111

原数	-1	取反	绝对值	加负号
1011 0111	$\begin{array}{r} 1011\ 0111 \\ -) \qquad 1 \\ \hline 1011\ 0110 \end{array}$	0100 1001	73	-73



§ . 基础知识题

2、二进制补码转十进制整数（只考虑有符号数，写出具体步骤，包括-1、取反、绝对值、加负号）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001”或“11010100-00110001”）

B. 1101 1101 1101 1110

原数	-1	取反	绝对值	加负号
1101 1101 1101 1110	<div>1101 1101 1101 1110 -) 1 1101 1101 1101 1101</div>	0010 0010 0010 0010	8738	-8738



§ . 基础知识题

2、二进制补码转十进制整数（只考虑有符号数，写出具体步骤，包括-1、取反、绝对值、加负号）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001” 或 “11010100-00110001”）

C. 1111 1111 1111 1111 1111 1110 1010 1110

原数	-1	
11111111 11111111 11111110 10101110	11111111 11111111 11111110 10101110 -) 1	
	11111111 11111111 11111110 10101101	
取反	绝对值	加负号
00000000 00000000 00000001 01010010	338	-338



§ . 基础知识题

2、二进制补码转十进制整数（只考虑有符号数，写出具体步骤，包括-1、取反、绝对值、加负号）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001” 或 “11010100-00110001”）

D. 1101 1101 0110 0000 0110 1011 1001 0000

原数	-1	
11011101 01100000 01101011 10010000	11011101 01100000 01101011 10010000	
	-)	
	1	
	11011101 01100000 01101011 10001111	
取反	绝对值	加负号
00100010 10011111 10010100 01110000	580883568	-580883568



§ . 基础知识题

2、二进制补码转十进制整数（只考虑有符号数，写出具体步骤，包括-1、取反、绝对值、加负号）

格式要求：多字节时，每8bit中间加一个空格或-（例：“11010100 00110001” 或 “11010100-00110001”）

E. 学号对应的int型十进制负数的二进制补码形式（1. E的结果直接拿来当做本题初始数据即可）

原数	-1	
11111111 10000100 00110110 10000000	11111111 10000100 00110110 10000000	
	-)	
	1	
	11111111 10000100 00110110 01111111	
取反	绝对值	加负号
00000000 01111011 11001001 10000000	8112512	-8112512



§. 基础知识题

3、求出下列字符串的长度（要求仿照图例标出具体的字符分解）

"\r\n\t\\A\\t\x1b\"1234\xft\x2f\33" = 15

A. "\b\\nr\v384\x3fr2a\"r\\a\v\f"

"\b\\nr\v384\x3fr2a\"r\\a\v\f" = 18



§. 基础知识题

3、求出下列字符串的长度（要求仿照图例标出具体的字符分解）

"\r\n\t\\A\\t\x1b\"1234\xft\x2f\33" = 15

B. "\138\xa2\214\x6w\383\x65\042\xd5\257\x3e\1325\x6a\175\x2e"

"\138\xa2\214\x6w\383\x65\042\xd5\257\x3e\1325\x6a\175\x2e" = 19



§. 基础知识题

3、求出下列字符串的长度（要求仿照图例标出具体的字符分解）

"\r\n\t\\A\\t\x1b\""\1234\xft\x2f\33" = 15

这两个，第一个是正确的，第二个有问题，请构造测试程序验证，并将构造的测试程序及相应的error或warning信息的截图贴在文档中；对有warning的测试程序的运行结果给出长度分析，对有error的测试程序给出你的理解（主要是对比从八进制和十六进制转义的差异）

C. "\t\\r\nv\293\23456f\"r\\av\f"
"\t\\r\nv\293\x23456f\"r\\av\f"

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << strlen("\t\\r\nv\293\23456f\"r\\av\f") << endl;
6     return 0;
7 }
```



八进制严格按照读取三位（1-3 位）的形式，

但是十六进制的读取就不限制位数了，

我们将\x23456f 转换为十进制就是 2311535

说明十六进制在读取的过程中将后面的六位

全部读取了

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << strlen("\t\\r\nv\293\x23456f\"r\\av\f") << endl;
6     return 0;
7 }
```




§. 基础知识题

3、求出下列字符串的长度（要求仿照图例标出具体的字符分解）

"\r\n\t\\A\\t\x1b\""\1234\xft\x2f\33" = 15

这两个，都是不完全正确的，请构造测试程序验证，并将你构造的测试程序及相应的error或warning信息的截图贴在文档中；对有warning的测试程序的运行结果给出长度分析，对有error的测试程序给出你的理解

D. "\9876"
"*321"

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout << strlen("\9876") << endl;
6      return 0;
7  }
```

整个解决方案 错误 0 警告 1 消息 0

代码	说明
C4129	"9": 不可识别的字符转义序列

4 个长度\后面的 9 超出八进制范围，直接算新的字符，
后面的 9876 都算新字符，所以 4 个长度

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      cout << strlen("\*321") << endl;
6      return 0;
7  }
```

整个解决方案 错误 0 警告 1 消息 0

代码	说明
C4129	"*": 不可识别的字符转义序列

4 个长度，\后面的*超出八进制范围，直接算新的字符，
后面的 321 都算新字符，所以 4 个长度



§. 基础知识题

3、求出下列字符串的长度（要求仿照图例标出具体的字符分解）

"\r\n\t\\A\\t\x1b\""\1234\xft\x2f\33" = 15

这两个，都是不完全正确的，请构造测试程序验证，并将你构造的测试程序及相应的error或warning信息的截图贴在文档中；对有warning的测试程序的运行结果给出长度分析，对有error的测试程序给出你的理解

E. "\xg321"

"\x*321"

