§14.C 和 C++知识点补充

【注意:】

1、 除明确要求外，已学过的知识中，不允许使用 goto、不允许使用全局变量，不允许使用 C++的 string 变量，不允许使用 STL 容器等后续知识
2、 多编译器下均要做到“0 errors, 0 warnings”
3、 部分题目要求 C 和 C++两种方式实现，具体见网页要求
4、 给出的 demo.exe 均为 cmd 下运行

【特别说明:】

1、 第 14 章视频学习+基础作业请大家抽时间尽快完成，10.27 日前无大作业
2、 union 无单独作业，需要使用时会特别提醒（基本概念考试仍会涉及）
3、 enum/typedef 无专门的作业，是否在自己的作业中采用可自主决定（基本概念考试仍会涉及）

补充:

1、用位运算方式记录多开关的状态切换

假设一共有 A-J 共 10 个开关，相互之间独立，没有依赖关系，每个开关都仅有 ON/OFF 两种状态，初始值全部为 OFF。

【要求:】1、输入 A ON / J OFF 的形式(大小写不限/有无空格不限)，表示某个开关置 ON/OFF

2、程序要循环输入，输入错误则继续输入，输入 Q on/off 则表示结束

3、所有开关的状态，只能记录在一个 short 型变量中，即使用 short 型变量的 16 个 bit 中的低 10 个 bit 来表示（右起第 1bit 表示 A，第 10bit 表示 J）

4、程序执行后，输出 10 个开关的初始状态，每次改变某个开关的状态后，都输出 10 个开关的状态，以验证设置是否正确（主要看是否影响到其它开关/本质就是对某个 bit 位置 0/1 而不要影响其它 bit 位）

例：short on-off-switch = 0x0000;

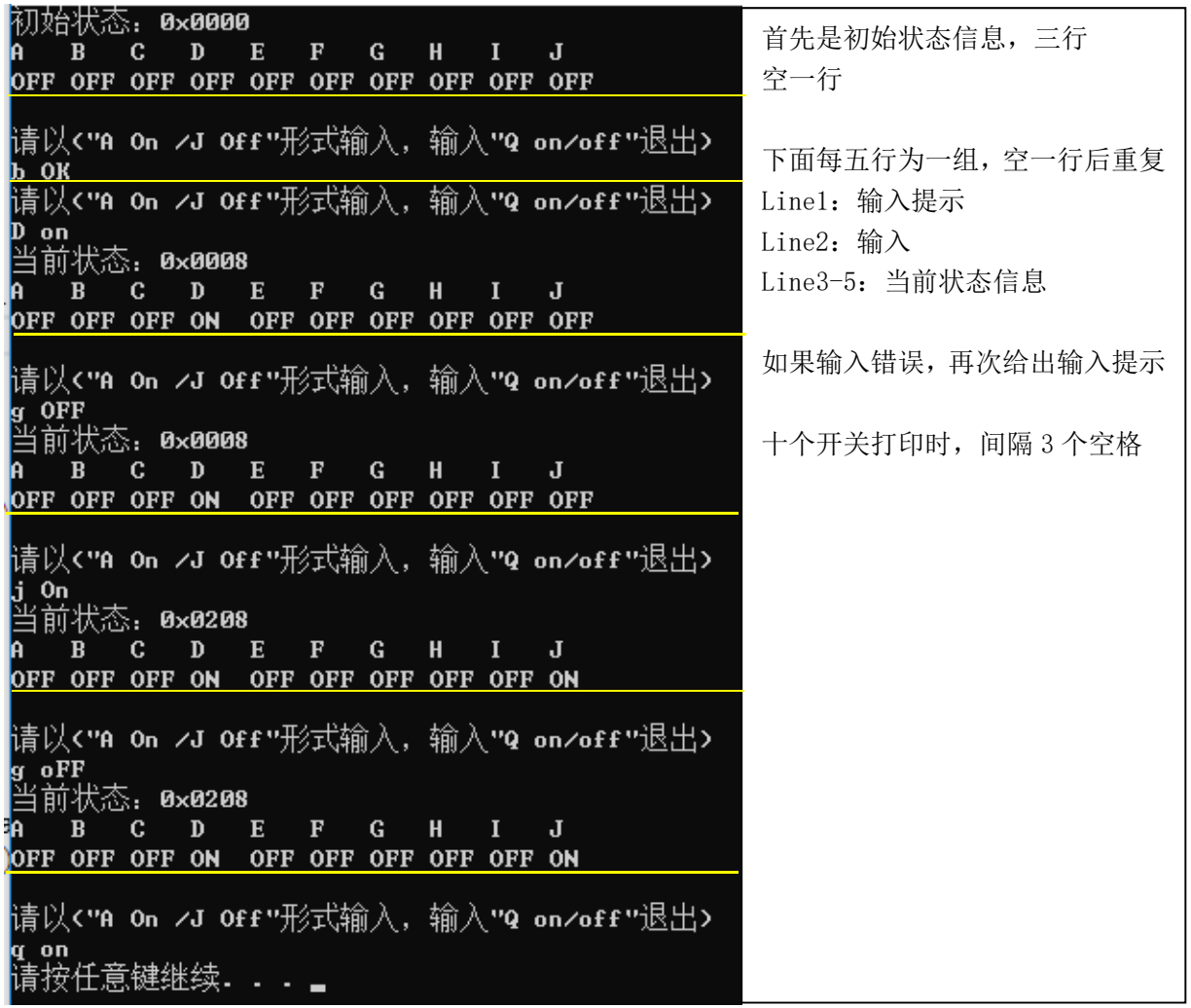
依次输入：D ON， 则 short on-off-switch 为 0x0008
G ON， 则 short on-off-switch 为 0x0048
J ON， 则 short on-off-switch 为 0x0248
G OFF，则 short on-off-switch 为 0x0208

