

## §. 基础知识题



要求:

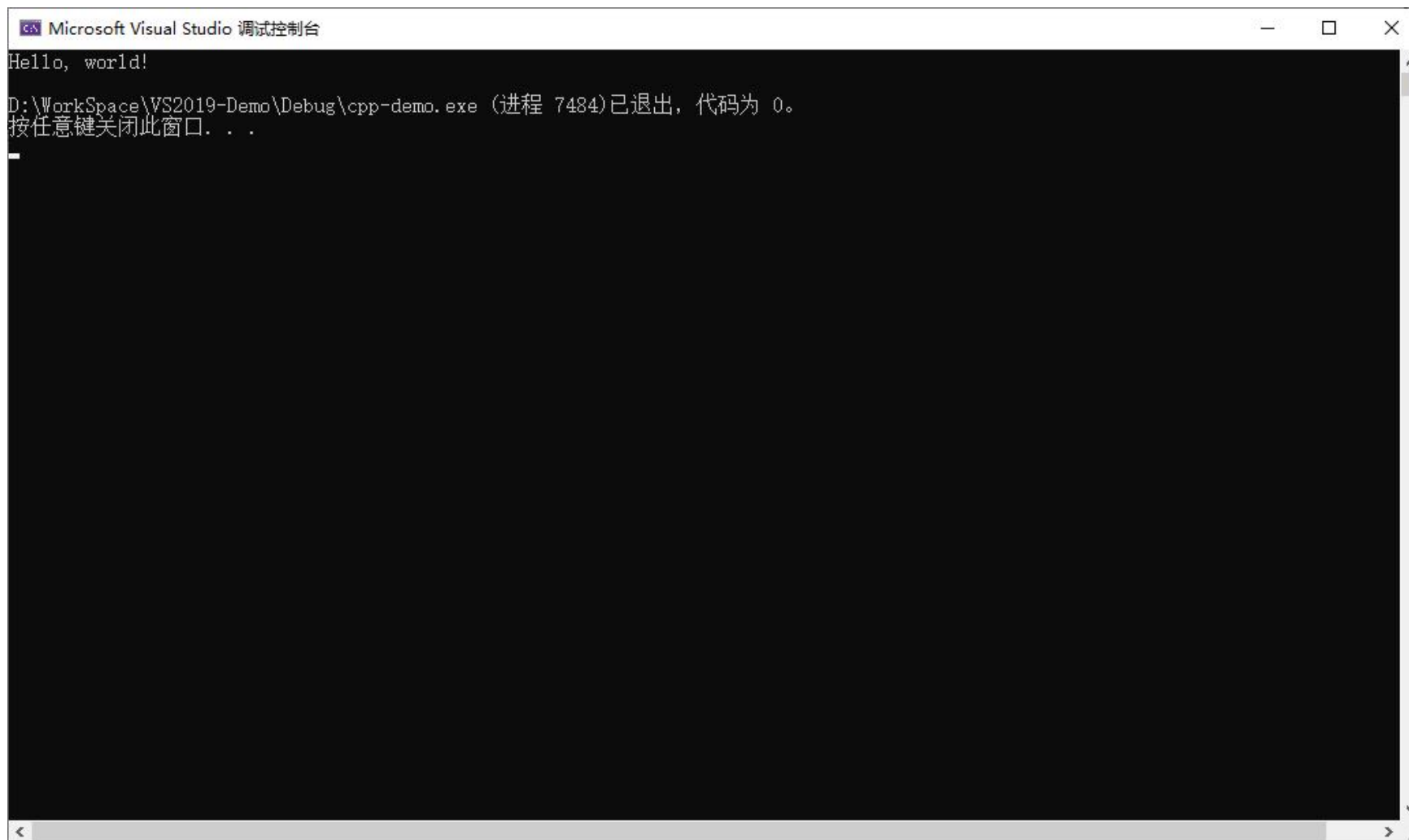
- 1、完成本文档中所有的题目并写出分析、运行结果
- 2、无特殊说明，均使用VS2022编译即可
- 3、直接在本文件上作答，**写出答案/截图（不允许手写、手写拍照截图）**即可；填写答案时，为适应所填内容或贴图，**允许调整**页面的字体大小、颜色、文本框的位置等
  - ★ 贴图要有效部分即可，不需要全部内容
  - ★ 在保证一页一题的前提下，具体页面布局可以自行发挥，简单易读即可
  - ★ **不允许**手写在纸上，再拍照贴图
  - ★ **允许**在各种软件工具上完成（不含手写），再截图贴图
- 4、转换为pdf后提交
- 5、**9月19日前**网上提交本次作业（在“文档作业”中提交）

## §. 基础知识题

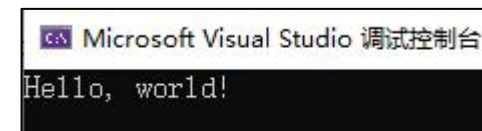


贴图要求：只需要截取输出窗口中的有效部分即可，如果全部截取/截取过大，则视为无效贴图

例：无效贴图



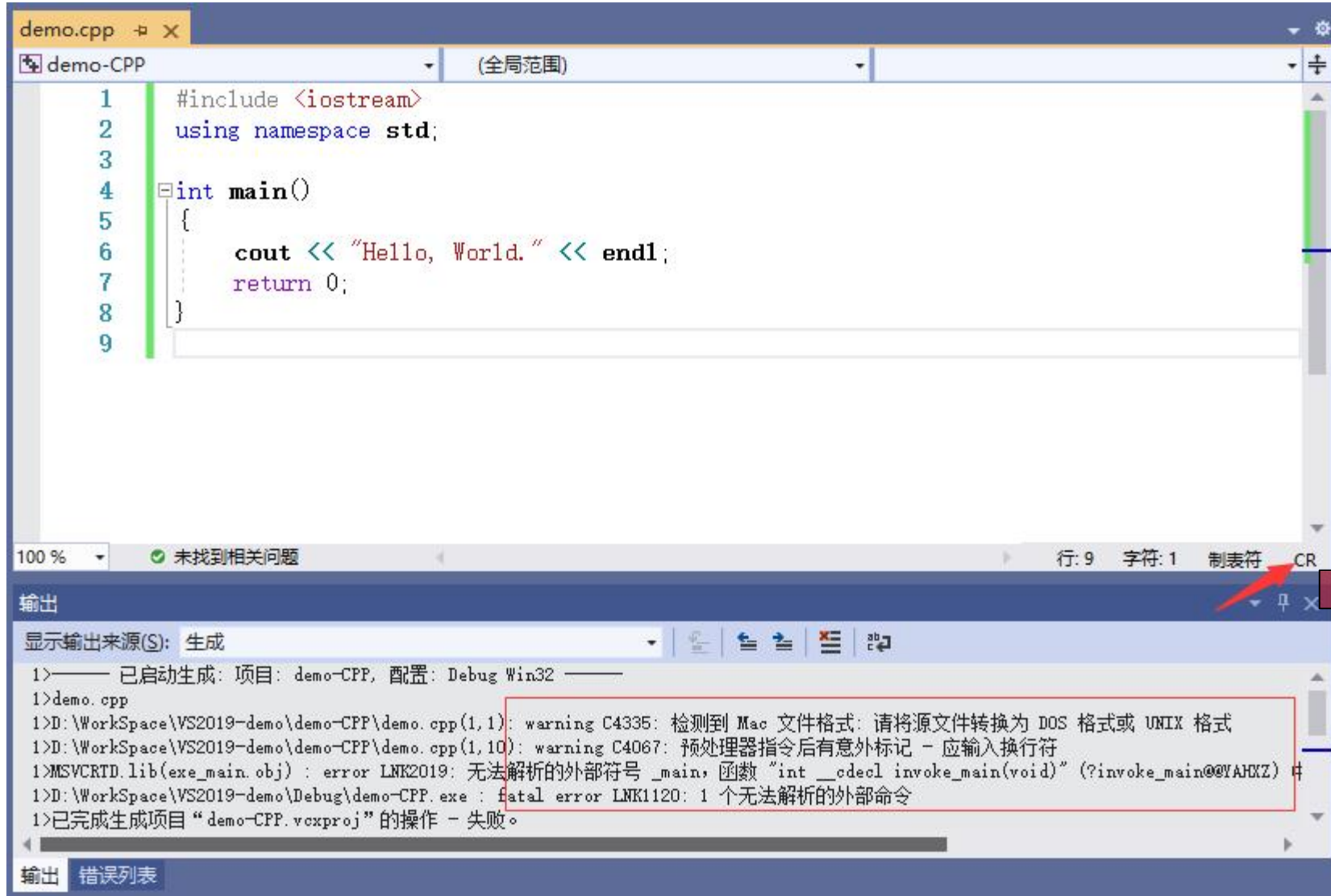
例：有效贴图





## §. 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、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 （十进制表示形式）