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << strlen("\xg321") << endl;
6     return 0;
7 }
```



两个十六进制数在读取过程中由于遇到超出范围的字符，直接算新的字符，导致\x后面没有一位数字，出现了无效的十六进制数，导致错误

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << strlen("\x*321") << endl;
6     return 0;
7 }
```

§. 基础知识题



4、给出下列程序段中变量b的值（要综合参考课件P. 45-51 和 P. 86-89，给出包含整型提升+丢弃的过程，具体见下）

例: short a=1;
short b=a-2;

Step1: b=a-2, 得b二进制补码形式

a = 00000000 00000000 00000000 00000001 → a （红色表示整型提升的填充位）
-) 2 = 00000000 00000000 00000000 00000010 → 2

11111111 11111111 11111111 11111111 → a-2(int型)
b = ~~11111111 11111111~~ 11111111 11111111 → b=a-2(二进制补码形式, 删除线表示丢弃的位数)

Step2: 求b的十进制表示

(1) 减一 11111111 11111111
 -) 00000000 00000001

11111111 11111110

(2) 取反 00000000 00000001

(3) 绝对值 1

(4) 加负号 -1 (b的十进制表示形式)

本页不用作答



§. 基础知识题

4、给出下列程序段中变量b的值（要综合参考课件P. 45-51 和 P. 86-89，给出包含整型提升+丢弃的过程）

A. short a=32760;
short b=a+14;

Step1: b=a+14, 得 b 二进制补码形式

a = 00000000 00000000 01111111 11111000 -> a (整型提升)
+) 14 = 00000000 00000000 00000000 00001110 -> 14

00000000 00000000 10000000 00000110 -> a+14(int 型)
b = ~~00000000 00000000~~ 10000000 00000110 -> b=a+14
(二进制补码形式, 删除线表示丢弃的位数)

Step2: 求 b 的十进制表示

(1) 减一 10000000 00000110
 -) 00000000 00000001

10000000 00000101

(2) 取反 01111111 11111010

(3) 绝对值 32762

(4) 加负号 -32762 (b 的十进制表示形式)



§. 基础知识题

4、给出下列程序段中变量b的值（要综合参考课件P. 45-51 和 P. 86-89，给出包含整型提升+丢弃的过程）

```
B. unsigned short a=65530;  
   short b=a;
```

Step1: b=a, 得 b 二进制补码形式

a = 00000000 00000000 11111111 11111010 -> a (红色表示整型提升填充)

b = ~~00000000 00000000~~ 11111111 11111010 -> b=a

(二进制补码形式, 删除线表示丢弃的位数)

Step2: 求 b 的十进制表示

(1) 减一 11111111 11111010
 -) 00000000 00000001

 11111111 11111001

(2) 取反 00000000 00000110

(3) 绝对值 6

(4) 加负号 -6 (b 的十进制表示形式)



§. 基础知识题

4、给出下列程序段中变量b的值（要综合参考课件P. 45-51 和 P. 86-89，给出包含整型提升+丢弃的过程）

```
C. short a=-8191;  
   int b=a;
```

Step1: 求 a 的补码形式

```
(1) 绝对值原码 00011111 11111111  
(2) 取反      11100000 00000000  
(3) +1        11100000 00000000  
              +)                1
```

11100000 00000001

Step2: 求 b 的十进制表示

(1) int b=a 11111111 11111111 11100000 00000001 -> a (整型提升)

11111111 11111111 11100000 00000001 -> b=a

```
(2) 减一      11111111 11111111 11100000 00000001  
              -) 00000000 00000000 00000000 00000001
```

11111111 11111111 11100000 00000000

(2) 取反 00000000 00000000 00011111 11111111

(3) 绝对值 8191

(4) 加负号 -8191 (b 的十进制表示形式)



§. 基础知识题

4、给出下列程序段中变量b的值（要综合参考课件P. 45-51 和 P. 86-89，给出包含整型提升+丢6弃的过程）

```
D. unsigned short a=65530;  
   long long int b=a;
```

Step1: long long int b=a, 得 b 二进制补码形式

```
a = 00000000 00000000 00000000 00000000 00000000 00000000  
    11111111 11111010  -> a （红色表示整型提升的填充位）  
b = 00000000 00000000 00000000 00000000 00000000 00000000  
    11111111 11111010  -> long long int b=a
```

Step2: 求 b 的十进制表示

(1) 通过计算权位得到 b 的十进制表示形式

$$\begin{aligned} b = & 1 \times 2^{15} + 1 \times 2^{14} + 1 \times 2^{13} + 1 \times 2^{12} + 1 \times 2^{11} + 1 \times 2^{10} + 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 \\ & + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 65530 \end{aligned}$$

b 的十进制表示是 65530



§. 基础知识题

4、给出下列程序段中变量b的值（要综合参考课件P. 45-51 和 P. 86-89，给出包含整型提升+丢弃的过程）

```
E. long long int a=4207654321;  
   int b=a;
```

Step1: long long int a=4207654321

```
   a = 00000000 00000000 00000000 00000000 11111010  
        11001011 10110101 10110001
```

由于 int b=a

```
   b=00000000 00000000 00000000 00000000 11111010  
        11001011 10110101 10110001      (高位丢弃)
```

Step2: 求 b 的十进制表示

```
(1) 减一    11111010 11001011 10110101 10110001  
   -)      00000000 00000000 00000000 00000001  
-----
```

```
          11111010 11001011 10110101 10110000
```

(2) 取反 00000101 00110100 01001010 01001111

(3) 绝对值 87312975

(4) 加负号 -87312975 (b 的十进制表示形式)



