实 验 报 告

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 班级学号 | 202015116 | 班级排序号 | 200124 | 姓名 | 梁祺若 | 实验日期 | 2021-5-17 |
| 实验名称 | | | | 实验八: 指针 | | | |

**第1题：**阅读文件夹“pointer”里所有的代码（共计24个）。针对每一段代码，回答下列问题：

（1）此段代码编译时是否会出现编译错误？若是请分析原因。

（2）若此段代码不出现编译错误，则请解释其运行结果。

代码1：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

0x6ffe14

0x6ffe14

4

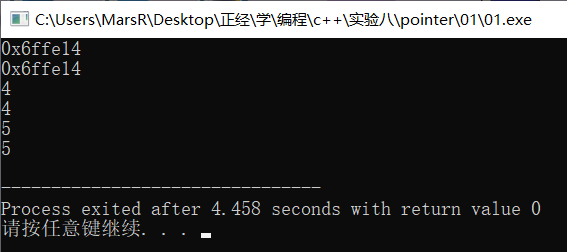
4

5

5

原因为：先输出i和pointer的地址（16进制），再输出i和pointer指向的i的值为4，后i++，pointer 指向的++，再输出。

此段代码的实际运行结果如下：



代码2：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

3

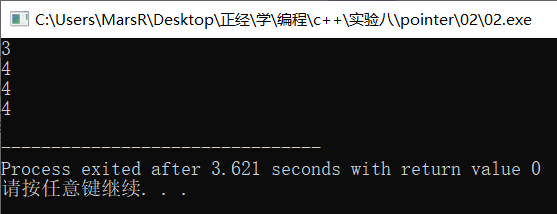
4

4

4

原因为：先输出p和q指向的数字3和4，再将q的值赋给p，故p为4，此时再输出p和q的值。

此段代码的实际运行结果如下：



代码3：此段代码编译时会出现编译错误，出错的位置位于第7行，出错的原因是pointer为指向int型指针，而i为float型。

代码4：此段代码编译时会出现编译错误，出错的位置位于第8行，出错的原因是int和float的存储方式不同，导致存储空间混乱，输出错误。

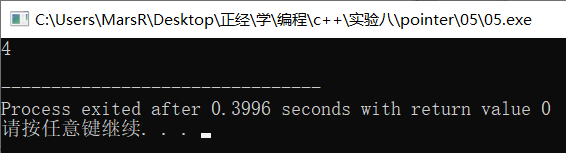
代码5：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

4

原因为：强制转换时格式正确，能正确输出float型的pointer指向的数。

此段代码的实际运行结果如下：



代码6：此段代码编译时会出现编译错误，出错的位置位于第9行，出错的原因是pointer为void型，无法直接输出。

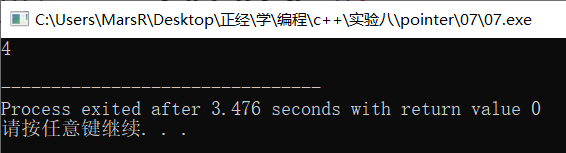
代码7：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

4

原因为：void被强制转化为float型输出。

此段代码的实际运行结果如下：



代码8：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

0x6ffe0c

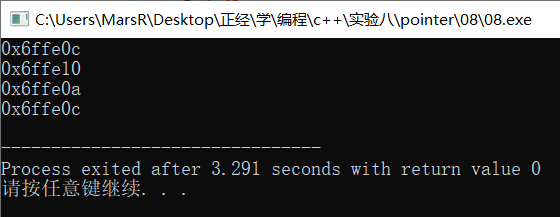
0x6ffe10

0x6ffe0a

0x6ffe0c

原因为：

此段代码的实际运行结果如下：输出要求为地址，int型占4个字节，short int 为2字节，故++后输出结果+4/+2.