(4) 加负号   -1 （十进制表示形式）

本页不用作答



## §. 基础知识题

1、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=32740;  
   short b=a+34;
```

Step1:  $b=a+34$ ，得b二进制补码形式

$a =$  **00000000 00000000** 01111111 11100100  $\rightarrow a$ （红色表示整型提升的填充位）

$+ ) \quad 34 =$  00000000 00000000 00000000 00100010  $\rightarrow 34$

-----  
00000000 00000000 10000000 00000110  $\rightarrow a+34$  (int型)

$b =$  ~~00000000 00000000~~ 10000000 00000110  $\rightarrow b=a+34$  (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

-----  
1000000000000101

(2) 取反      0111111111111010

(3) 绝对值    32762    (十进制表示形式)

(4) 加负号    **-32762**    (十进制表示形式)



## §. 基础知识题

1、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=65420;  
   short b=a;
```

Step1: b=a, 得b二进制补码形式

a = 00000000 00000000 11111111 10001100 → a （红色表示整型提升的填充位）

b = ~~00000000 00000000~~ 11111111 10001100 → 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 10001100  
              -) 00000000 00000001

-----  
              11111111 10001011

(2) 取反      00000000 01110100

(3) 绝对值    116 （十进制表示形式）

(4) 加负号    -116 （十进制表示形式）



## §. 基础知识题

1、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=-2047;  
   int b=a;
```

Step1: b=a, 得b二进制补码形式

a= 1111 1000 0000 0001 (a在机内存储的二进制补码形式)

a = 11111111 11111111 11111000 00000001 → a (红色表示整型提升的填充位)

b = 11111111 11111111 11111000 00000001 → b=a(二进制补码形式, 无丢弃)

Step2: 求b的十进制表示

(1) 减一      11111111 11111111 11111000 00000001  
              -) 00000000 00000000 00000000 00000001

-----  
              11111111 11111111 11111000 00000000

(2) 取反      00000000 00000000 00000111 11111111

(3) 绝对值    2047 (十进制表示形式)

(4) 加负号    -2047 (十进制表示形式)



## §. 基础知识题

1、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）

```
D. unsigned short a=65420;  
   long long int b=a;
```

Step1: b=a, 得b二进制补码形式

a = 00000000 00000000 11111111 10001100 → a （红色表示整型提升的填充位）

a = 00000000 00000000 00000000 00000000 00000000 00000000 11111111 10001100  
（蓝色表示从int提升为long long int）

b = 00000000 00000000 00000000 00000000 00000000 00000000 11111111 10001100 →  
b=a(二进制补码形式，无丢弃)

Step2: 求b的十进制表示

b为有符号数，最高位为符号位，可知b为正数，所以补码即原码，按权相加得 b=65420





## §. 基础知识题

1、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=4201234567;  
   int b=a;
```

Step1: b=a, 得b二进制补码形式

a = 00000000 00000000 00000000 00000000 11111010 01101001 11000000 10000111

长赋短，低位赋值，高位直接丢弃，得

b = ~~00000000 00000000 00000000 00000000~~ 11111010 01101001 11000000 10000111

Step2: 求b的十进制表示

(1) 减一      11111010 01101001 11000000 10000111  
              -) 00000000 00000000 00000000 00000001

-----  
              11111010 01101001 11000000 10000110

(2) 取反      00000101 10010110 00111111 01111001

(3) 绝对值    93732729 （十进制表示形式）

(4) 加负号    **-93732729** （十进制表示形式）



## §. 基础知识题

1、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=-4201234567; //提示：本题先确定 -4201234567 什么类型，a是多少，才能进行b=a的计算  
unsigned short b=a;

Step1: b=a, 得b二进制码形式

a = 11111111 11111111 11111111 11111111 00000101 10010110 00111111 01111001

-> a (long型a的二进制补码形式)

b = ~~11111111 11111111 11111111 11111111~~ 00000101 10010110 00111111 01111001

-> b=a

长赋短，低位赋值，高位直接丢弃。（蓝色表示丢弃的位数）

Step2: 求b的十进制表示

**b为无符号数，故按权相加直接转十进制数字即可，具体过程略，得b= 16249**



## §. 基础知识题

2、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

例.  $1 + 2 + 3$

表达式一共有2个运算符，因此计算的2个步骤分别是（仿课件P. 85，本页不需要画栈，但要有栈思维，下同）：

步骤①:  $1 + 2 \Rightarrow \text{式1}$

步骤②:  $\text{式1} + 3$

本页不用作答



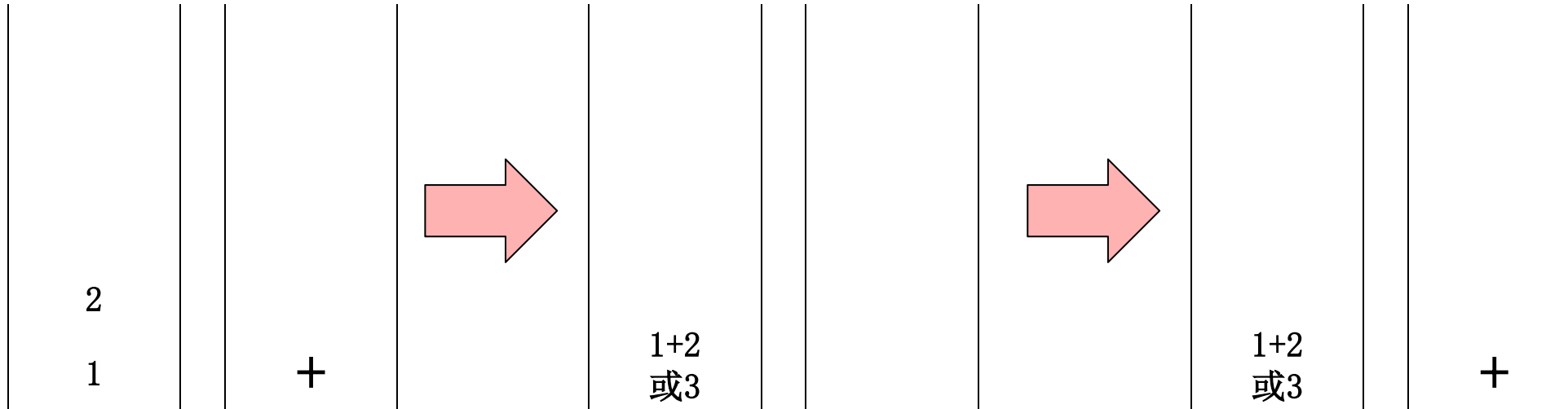
## §. 基础知识题

2、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

例.  $1 + 2 + 3$



目前准备进栈的运算符如箭头所示，画出从该运算符**准备进栈到进栈完成**的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，提示：本例是3组，等价于课件P. 69~71）



要进栈的+等于栈顶的+，  
左结合，先计算

要进栈的+等于栈顶的+，  
左结合，先计算

本页不用作答



## §. 基础知识题

2、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.  $11 / 2 + 37 \% 4 - 3.2 + 2.5 * 2$

表达式一共有6个运算符，因此计算的6个步骤分别是（仿课件P. 85，本页不需要画栈，但要有栈思维，下同）：

步骤①:  $11 / 2 \Rightarrow \text{式1}$

步骤②:  $37 \% 4 \Rightarrow \text{式2}$

步骤③:  $\text{式1} + \text{式2} \Rightarrow \text{式3}$

步骤④:  $\text{式3} - 3.2 \Rightarrow \text{式4}$

步骤⑤:  $2.5 * 2 \Rightarrow \text{式5}$

步骤⑥:  $\text{式4} + \text{式5} \Rightarrow \text{式6}$



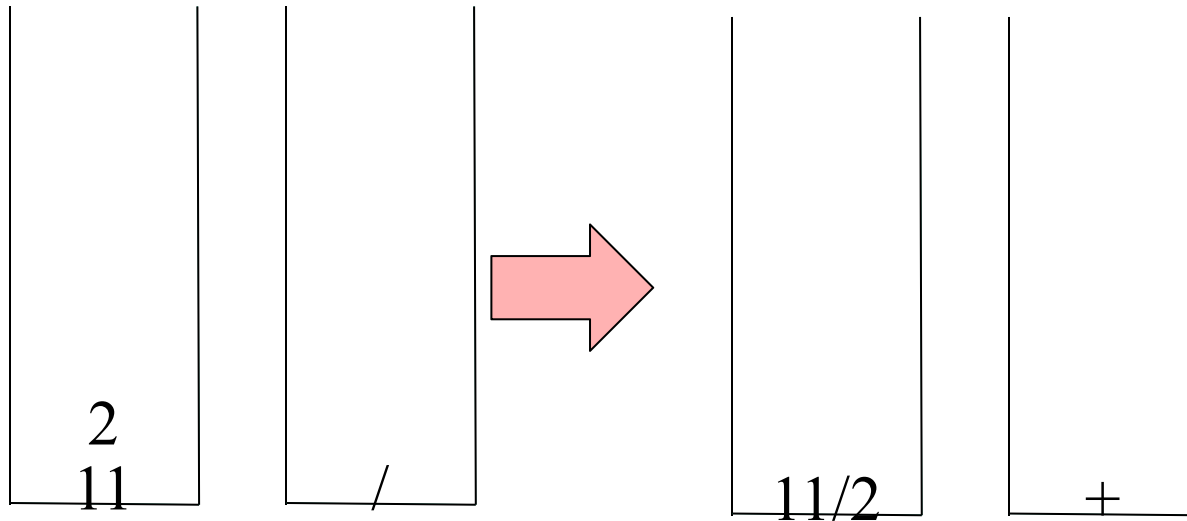
## §. 基础知识题

2、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.  $11 / 2 + 37 \% 4 - 3.2 + 2.5 * 2$