§. 基础知识题

4、给出下列程序段中变量b的值（要综合参考课件P. 45-51 和 P. 86-89，给出包含整型提升+丢弃的过程）

F. long a=-4207654321; //提示：本题先确定 -4207654321 什么类型，a是多少，才能进行b=a的计算
unsigned short b=a;

Step1: long int a=-4207654321，得 a 二进制补码形式

(1) 由于 -4207654321 应该是一个 signed long long int 的数据类型
所以它在计算机中储存的二进制补码应该是

1. 绝对值原码 = 00000000 00000000 00000000 00000000 11111010
11001011 10110101 10110001

2. 取反= 11111111 11111111 11111111 11111111 00000101
00110100 01001010 01001110

3. +1 11111111 11111111 11111111 11111111 00000101
00110100 01001010 01001110
+)

1

4. 11111111 11111111 11111111 11111111 00000101
00110100 01001010 01001111

(2) 将 -4207654321 通过 long int a=-4207654321 知

a = ~~11111111 11111111 11111111 11111111~~ 00000101
00110100 01001010 01001111 (高位丢弃)
a = 00000101 00110100 01001010 01001111

(3) 由于 unsigned short b=a 知

a = 00000101 00110100 01001010 01001111

b = ~~00000101 00110100~~ 01001010 01001111 (高位丢弃)

b = 01001010 01001111

Step2: 求 b 的十进制表示

求绝对值

$b = 1 \times 2^{14} + 1 \times 2^{11} + 1 \times 2^9 + 1 \times 2^6 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19023$

b 的十进制表示是 19023

§. 基础知识题



5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

A. $21 / 2 + 47 \% 3 - 1.3 + 3.5 * 2$

表达式一共有6个运算符，因此计算的6个步骤分别是：

步骤①： $21 / 2$

步骤②： $47 \% 3$

步骤③： ① + ②

步骤④： ③ - 1.3

步骤⑤： $3.5 * 2$

步骤⑥： ④ + ⑤



§. 基础知识题

5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

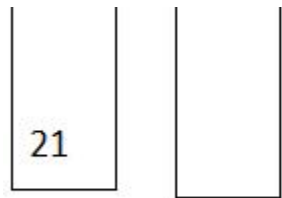
A. $21 / 2 + 47 \% 3 - 1.3 + 3.5 * 2$



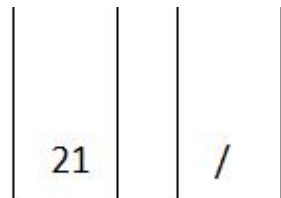
目前准备进栈的运算符如箭头所示，画出从初始分析到该运算符直到该运算符进栈的整个过程

例：以课件的第2个+为例，就是P. 69-P. 71三张图；以-为例，就是P. 75-P. 79五张图

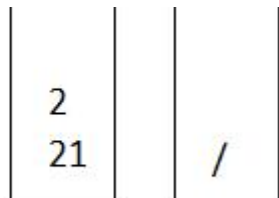
注：尽量画在本页上，从左到右依次排开即可，如果实在空间不够，可以加页



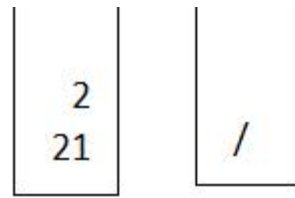
21 进栈



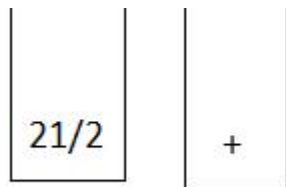
/ 进栈



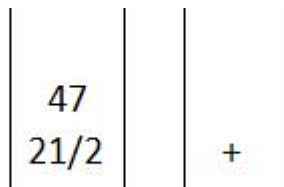
2 进栈



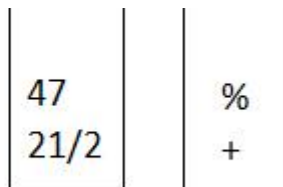
要进栈的 (+) 小于 (/), 且左结合, 求值



+ 进栈



47 进栈



% 进栈



§. 基础知识题

5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

A. $21 / 2 + 47 \% 3 - 1.3 + 3.5 * 2$



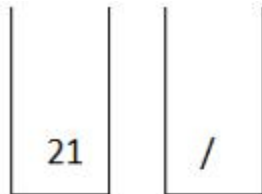
目前准备进栈的运算符如箭头所示，画出从初始分析到该运算符直到该运算符进栈的整个过程