代码9：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

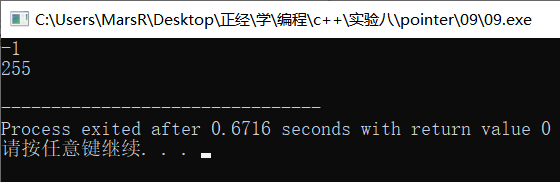
-1

255

原因为：

i在内存中存储为0000FFFF，pointer存为FF，char只有1个字节故q为00FF，十进制为255。FFFF的补码为-1。

此段代码的实际运行结果如下：



代码10：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

1

1

1

1

1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2

2

2

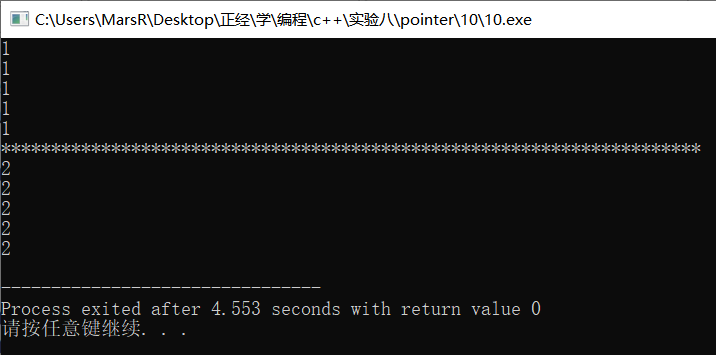
2

2

原因为：

数组名就是首字符，故按要求输出前五个都为首字符。后五个输出都为第二位字符。

此段代码的实际运行结果如下：



代码11：此段代码编译时不会出现编译错误。

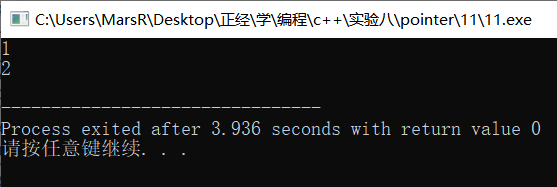
此段代码的预期运行结果如下：

1

2

原因为：第一个输出要求为首位字符，第二个输出要求输出后一位字符。

此段代码的实际运行结果如下：



代码12：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

1

2

1

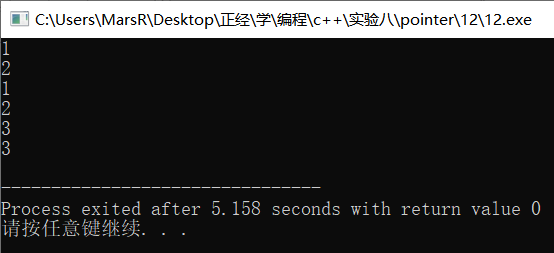
2

3

3

原因为：第一个输出要求为首位字符，第二个输出要求输出后一位字符。

此段代码的实际运行结果如下：



代码13：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

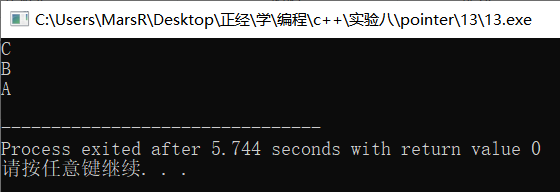
C

B

A

原因为：num实际数据为00410000+00004200+43，存储时为43424100

此段代码的实际运行结果如下：



代码14：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

Hello

Hello

Hello

Hello

Hello

Hello

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1

0

0

0

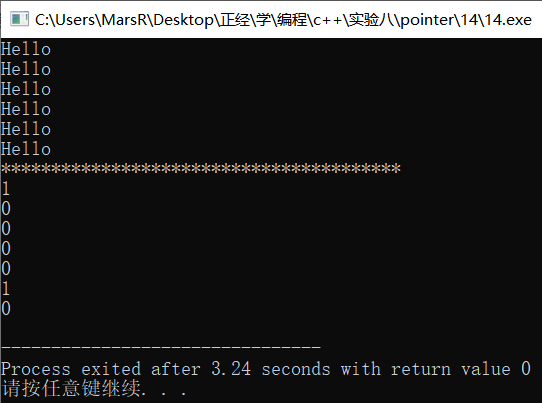
0

1

0

原因为：前六个均为输出存储内容，即输出hello。后面是判断是否相等，故输出0和1。

此段代码的实际运行结果如下：



代码15：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