目前准备进栈的运算符如箭头所示，画出从该运算符**准备进栈到进栈完成**的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，每页三组，不够可加页）



要入栈的+优先级低于/,  
左结合，故先计算11/2



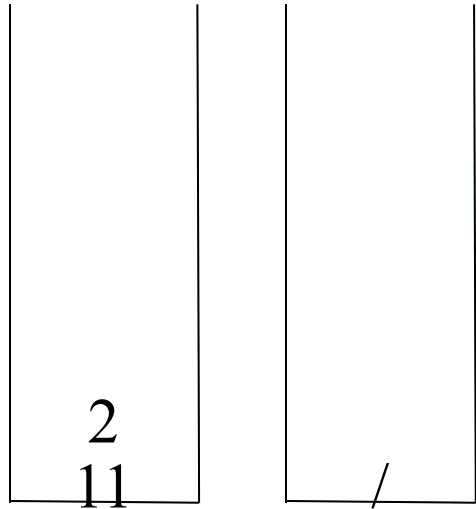
## §. 基础知识题

2、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

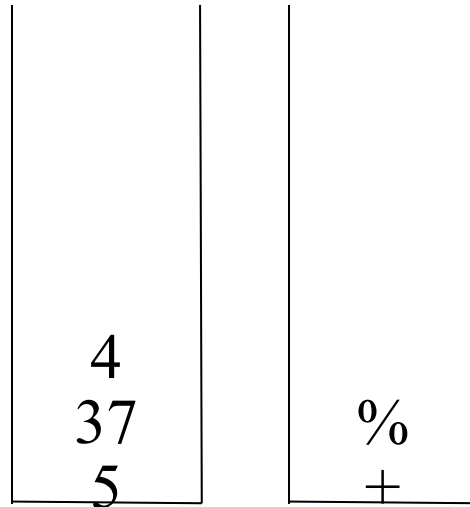
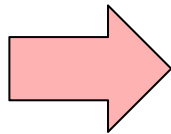
A.  $11 / 2 + 37 \% 4 - 3.2 + 2.5 * 2$



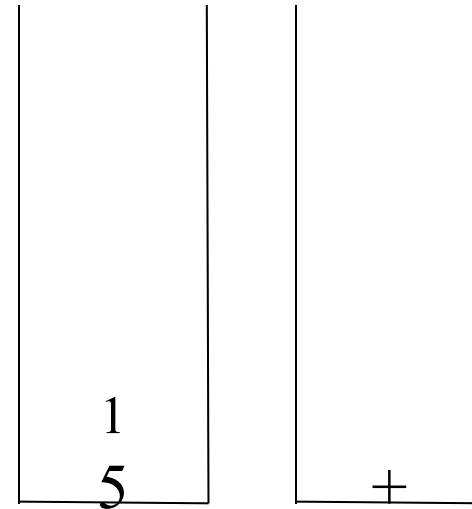
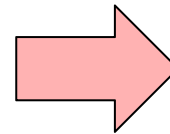
目前准备进栈的运算符如箭头所示，画出从该运算符**准备进栈到进栈完成**的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，每页三组，不够可加页）



要入栈的+优先级低于/,  
左结合，故先计算 $11/2$



要入栈的-优先级低于%,  
左结合，故先计算 $37\%4$



要入栈的-优先级等于+，  
左结合，故先计算  
 $11/2+37\%4$



式1+式2 (6)

-





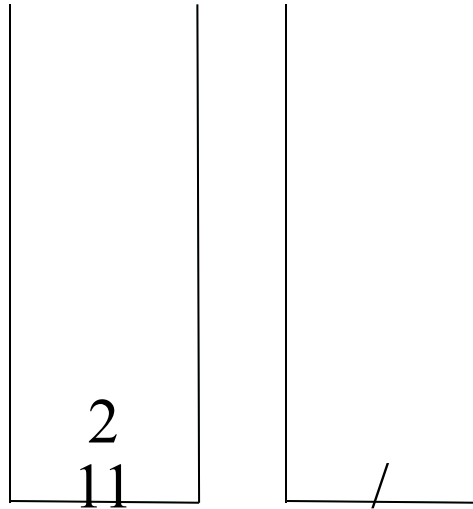
## §. 基础知识题

2、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

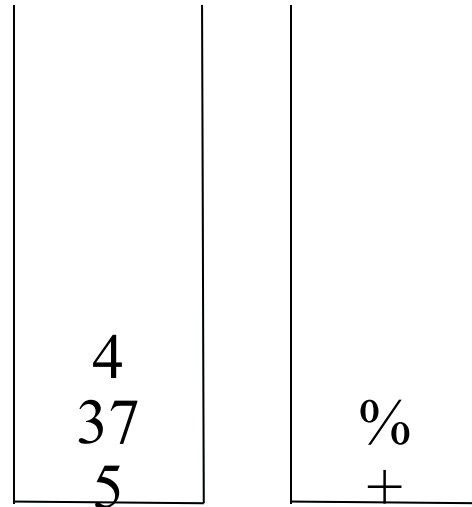
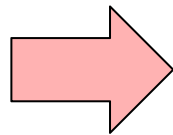
A.  $11 / 2 + 37 \% 4 - 3.2 + 2.5 * 2$



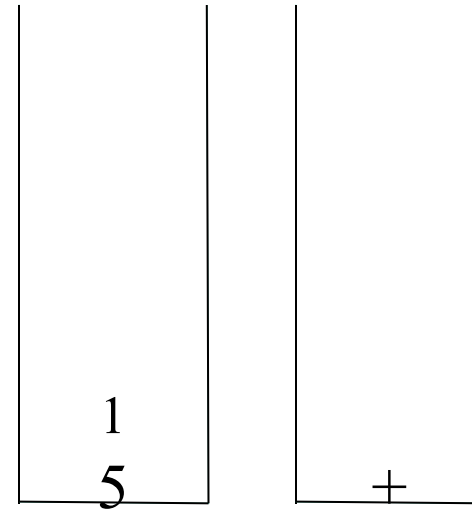
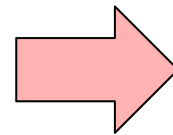
目前准备进栈的运算符如箭头所示，画出从该运算符**准备进栈到进栈完成**的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，每页三组，不够可加页）



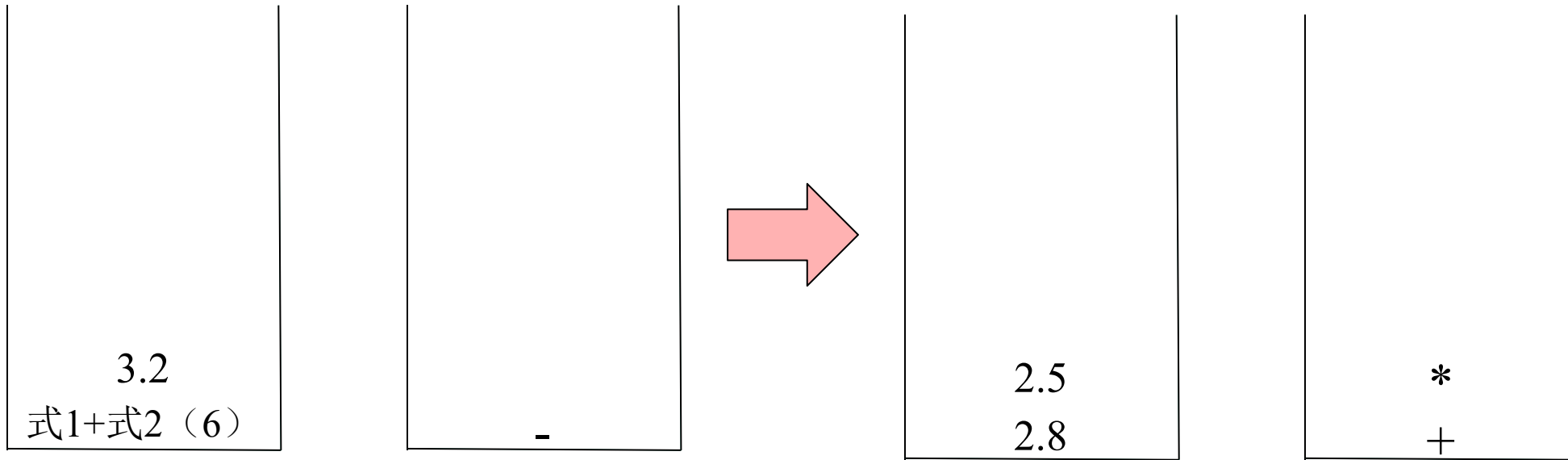
要入栈的+优先级低于/,  
左结合，故先计算 $11/2$



要入栈的-优先级低于%,  
左结合，故先计算 $37\%4$



要入栈的-优先级等于+,  
左结合，故先计算  
 $11/2+37\%4$



要入栈的+优先级等于-, 左结合, 故  
先计算 (式1+式2) 式三-3.2

## §. 基础知识题



2、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.  $a = 2 * 4$  ,  $a = b = 3 * 5$  （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:  $2 * 4$   $\Rightarrow$  式1