例：以课件的第2个+为例，就是P. 69-P. 71三张图；以-为例，就是P. 75-P. 79五张图

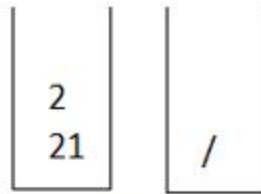
注：尽量画在本页上，从左到右依次排开即可，如果实在空间不够，可以加页



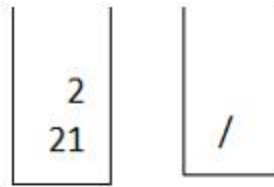
21 进栈



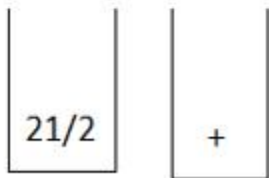
/ 进栈



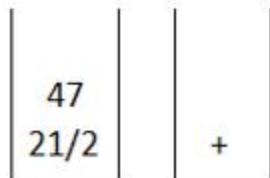
2 进栈



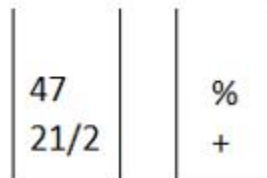
要进栈的 (+) 小于 (/), 且左结合, 求值



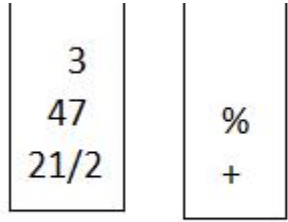
+ 进栈



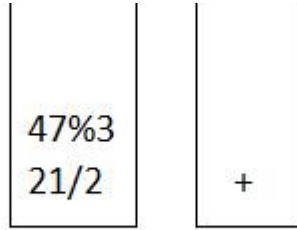
47 进栈



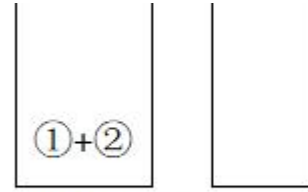
% 进栈



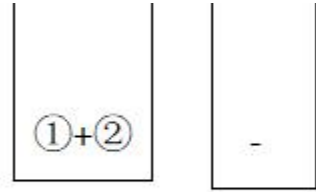
3 进栈



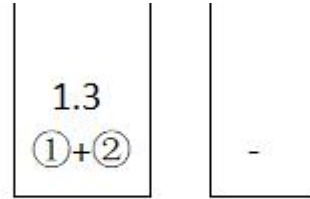
要进栈的 (-) 小于 (%), 且左结合, 求值



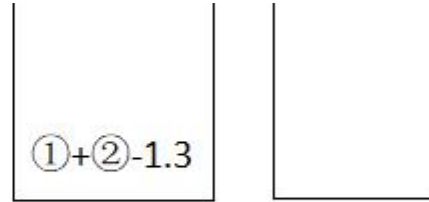
要进栈的 (-) 等于 (+), 且左结合, 求值
①=21/2 ②=47%3



- 进栈



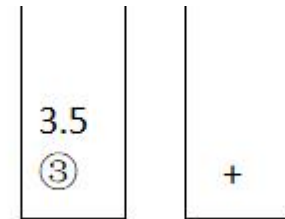
1.3 进栈



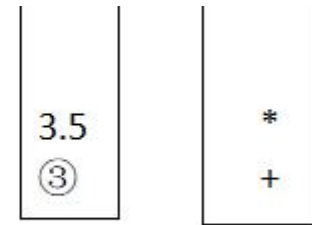
要进栈的 (+) 等于 (-), 且左结合, 求值
③=①+②-1.3



+进栈



3.5 进栈



*进栈

§. 基础知识题



5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

B. $a = 3 * 5$, $a = b = 6 * 4$ （假设所有变量均为int型）

表达式一共有6个运算符，因此计算的6个步骤分别是：

步骤①： $3 * 5 = 15$

步骤②： $a = 15$

步骤③： $6 * 4 = 24$

步骤④： $b = 24$

步骤⑤： $a = 24$

步骤⑥： $15 , 24 = 24$



§. 基础知识题

5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

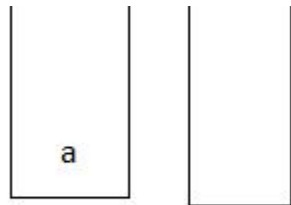
B. $a = 3 * 5$, $a = b = 6 * 4$ (假设所有变量均为int型)



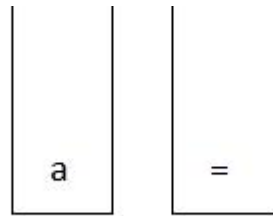
目前准备进栈的运算符如箭头所示，画出从初始分析到该运算符直到该运算符进栈的整个过程

例：以课件的第2个+为例，就是P. 69-P. 71三张图；以-为例，就是P. 75-P. 79五张图

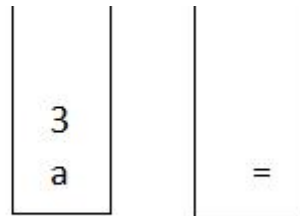
注：尽量画在本页上，从左到右依次排开即可，如果实在空间不够，可以加页



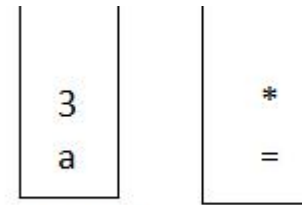
a 进栈



= 进栈



3 进栈



* 进栈



§. 基础知识题

5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

B. $a = 3 * 5$, $a = b = 6 * 4$ (假设所有变量均为int型)



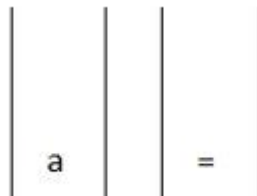
目前准备进栈的运算符如箭头所示，画出从初始分析到该运算符直到该运算符进栈的整个过程

例：以课件的第2个+为例，就是P. 69-P. 71三张图；以-为例，就是P. 75-P. 79五张图

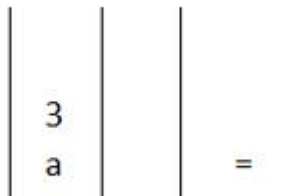
注：尽量画在本页上，从左到右依次排开即可，如果实在空间不够，可以加页



a 进栈



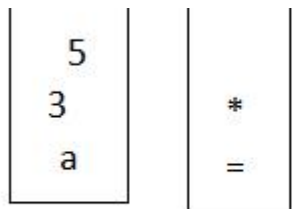
= 进栈



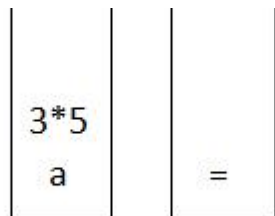
3 进栈



* 进栈

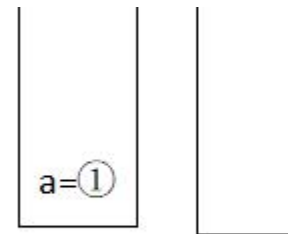


5 进栈

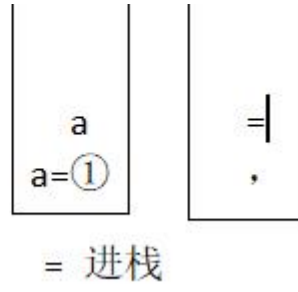
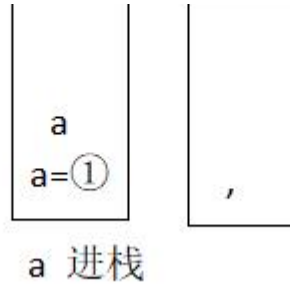
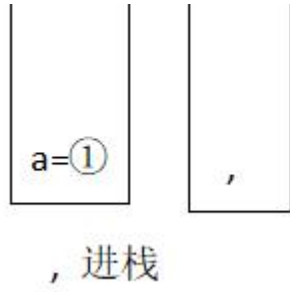