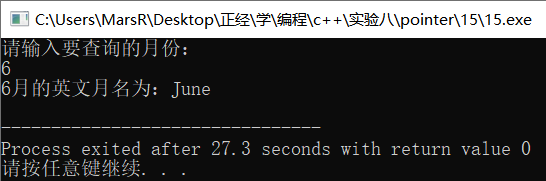
请输入要查询的月份：

6

6月的英文月名为：June

原因为：把月份名称定义为数组，需要查询的月份就是相对应位置的英语单词。

此段代码的实际运行结果如下：



代码16：此段代码编译时不会出现编译错误。

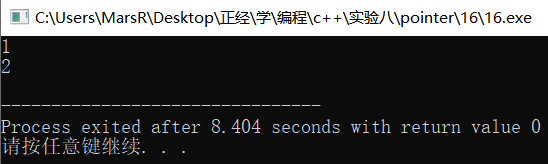
此段代码的预期运行结果如下：

1

2

原因为：i原来定义为1，后经过fun函数变为2，并输出。

此段代码的实际运行结果如下：



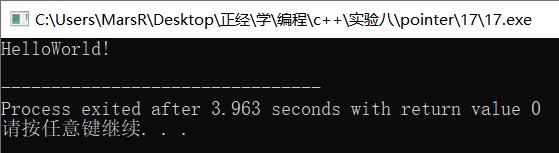
代码17：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

HelloWorld!

原因为：定义一个fun函数，运行并输出结果，输出i的内容。

此段代码的实际运行结果如下：



代码18：此段代码编译时不会出现编译错误。

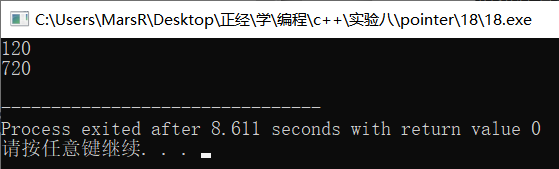
此段代码的预期运行结果如下：

120

720

原因为：定义了一个fact函数，作用为从输入的数字递减相乘到1，故第一个为5\*4\*3\*2\*1=120，第二个为6\*5\*4\*3\*2\*1=720。

此段代码的实际运行结果如下：



代码19：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

1

1

1

1

原因为：定义的output函数能够输出，同时cout也能输出，此时，会输出四个结果。同时分析程序得到四个结果都应为1.

此段代码的实际运行结果如下：



代码20：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

0x6ffe10

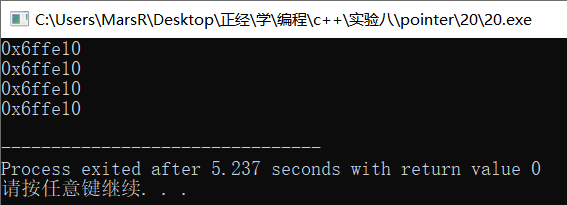
0x6ffe10

0x6ffe10

0x6ffe10

原因为：分析程序得到输出结果都应为地址，故都应为16进制数。同时，都要求输出i的地址，故四个输出结果应该相同。

此段代码的实际运行结果如下：



代码21：此段代码编译时会出现编译错误，出错的位置位于第6行，出错的原因是此处year，month，day被定义为私有数据，故后续调用无法改变他们的值，故出错。

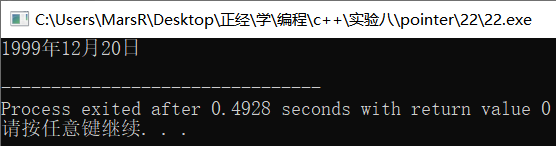
代码22：此段代码编译时不会出现编译错误。

此段代码的预期运行结果如下：

1999年12月20日

原因为：最后输出时，实在函数内输出，此时函数可以被改变，故结果正确。

此段代码的实际运行结果如下：



代码23：此段代码编译时会出现编译错误，出错的位置位于第21行，无输出内容。

代码24：此段代码编译时会出现编译错误，出错的位置位于第23行，无输出内容。