步骤②:  $a =$  式1  $\Rightarrow$  式2

步骤③:  $a = b$   $\Rightarrow$  式3

步骤④:  $3 * 5$   $\Rightarrow$  式4

步骤⑤: 式3  $=$  式4  $\Rightarrow$  式5

步骤⑥: 式2  $=$  式5  $\Rightarrow$  式6



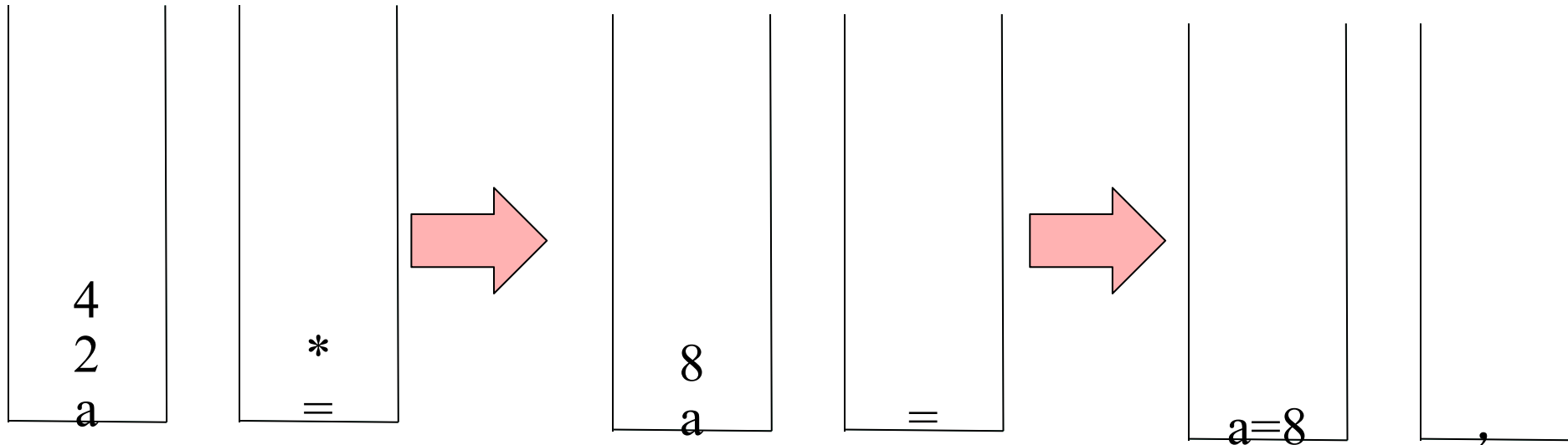
## §. 基础知识题

2、仿照课DF的P. 65-85，用栈方式给出下列表达式的求解过程

B.  $a = 2 * 4$  ,  $a = b = 3 * 5$  (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，画出从该运算符**准备进栈到进栈完成**的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，每页三组，不够可加页）



要入栈的，优先级低于\*，故  
先计算 $2*4$ （式1）

要入栈的，优先级低于=，故  
先计算 $a=$ 式1（式2）



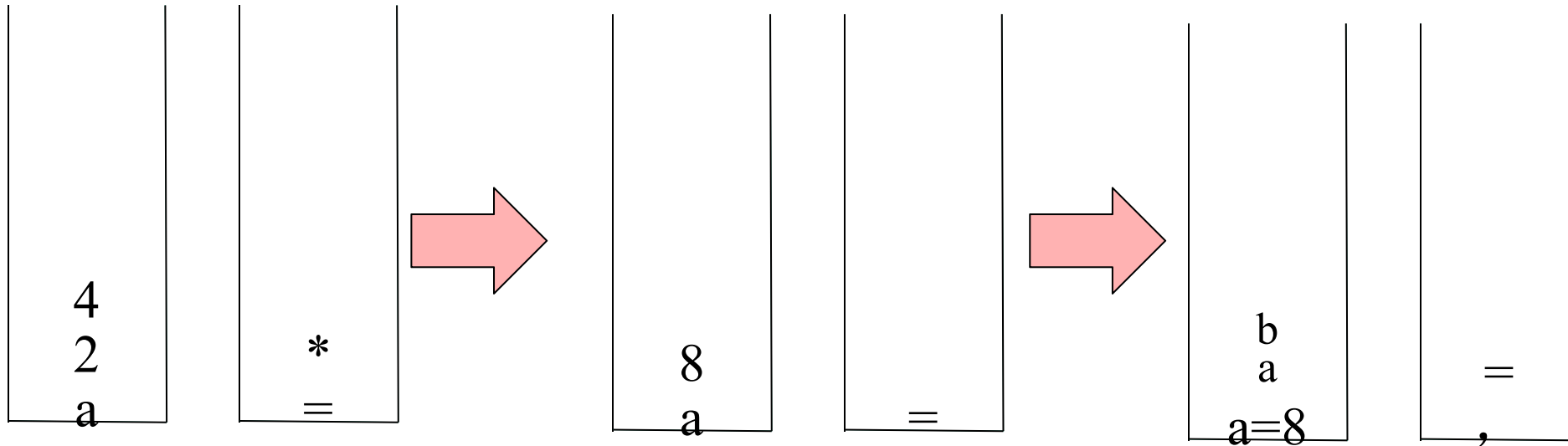
## §. 基础知识题

2、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.  $a = 2 * 4, a = b = 3 * 5$  (假设所有变量均为int型)



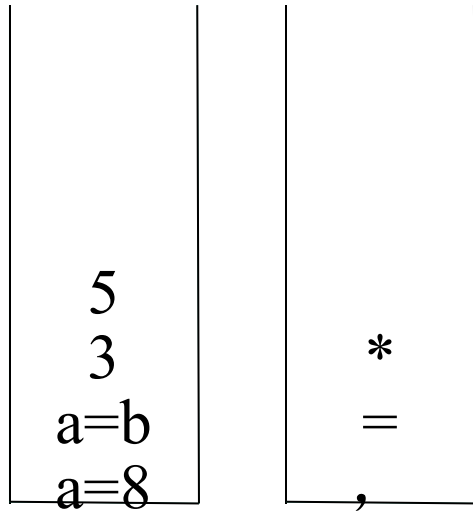
目前已分析到整个表达式的尾部，画出到表达式求值完成的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，每页三组，不够可加页）



要入栈的，优先级低于\*，故先计算 $2*4$ （式1）

要入栈的，优先级低于=，故先计算 $a=$ 式1（式2）

要进栈的=，与栈顶的=优先级相等，故先计算 $a=b$ （式3）



进栈完毕，从栈顶开始计算：

$3*5=15$ ;

$a=b=15$ ;

$a=8$ ,  $a=b=15$ ;     最终答案为15



## §. 基础知识题

2、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.  $a + (a + 3 * (b + c) - 5) \% 4$  （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 **10** 个运算符，因此计算的 **6** 个步骤分别是（左右括号不算步骤）：