要进栈的 (,) 小于 (*), 且左结合, 求值

①=3*5



要进栈的 (,) 小于 (=), 求值



§. 基础知识题



5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

C. $a + (b - 3 * (a + c) - 2) \% 3$ （假设所有变量均为int型）

（本题提示：将左右小括号分开处理，

1、“(”**进栈前**优先级最高，**进栈后**优先级最低；

2、“)”优先级最低，因此要将栈中压在“(”之上的全部运算符都计算完成，随后和“(”成对消除即可

表达式一共有__8__个运算符，因此计算的__6__个步骤分别是：

步骤①： $a + c$

步骤②： $3 * \text{①}$

步骤③： $b - \text{②}$

步骤④： $\text{③} - 2$

步骤⑤： $\text{④} \% 3$

步骤⑥： $a + \text{⑤}$



§. 基础知识题

5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

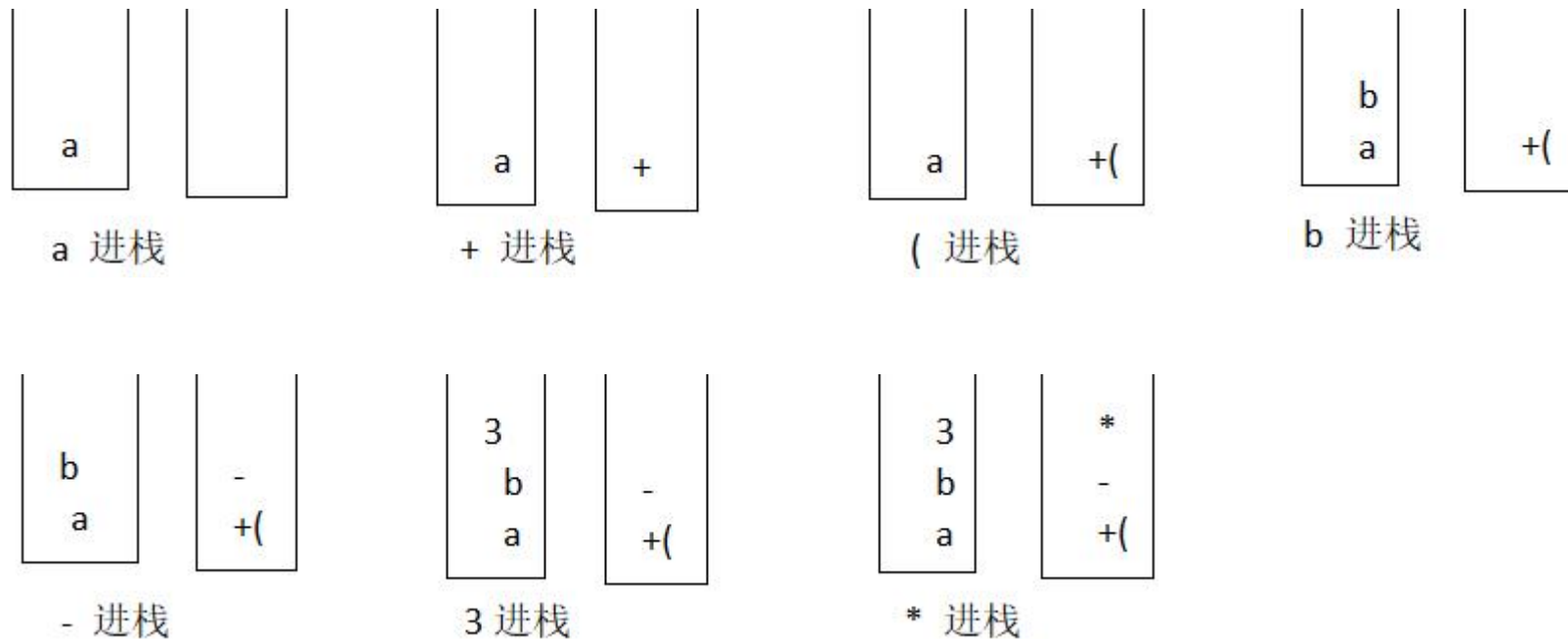
C. $a + (b - 3 * (a + c) - 2) \% 3$ (假设所有变量均为int型)

(本题提示：将左右小括号分开处理，

1、“(”进栈前优先级最高，进栈后优先级最低；

2、“)”优先级最低，因此要将栈中压在“(”之上的全部运算符都计算完成，随后和“(”成对消除即可

目前准备进栈的运算符如箭头所示，画出从初始分析到该运算符直到该运算符进栈的整个过程





§. 基础知识题

5、仿照课件PDF的P. 65-85，用栈方式给出下列表达式的求解过程

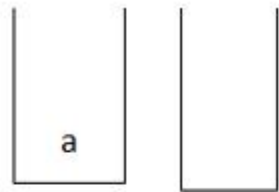
C. $a + (b - 3 * (a + c) - 2) \% 3$ (假设所有变量均为int型)