步骤①:  $b + c$        $\Rightarrow$  式1

步骤②:  $3 * \text{式1}$        $\Rightarrow$  式2

步骤③:  $a + \text{式2}$        $\Rightarrow$  式3

步骤④:  $\text{式3} - 5$        $\Rightarrow$  式4

步骤⑤:  $\text{式4} \% 4$        $\Rightarrow$  式5

步骤⑥:  $a + \text{式5}$        $\Rightarrow$  式6

后面自行添加，主要是对()的理解，本页中一对括号可以当做一个步骤理解，后续画栈时要分开



## §. 基础知识题

2、仿照课件PF的P. 65-85，用栈方式给出下列表达式的求解过程

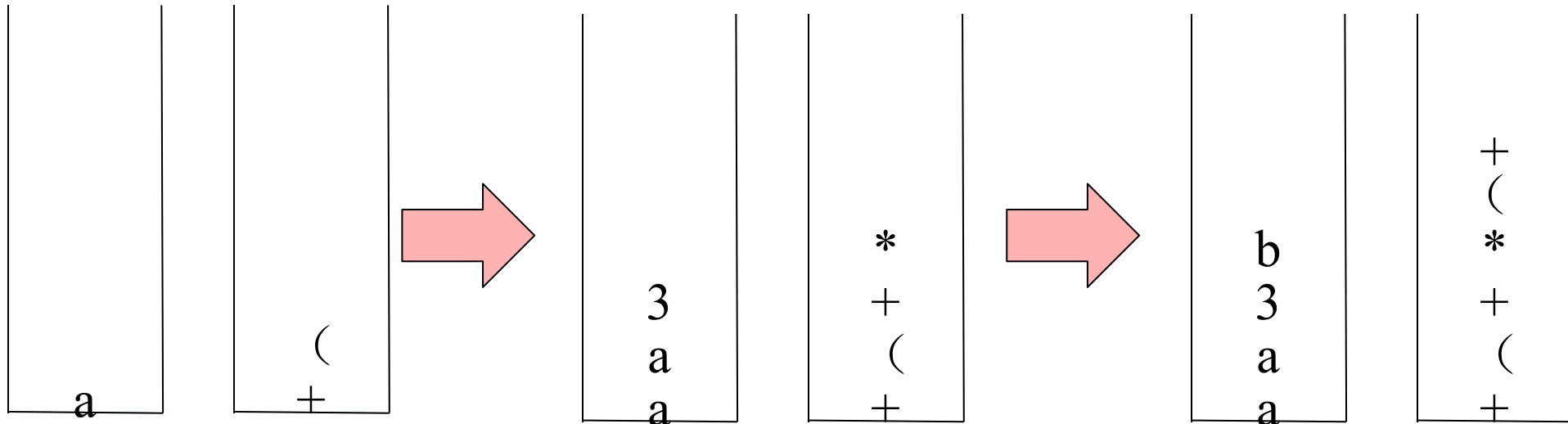
C.  $a + (a + 3 * (b + c) - 5) \% 4$  (假设所有变量均为int型)

(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，画出从该运算符**准备进栈到进栈完成**的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，每页三组，不够可加页）



进栈前（优先级最高，直接压入栈

进栈后，（优先级变为最低，故+直接入栈，之后的\*优先级高于+，也直接入栈

进栈前，（优先级最高，直接入栈；入栈后优先级变为最低，故+直接压入栈





## §. 基础知识题

2、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

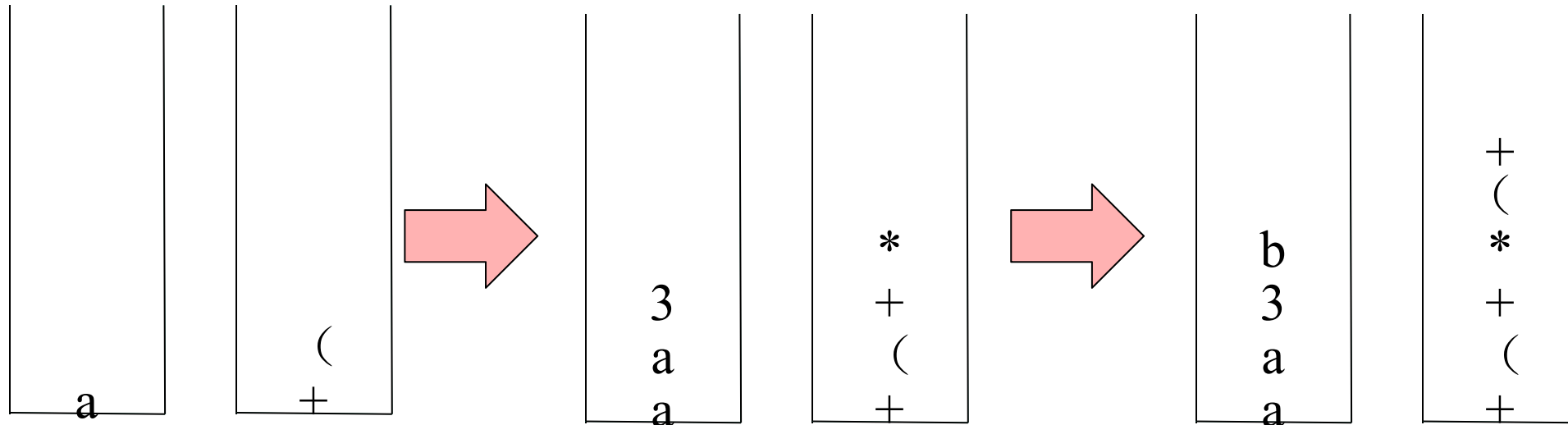
C.  $a + (a + 3 * (b + c) - 5) \% 4$  (假设所有变量均为int型)

(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

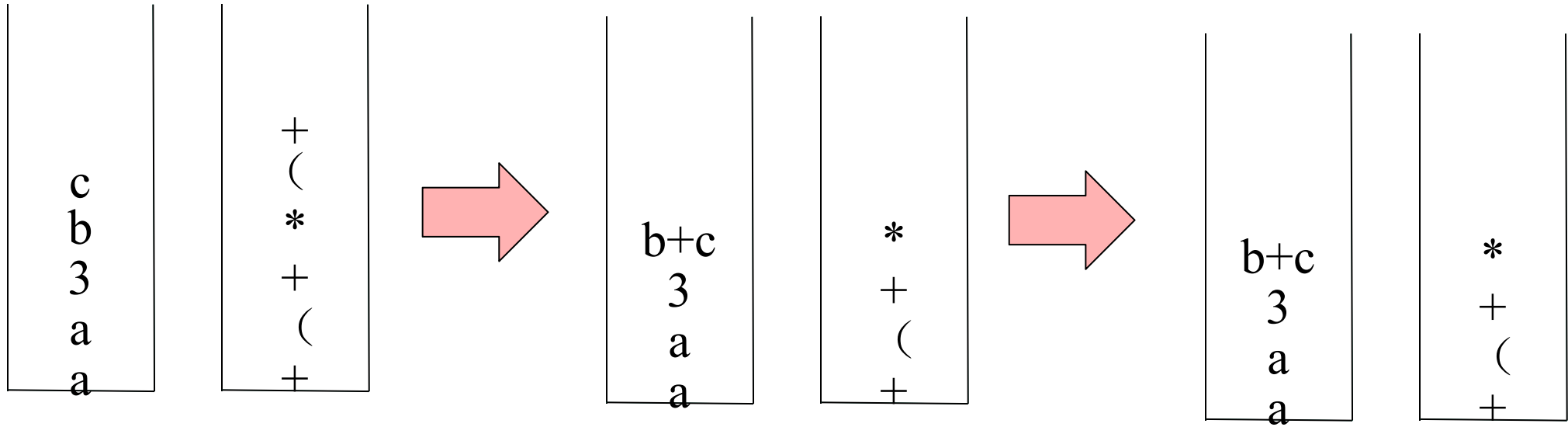
目前准备进栈的运算符如箭头所示，画出从该运算符**准备进栈到进栈完成**的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，每页三组，不够可加页）



进栈前（优先级最高，直接压入栈

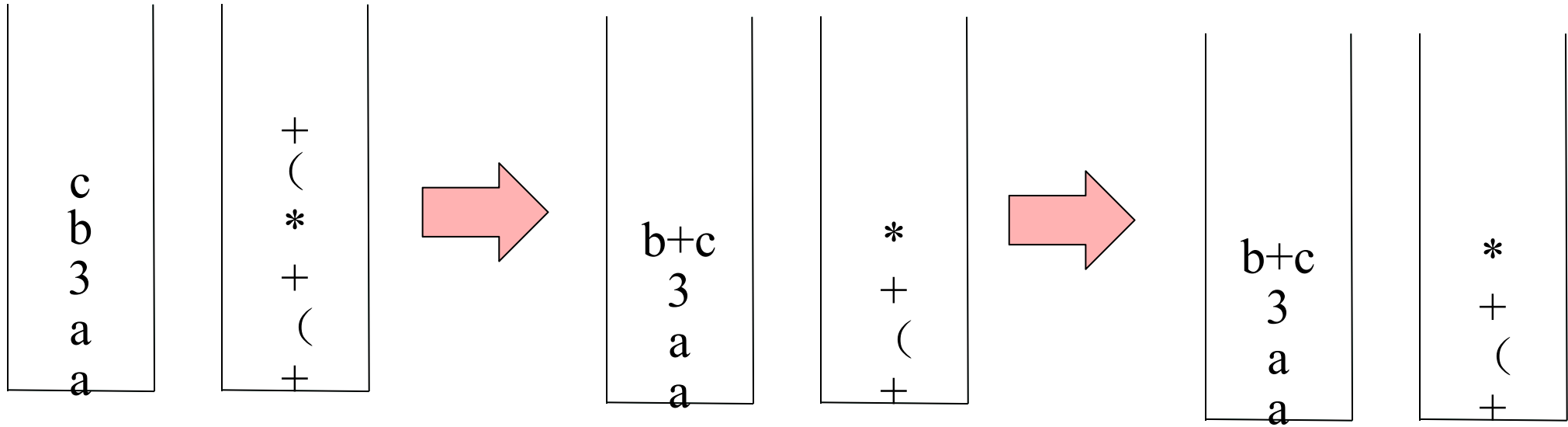
进栈后，（优先级变为最低，故+直接入栈，之后的\*优先级高于+，也直接入栈

进栈前，（优先级最高，直接入栈；入栈后优先级变为最低，故+直接压入栈



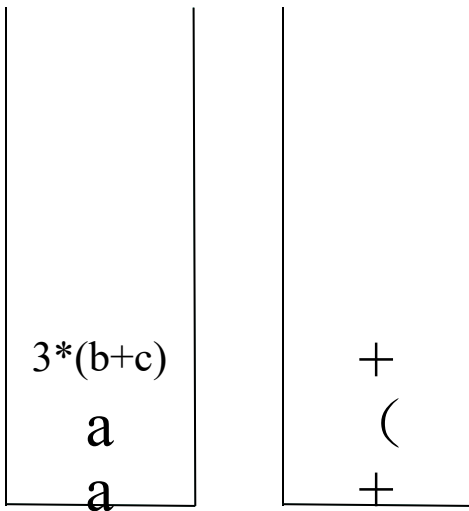
) 即将入栈, 由于) 优先级最低, 故要将栈中压在 "(" 之上的全部运算符都计算完成, 随后和 "(" 成对消除

-即将入栈, 但优先级低于\*, 故先计算  $3 * (b+c)$

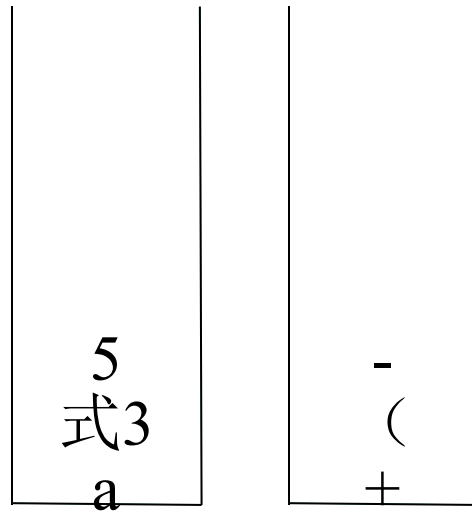
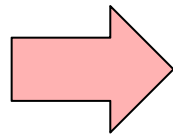


) 即将入栈, 由于) 优先级最低, 故要将栈中压在 "(" 之上的全部运算符都计算完成, 随后和 "(" 成对消除

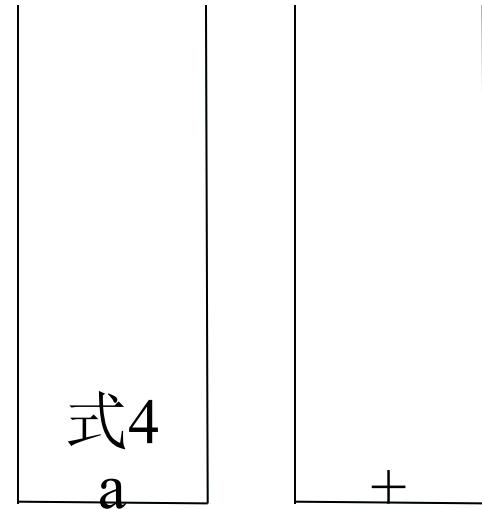
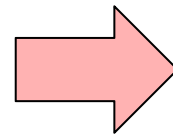
-即将入栈, 但优先级低于\*, 故先计算  $3 * (b+c)$



-即将入栈，其和+的优先级相等，且为左结合，故先计算a+式2（式3）



)即将入栈，由于)优先级最低，故要将栈中压在"("之上的全部运算符都计算完成，随后和"("成对消除



式4为“式3-5”

## §. 基础知识题

2、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

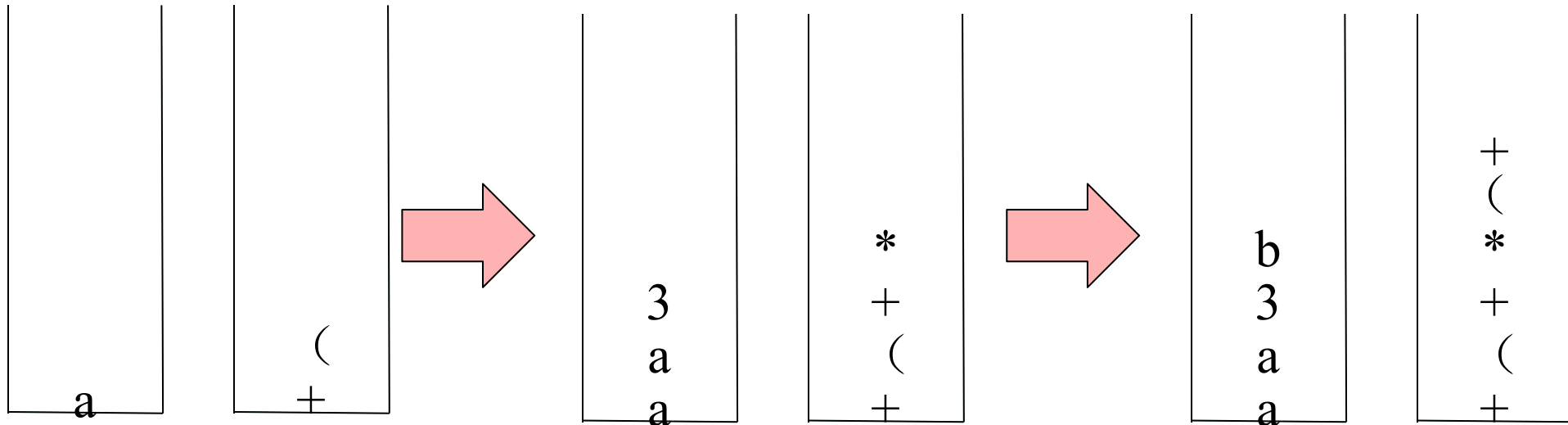
C.  $a + (a + 3 * (b + c) - 5) \% 4$  (假设所有变量均为int型)

(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

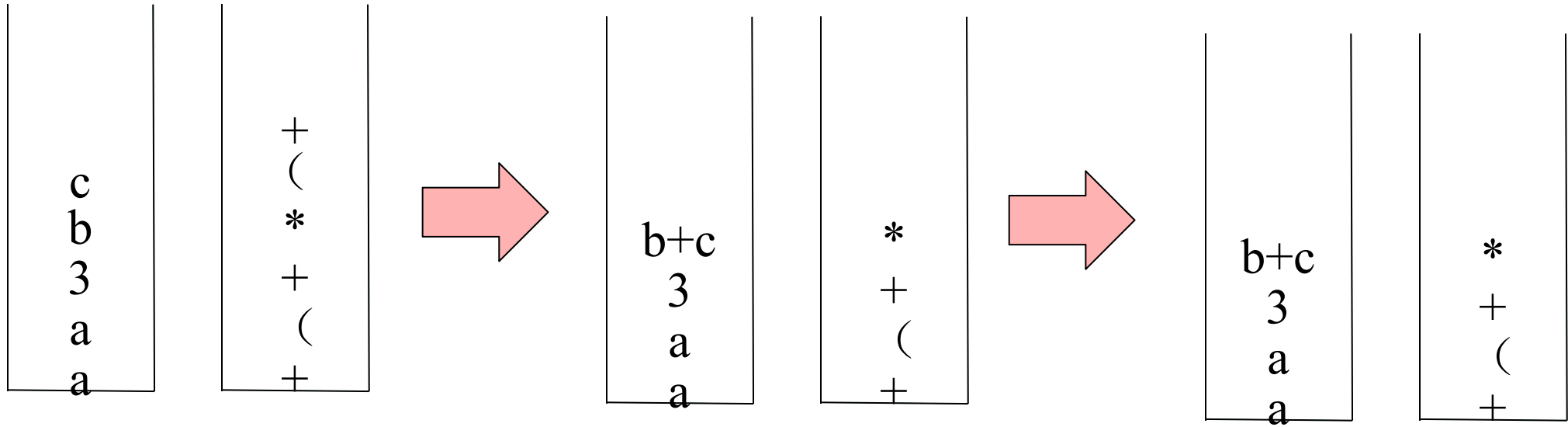
目前已分析到整个表达式的尾部，画出到表达式求值完成的过程中，当前运算数栈和运算符栈的状态（本页需要画栈，两个栈算一组，最少两组，可能多组，每页三组，不够可加页）