(本题提示：将左右小括号分开处理，

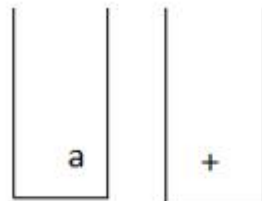
1、“(”**进栈前**优先级最高，**进栈后**优先级最低；

2、“)”优先级最低，因此要将栈中压在“(”之上的全部运算符都计算完成，随后和“(”成对消除即可

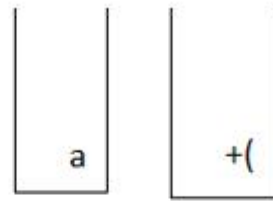
目前准备进栈的运算符如箭头所示，画出从初始分析到该运算符直到该运算符进栈的整个过程



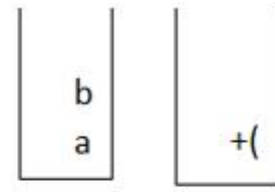
a 进栈



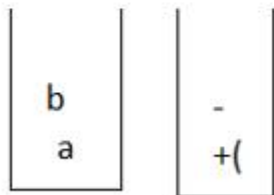
+ 进栈



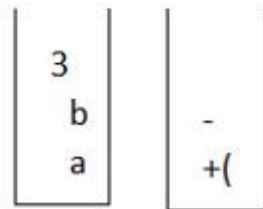
(进栈



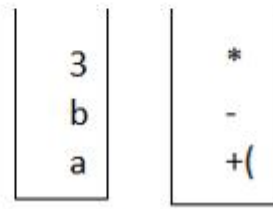
b 进栈



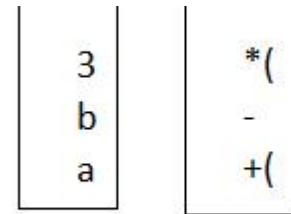
- 进栈



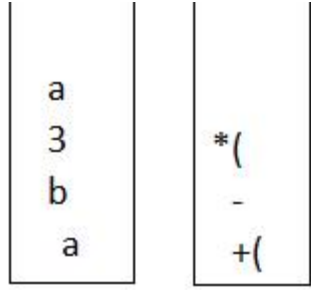
3 进栈



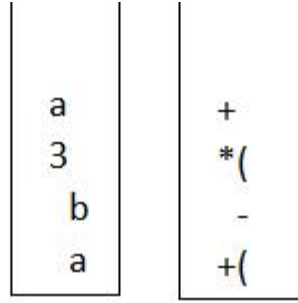
* 进栈



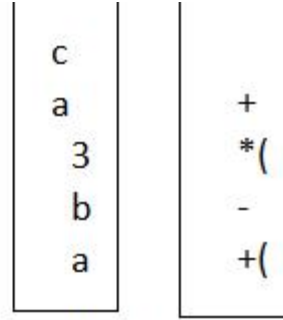
(进栈



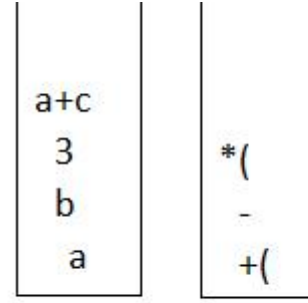
a 进栈



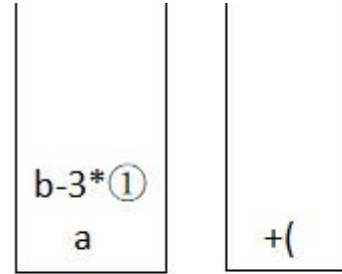
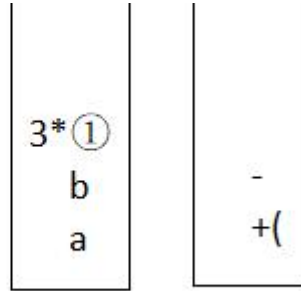
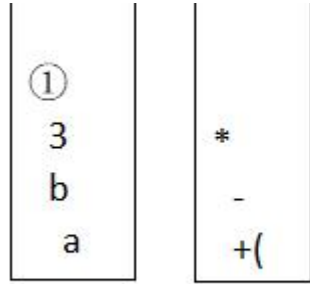
+ 进栈



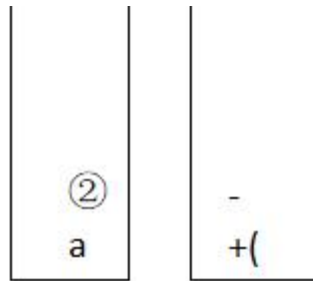
c 进栈



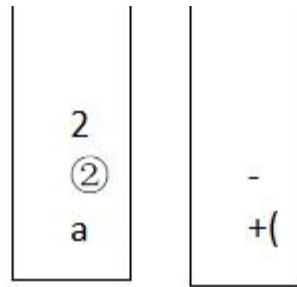
要进栈的()的优先级比(+)小, 求值



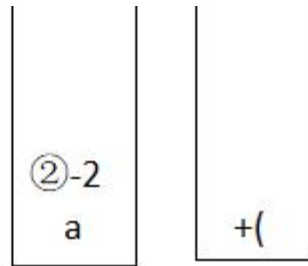
①=a+c)进栈与(消掉 要进栈的 (-) 小于 (*), 求值 要进栈的 (-) 等于 (-), 且左结合, 求值



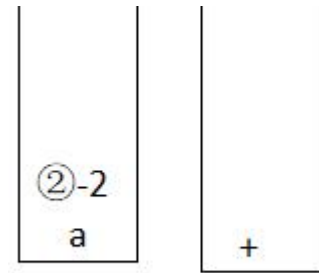
- 进栈 ②=b-3*①



2 进栈



要进栈的()小于(-), 求值



)进栈 与(消掉



§. 基础知识题

6、求表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图，示例见下）

例： $2LL - 32L * \text{int}(11.7) + 2.3f$

- | | | | |
|---|----|--------|------------|
| (1) $\text{int}(11.7)$ | => | 11 | int型 |
| (2) $32L * \text{int}(11.7)$ | => | 352 | long型 |
| (3) $2LL - 32L * \text{int}(11.7)$ | => | -350 | long long型 |
| (4) $2LL - 32L * \text{int}(11.7) + 2.3f$ | => | -347.7 | float型 |

```
demo.cpp 1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << 2LL - 32L * int(11.7) + 2.3f << endl;
6     cout << typeid(2LL - 32L * int(11.7) + 2.3f).name() << endl;
7     return 0;
8 }
9
```