进栈前（优先级最高，直接压入栈

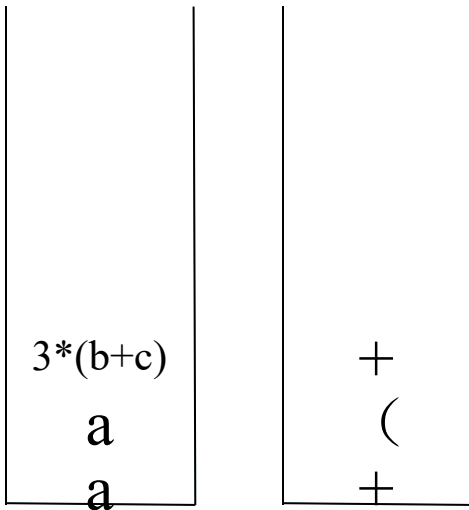
进栈后，（优先级变为最低，故+直接入栈，之后的\*优先级高于+，也直接入栈

进栈前，（优先级最高，直接入栈；入栈后优先级变为最低，故+直接压入栈

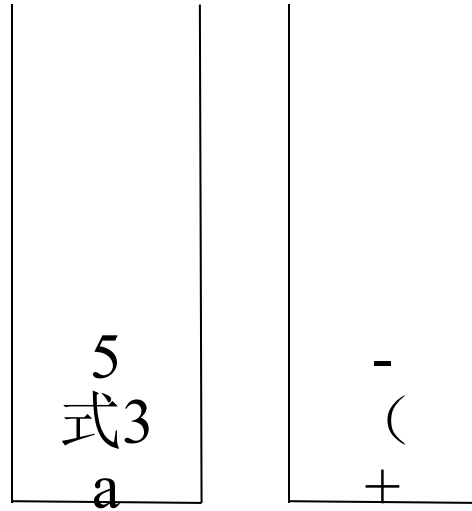
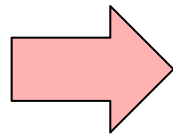


) 即将入栈, 由于) 优先级最低, 故要将栈中压在 "(" 之上的全部运算符都计算完成, 随后和 "(" 成对消除

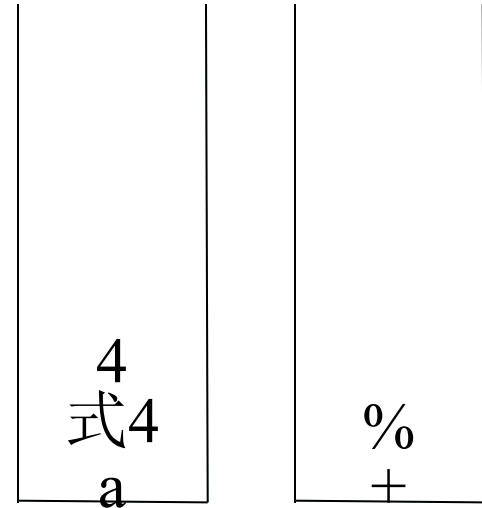
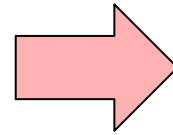
-即将入栈, 但优先级低于\*, 故先计算  $3 * (b+c)$



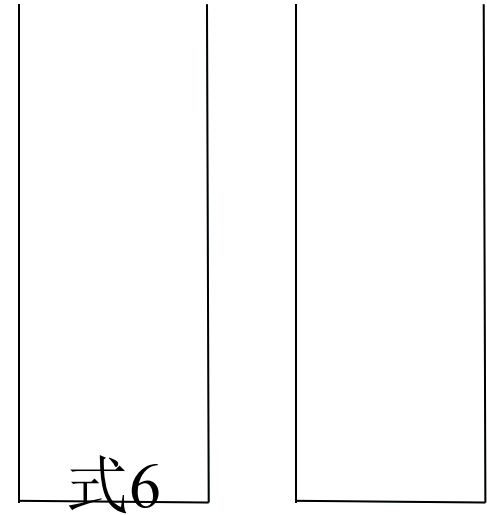
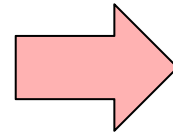
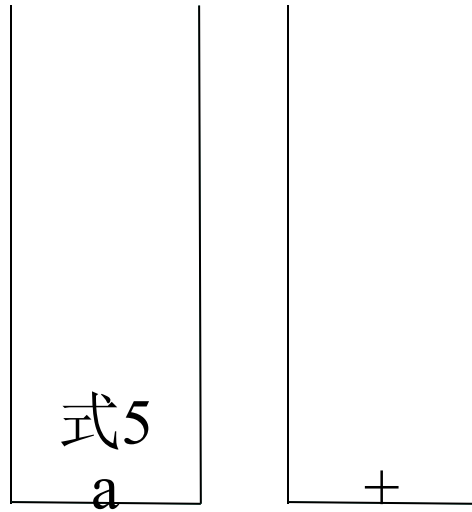
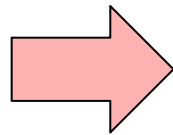
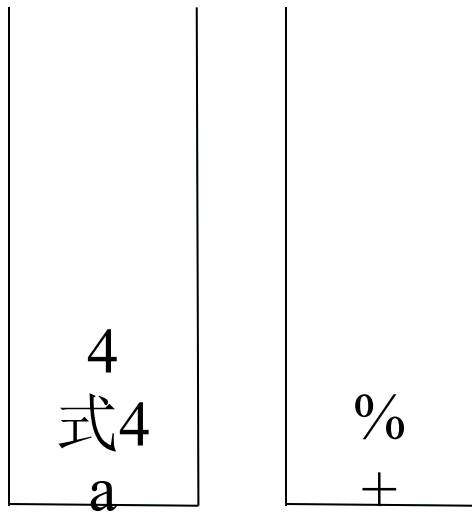
-即将入栈，其和+的优先级相等，且为左结合，故先计算 $a + \text{式2}$ （式3）



)即将入栈，由于)优先级最低，故要将栈中压在 "(" 之上的全部运算符都计算完成，随后和 "(" 成对消除



式4为“式3-5”



入栈完毕，开始计算：  
式4%4 （式5）

$a + \text{式5}$

求值完毕。





## §. 基础知识题

3、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     |

注：回答问题时，4步一定要标清楚，  
验证程序不强制，可以验证多步  
或最后一步

```
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

3、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.  $a = 2 * 4$  ,  $a = b = 3 * 5$

（写验证程序时，假设所有变量均为int型）

(1) $2*4$	=>	8	int型
(2) $a=8$	=>	$a=8$	int型
(3) $a=b$	=>	$a=b$	int型
(4) $5*3$	=>	15	int型
(5) $a=b=15$	=>	$a=b=15$	int型
(6) $a=8,a=b=15$	=>	15	int型

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int a, b;
7     cout << typeid(a = 2 * 4).name() << endl;
8     cout << typeid(a = b).name() << endl;
9     cout << typeid(a = b = 3 * 5).name() << endl;
10    cout << (a=2*4, a=b=3*5) << endl;
11    cout << typeid(a = 2 * 4, a = b = 3 * 5).name() << endl;
12    return 0;
13 }
```

(ps:只验证了部分步骤  
类型和最后一步类型)



## §. 基础知识题

3、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.  $a - (b + 3 * (b - c) \% 3) / 5$  （写验证程序时，假设所有变量均为int型，abc的值自定义即可）

(1) $(b-c)$	=>	1	int型
(2) $3 * (b-c)$	=>	3	int型
(3) $b \% 3$	=>	1	int型
(4) $b+1$	=>	5	int型
(5) $5/5$	=>	1	int型
(6) $a-1$	=>	0	int型

假设：  
 $a=1;$   
 $b=4;$   
 $c=3;$

```
文件(F) 编辑(E) 视图(V) Git(G) 项目(P) 生成(B) 调试(D) 测试(S) 分析(N) 工具(T)
Debug x86 本地 Windows 调试器
111.cpp
9.16.1 (全局范围)
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int a, b, c;
7     a = 1;
8     b = 4;
9     c = 3;
10    cout << typeid(b - c).name() << endl;
11    cout << typeid(b%3).name() << endl;
12    cout << typeid(a - (b + 3 * (b - c) % 3) / 5).name() << endl;
13    cout << a - (b + 3 * (b - c) % 3) / 5 << endl;
14
15    return 0;
16 }
```

(ps:只验证了部分步骤类型和最后一步类型)

```
Microsoft Vis
int
int
int
1
C:\Users\HP\
按任意键关闭
```



## §. 基础知识题

3、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.  $2.5 * 3UL + 4U * 7ULL - 'X'$

(1) $2.5 * 3UL$	$\Rightarrow$	7.5	double型
(2) $4U * 7ULL$	$\Rightarrow$	28	unsigned long long型
(3) $7.5 + 28ULL$	$\Rightarrow$	35.5	double型
(4) $35.5 - 'X'$	$\Rightarrow$	-52.5	double型

```
文件(F) 编辑(E) 视图(V) Git(G) 项目(P) 生成(B) 调试(D) 测试(S) 分析(N) 工具(T) 扩展(X)
Debug x86 本地 Windows 调试器
111.cpp*
9.16.1 (全局范围)
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6
7     cout << typeid(2.5*3UL).name() << endl;
8     cout << 2.5 * 3UL << endl;
9     cout << typeid(4U*7ULL).name() << endl;
10    cout << 4U * 7ULL << endl;
11    cout << typeid(2.5 * 3UL + 4U * 7ULL).name() << endl;
12    cout << 2.5 * 3UL + 4U * 7ULL << endl;
13    cout << typeid(2.5 * 3UL + 4U * 7ULL - 'X').name() << endl;
14    cout << 2.5 * 3UL + 4U * 7ULL - 'X' << endl;
15
16    return 0;
17 }
```

```
Microsoft Visual Studio 调试
double
7.5
unsigned __int64
28
double
35.5
double
-52.5
C:\Users\HP\Desktop\
按任意键关闭此窗口.
```



## §. 基础知识题

3、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.  $2LU \% 7 + 23LL \% 3 + 2.5F$

(1) $2LU \% 7$	$\Rightarrow$	2UL	unsigned long型
(2) $23LL \% 3$	$\Rightarrow$	2LL	long long型
(3) $2UL+2LL$	$\Rightarrow$	4LL	long long型
(4) $4LL+2.5F$	$\Rightarrow$	6.5F	float型

The screenshot shows the Visual Studio IDE with a C++ file named 111.cpp. The code is as follows:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6
7      cout << typeid(2LU % 7).name() << endl;
8      cout << 2LU % 7 << endl;
9      cout << typeid(23LL % 3).name() << endl;
10     cout << 23LL % 3 << endl;
11     cout << typeid(2LU % 7 + 23LL % 3).name() << endl;
12     cout << 2LU % 7 + 23LL % 3 << endl;
13     cout << typeid(2LU % 7 + 23LL % 3 + 2.5F).name() << endl;
14     cout << 2LU % 7 + 23LL % 3 + 2.5F << endl;
15
16     return 0;
17 }
```

The output window on the right shows the following results:

```
unsigned long
2
__int64
2
__int64
4
float
6.5
```

The output matches the expected results from the table above.





## §. 基础知识题

3、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.  $2.3 + 14 \% 5 * \text{static\_cast}\langle\text{unsigned long}\rangle(2.8\text{F} + 7\text{LL}) \% 2 * 2.3\text{F}$

- |   |    |      |                |
|---|----|------|----------------|
| (1) $14 \% 5$                                   | => | 4    | int型           |
| (2) $2.8\text{F} + 7\text{LL}$                  | => | 9.8F | float型         |
| static_cast<unsigned long>后转成unsigned long型 9UL |    |      |                |
| (3) $4 * 9\text{UL}$                            | => | 36UL | unsigned long型 |
| (4) $36\text{UL} \% 2$                          | => | 0UL  | unsigned long型 |
| (5) $0\text{UL} * 2.3\text{F}$                  | => | 0F   | float型         |
| (6) $2.3 + 0\text{F}$                           | => | 2.3  | double型        |

```
111.cpp* 9.16.1 (全局范围) main()
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6
7     cout << typeid(2.8F + 7LL).name() << endl;
8     cout << 2.8F + 7LL << endl;
9     cout << typeid(static_cast<unsigned long>(2.8F + 7LL)).name() << endl;
10    cout << static_cast<unsigned long>(2.8F + 7LL) << endl;
11    cout << typeid(14 % 5 * static_cast<unsigned long>(2.8F + 7LL)).name() << endl;
12    cout << 14 % 5 * static_cast<unsigned long>(2.8F + 7LL) << endl;
13    cout << typeid(14 % 5 * static_cast<unsigned long>(2.8F + 7LL) % 2).name() << endl;
14    cout << 14 % 5 * static_cast<unsigned long>(2.8F + 7LL) % 2 << endl;
15    cout << typeid(14 % 5 * static_cast<unsigned long>(2.8F + 7LL) % 2 * 2.3F).name() << endl;
16    cout << 14 % 5 * static_cast<unsigned long>(2.8F + 7LL) % 2 * 2.3F << endl;
17    cout << typeid(2.3 + 14 % 5 * static_cast<unsigned long>(2.8F + 7LL) % 2 * 2.3F).name() << endl;
18    cout << 2.3 + 14 % 5 * static_cast<unsigned long>(2.8F + 7LL) % 2 * 2.3F << endl;
19
20    return 0;
21 }
```

```
Microsoft Visual Studio 调试
float
9.8
unsigned long
9
unsigned long
36
unsigned long
0
float
0
double
2.3
C:\Users\HP\Desktop\9.
按任意键关闭此窗口.
```



## §. 基础知识题

3、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.  $\text{long}(2.8 + 3.3) / 2 + (\text{int})1.9 \% 7\text{LU} - 'g' * 2\text{L}$

(1) $2.8 + 3.3$	$\Rightarrow$ 6.1	double型
$\text{long}(2.8 + 3.3)$ 后转成long型	6L	
(2) $6\text{L} / 2$	$\Rightarrow$ 3L	long型
(3) $(\text{int})1.9 \% 7\text{LU}$	$\Rightarrow$ 1UL	unsigned long型
(4) $3\text{L} + 1\text{UL}$	$\Rightarrow$ 4UL	unsigned long型
(5) $'g' * 2\text{L}$	$\Rightarrow$ 206L	long型
(6) $4\text{UL} - 206\text{L}$	$\Rightarrow$ 4294967094UL	unsigned long型

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6
7      cout << typeid(long(2.8 + 3.3)).name() << endl;
8      cout << long(2.8 + 3.3) << endl;
9      cout << typeid(long(2.8 + 3.3) / 2).name() << endl;
10     cout << long(2.8 + 3.3) / 2 << endl;
11     cout << typeid((int)1.9 % 7LU).name() << endl;
12     cout << (int)1.9 % 7LU << endl;
13     cout << typeid(long(2.8 + 3.3) / 2 + (int)1.9 % 7LU).name() << endl;
14     cout << long(2.8 + 3.3) / 2 + (int)1.9 % 7LU << endl;
15     cout << typeid('g' * 2L).name() << endl;
16     cout << 'g' * 2L << endl;
17     cout << typeid(long(2.8 + 3.3) / 2 + (int)1.9 % 7LU - 'g' * 2L).name() << endl;
18     cout << long(2.8 + 3.3) / 2 + (int)1.9 % 7LU - 'g' * 2L << endl;
19
20     return 0;
21 }
```

Microsoft Visual Stud

```
long
6
long
3
unsigned long
1
unsigned long
4
long
206
unsigned long
4294967094
```

C:\Users\HP\Desktop  
按任意键关闭此窗口

## §. 基础知识题



4、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

4、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 = 7, n = 11;`

A. `a += a - n`


- (1) `a-n`          差-4存放在中间变量中 `a=7, n=11;`
- (2) `a += a - n => a = a + (a-n)` 和3存放在中间变量中 `a=7, n=11;`
- (3) 和赋给a          **`a=3, n=11`**

```

#include <iostream>
using namespace std;

int main()
{
    int a = 7, n = 11;
    cout << a - n << endl;
    cout << a << ' ' << n << endl;
    cout << a + (a-n) << endl;
    cout << a << ' ' << n << endl;
    a += a - n;
    cout << a << ' ' << n << endl;

    return 0;
}
```





## §. 基础知识题

4、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 = 7, n = 11;`

B. `n += a += 5`

(1) `a += 5` （记为式1）

=> `a + 5`                      和12存放在中间变量中 `a=7, n=11;`

=> `a=和`                          `a=12, n=11;` 式1的值为12

(2) `n += 式1`

=> `n+式1`                      和23存放在中间变量中 `a=12, n=11;`

=> `n=和`                          `a=12, n=23;`

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

```
int main()
{
    int a = 7, n = 11;

    n += a += 5;
    cout << a << ' ' << n << endl;

    return 0;
}
```

```
C:\Users\HP\
12 23
按任意键关闭
```



## §. 基础知识题

4、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 = 7, n = 11;`

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

(1) `a *= a` （记为式1）

=> `a * a`          积49存放在中间变量中 `a=7, n=11;`

=> `a=积`              `a=49, n=11;` 式1的值为49;

(2) `a += 式1` （记为式2）

=> `a+式1`          和98存放在中间变量中 `a=49, n=11;`

=> `a=和`              `a=98, n=11;` 式2的值为98

(3) `a += 式2`

=> `a+式2`          和196存放在中间变量中 `a=98, n=11;`

=> `a=和`              `a=196, n=11;`

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

```
int main()
{
    int a = 7, n = 11;
    a += a += a *= a;
    cout << a << ' ' << n << endl;

    return 0;
}
```

Microsoft Visual Studio  
196 11  
C:\Users\HP\Desktop  
按任意键关闭此窗口



## §. 基础知识题

4、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 = 6, n = 11;`

D. `n %= a %= 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）

(1) `a %= 3` （记为式1）

=> `a % 3`                  模0存放在中间变量中 `a=6, n=11;`

=> `a=模`                      `a=0, n=11;` 式1的值为0

(2) `n %= 式1`

=> `n%式1`

=> `n=模`

?? ??

?? ??

合理猜测：

不报错是因为程序本身没有语法错误，  
但式1值为0，**而0不能做除数**，除0取余没办法计算。