Microsoft Visual Studio 调试控制台

```
-347.7
float
```

本页不用作答



§. 基础知识题

6、求表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图）

A. $a = 2 * 5$, $a = b = 4 * 4$

（写验证程序时，假设所有变量均为int型）

(1)	$2 * 5$	=>	10	int 型
(2)	$a = 10$	=>	10	int 型
(3)	$4 * 4$	=>	16	int 型
(4)	$b = 16$	=>	16	int 型
(5)	$a = 16$	=>	16	int 型
(6)	10, 16	=>	16	int 型

```
Project1 (全局范围)
1 #include <iostream>
2 #include <climits>
3 using namespace std;
4 int main()
5 {
6     int a, b;
7     cout << (a = 2 * 5, a = b = 4 * 4) << endl;
8     cout << typeid((a = 2 * 5, a = b = 4 * 4)).name() << endl;
9     return 0;
10 }
```

Microsoft Visual Studio 调试控制台

```
16
int
```



§. 基础知识题

6、求表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图）

B. $a + (b - 2 * (a + c) - 3) \% 4$ （写验证程序时，假设所有变量均为int型，**abc**的值自定义即可）

$a = 3$, $b = 2$, $c = 1$;

(1)	$a + c$	=>	4	int 型
(2)	$2 * (a + c)$	=>	8	int 型
(3)	$b - 2 * (a + c)$	=>	-6	int 型
(4)	$b - 2 * (a + c) - 3$	=>	-9	int 型
(5)	$(b - 2 * (a + c) - 3) \% 4$	=>	-1	int 型
(6)	$a + (b - 2 * (a + c) - 3) \% 4$	=>	2	int 型

```
1 #include <iostream>
2 #include <climits>
3 using namespace std;
4 int main()
5 {
6     int a=3, b=2, c=1;
7     cout << a + (b - 2 * (a + c) - 3) % 4 << endl;
8     cout << typeid(a + (b - 2 * (a + c) - 3) % 4).name() << endl;
9     return 0;
10 }
```

Microsoft Visual Studio

```
2
int
```




§. 基础知识题

6、求表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图）

C. $2.5F * 2LU + 3U * 5LL - 'b'$

(1)	$2.5F * 2LU$	=>	5	float 型
(2)	$3U * 5LL$	=>	15	_int 64 型
(3)	$2.5F * 2LU + 3U * 5LL$	=>	20	float 型
(4)	$2.5F * 2LU + 3U * 5LL - 'b'$	=>	-78	float 型

```
project1 (主函数范围)
1  #include <iostream>
2  #include <climits>
3  using namespace std;
4  int main()
5  {
6      cout << 2.5F * 2LU + 3U * 5LL - 'b' << endl;
7      cout << typeid(2.5F * 2LU + 3U * 5LL - 'b').name() << endl;
8      return 0;
9  }
```

```
-78
float
```



§. 基础知识题

6、求表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图）

D. $7LU \% 3 + 13LL \% 7 + 2.3$

- | | | | | |
|-----|------------------------------|----|-----|-----------------|
| (1) | $7LU \% 3$ | => | 1 | unsigned long 型 |
| (2) | $13LL \% 7$ | => | 6 | _int 64 型 |
| (3) | $7LU \% 3 + 13LL \% 7$ | => | 7 | _int 64 型 |
| (4) | $7LU \% 3 + 13LL \% 7 + 2.3$ | => | 9.3 | double 型 |

```
1  #include <iostream>
2  #include <climits>
3  using namespace std;
4  int main()
5  {
6      cout << 7LU % 3 + 13LL % 7 + 2.3 << endl;
7      cout << typeid(7LU % 3 + 13LL % 7 + 2.3).name() << endl;
8      return 0;
9  }
```

选择 Microsoft Visual Studio 调试控制台

9.3
double



§. 基础知识题

6、求表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图）

E. $3.2 + 11 \% 3 * \text{static_cast}\langle\text{unsigned int}\rangle(1.8F + 2) \% 3 * 4.2F$

- (1) $11 \% 3 \Rightarrow 2$ int 型
- (2) $1.8F + 2 \Rightarrow 3.8$ float 型
- (3) $7LU \% 3 + 13LL \% 7 \Rightarrow 3$ unsigned int 型
- (4) $11 \% 3 * \text{static_cast}\langle\text{unsigned int}\rangle(1.8F + 2) \Rightarrow 6$ unsigned int 型
- (5) $11 \% 3 * \text{static_cast}\langle\text{unsigned int}\rangle(1.8F + 2) \% 3 \Rightarrow 0$ unsigned int 型
- (6) $11 \% 3 * \text{static_cast}\langle\text{unsigned int}\rangle(1.8F + 2) \% 3 * 4.2F \Rightarrow 0$ float 型
- (7) $3.2 + 11 \% 3 * \text{static_cast}\langle\text{unsigned int}\rangle(1.8F + 2) \% 3 * 4.2F \Rightarrow 3.2$ double 型

```
Project1 (全局范围)
1  #include <iostream>
2  #include <climits>
3  using namespace std;
4  int main()
5  {
6      cout << 3.2 + 11 % 3 * static_cast<unsigned int>(1.8F + 2) % 3 * 4.2F
7          << endl;
8      cout << typeid(3.2 + 11 % 3 * static_cast<unsigned int>(1.8F + 2) % 3 * 4.2F
9          ).name() << endl;
10     return 0;
11 }
```

Microsoft Visual Studio 调试控制台

3.2
double



§. 基础知识题

6、求表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图）

F. $\text{long}(3.78 + 1.33) \% 2 + (\text{int})1.5 \% 7\text{U} - 'Y' * 6\text{L}$

- | | | | | |
|-----|--|----|------------|-----------------|
| (1) | $3.78 + 1.33$ | => | 5.11 | double 型 |
| (2) | $\text{long}(3.78 + 1.33)$ | => | 5 | long 型 |
| (3) | $\text{long}(3.78 + 1.33) \% 2$ | => | 1 | long 型 |
| (4) | $(\text{int})1.5$ | => | 1 | int 型 |
| (5) | $(\text{int})1.5 \% 7\text{U}$ | => | 1 | unsigned int 型 |
| (6) | $\text{long}(3.78 + 1.33) \% 2 + (\text{int})1.5 \% 7\text{U}$ | => | 2 | unsigned long 型 |
| (7) | $'Y' * 6\text{L}$ | => | 534 | long 型 |
| (8) | $\text{long}(3.78 + 1.33) \% 2 + (\text{int})1.5 \% 7\text{U} - 'Y' * 6\text{L}$ | => | 4294966764 | unsigned long 型 |

```
1 #include <iostream>
2 #include <climits>
3 using namespace std;
4 int main()
5 {
6     cout << long(3.78 + 1.33) % 2 + (int)1.5 % 7U - 'Y' * 6L
7         << endl;
8     cout << typeid(long(3.78 + 1.33) % 2 + (int)1.5 % 7U - 'Y' * 6L
9         ).name() << endl;
10    return 0;
11 }
```

Microsoft Visual Studio 调试控制台

```
4294966764
unsigned long
```

§. 基础知识题



7、求复合赋值表达式的值（要求给出计算过程、每步计算结果中变量的值、对应的验证程序及结果截图，示例见下）

假设 `int a=5, n = 12;`

例: `a += n`

=> `a = a + n`

(1) `a + n` `a=5` `n=12` 和17存放在中间变量中

(2) `a = 和` `a=17` `n=12`

```
demo.cpp  x
demo-CPP  (全局范围)

2  using namespace std;
3  int main()
4  {
5      int a = 5, n = 12;
6      a += n;
7      cout << a << ' ' << n << endl;
8      return 0;
9  }
```

Microsoft Visual Studio 调试控制台

17 12

本页不用作答

§. 基础知识题



7、求复合赋值表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图，具体见下）

假设 `int a=2, n = 3;`

A. `a -= a + n`

`a -= a + n`

$\Rightarrow a = a - (a + n)$

(1) `a = a - (a + n)` `a=2, n=3` 结果-3 存在中间变量里面

(2) `a = 结果` `a = -3, n=3`

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a = 2, n = 3;
6      a -= a + n;
7      cout << a << ' ' << n << endl;
8      return 0;
9  }
```

Microsoft Visual Studio 调试控制台

-3 3

§. 基础知识题



7、求复合赋值表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图，具体见下）

假设 `int a=2, n = 3;`

B. `a += n += 3`

`a += n += 3`

\Rightarrow `a+=n=n+3`

- (1) `a+=n=n+3` `a=2, n = 3` `n+3` 的和保存在中间变量里
- (2) `a+=n=和` `a=2, n=6`
- (3) `a=a+n` `a=2, n=6` `a+n` 的结果保存在中间变量里
- (4) `a=结果` `a=8, n=6`

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a = 2, n = 3;
6      a += n+=3;
7      cout << a << ' ' << n << endl;
8      return 0;
9  }
```

Microsoft Visual Studio 调试控制台

8 6



§. 基础知识题

7、求复合赋值表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图，具体见下）

假设 `int a=2, n = 3;`

C. `a *= a += a /= a`

`a *= a += a /= a`

`==> a *= a += a = a / a`

(1) `a *= a += a = a / a` `a=2, n=3` `a/a` 的商保存在中间变量里

(2) `a *= a += a = 商` `a=1, n=3`

(3) `a *= a = a + a` `a=1, n=3` `a+a` 的和保存在中间变量里

(4) `a *= a = 和` `a=2, n=3`

(5) `a = a * a` `a=2, n=3` `a * a` 的结果保存在中间变量里

(6) `a = 结果` `a=4, n=3`

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a = 2, n = 3;
6      a *= a += a /= a;
7      cout << a << ' ' << n << endl;
8      return 0;
9  }
```

Microsoft Visual Studio 调试控制台

4 3



§. 基础知识题

7、求复合赋值表达式的值（要求给出计算过程、每步计算结果及数据类型、对应的验证程序及结果截图，具体见下）

假设 `int a=2, n = 3;`

D. `a %= n %= 3` 本题需要解释，为什么编译不报错，但运行无输出、返回代码为负值、且运行时间比7. ABC长（无法理解或说清楚原因的，给出合理猜测也可）

编译没有语法错误，但是在运行过程中出现了逻辑错误，即对0进行求余。
运行时间长可能是求余运算中是通过减法方式，直至出现一个数比除数小，
但由于对0求余，无法得出结果，程序自动停止，返回负数说明出现错误。

The screenshot shows a Visual Studio IDE with a C++ file named `demo.cpp`. The code is as follows:

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a = 2, n = 3;
6      a %= n %= 3;
7      cout << a << ' ' << n << endl;
8      return 0;
9  }
```

The output window at the bottom shows the execution result: `D:\Workspace\VS2022-demo\Debug\demo-cpp.exe (进程 3004) 已退出, 代码为 -1073741676。`. A red arrow points to the error code `-1073741676`, which is circled in red.