输出形式如下（假设 on-off-switch 的值是 0x0208）：

A	B	C	D	E	F	G	H	I	J
OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON

注：输出顺序与 short 数据的 bit 顺序是反序的

5、给出 14-b1-demo.exe 供参考



首先是初始状态信息，三行
空一行

下面每五行为一组，空一行后重复
Line1：输入提示
Line2：输入
Line3-5：当前状态信息

如果输入错误，再次给出输入提示

十个开关打印时，间隔 3 个空格

2、模拟斗地主的发牌程序

【要求:】1、斗地主的基本规则：一副扑克牌，54 张，三人参加游戏，首先按顺序每人发 17 张牌，然后键盘输入一个地主，再将最后剩余的三张牌发给地主

2、发牌过程必须在 3 人间轮流，不允许一个人发完 17 张牌后，再发下一个人，要求每发完一轮，打印三个人的牌面信息

3、三个玩家，每个人的牌面信息只允许记录在一个 64bit 的整数（long long int）中，给出 14-b2.cpp，按要求将代码补充完整即可（注意每个函数内部的具体要求）

4、给出 Windows 下的 14-b2-demo.exe 供参考

5、给出 Linux 下的 14-b2-demo 供参考（在$下输入 14-b2-demo 即可运行，不需要./）
6、扑克牌的花色符号的 ASCII 码对应值(Club-5 Diamond-4 Heart-3 Spade-6)，在 Windows 下要做到打印花色，Linux 因为字体设置问题，花色换为字母 CDHS 即可（Windows/Linux 的显示差异通过条件编译方式来实现）
7、程序调试（包括运行 demo 程序）时，为了保证 ASCII 码的花色符号显示正确，cmd 窗口的字体一定要设置为**点阵字体，大小 8*16**

3、完成作业相似度检查程序的参数解析

说明：能完成以下四种条件的五个参数的任意正确组合并分析解析结果

(1) 学生的匹配

要求能在两个特定的学生之间检查

某个特定学生和全体学生之间检查

全体学生之间相互检查

★ 除“all”（纯小写）表示全体学生外，其余均表示某个具体学号，要求 7 位，纯数字
★ 如果要检查的学生是 all，则匹配学生必须是 all
★ 如果要检查的学生的学号和匹配学生的学号同时错误，则报检查学生学号错
★ 检查学生的学号错误分别是“要检查的学号不是 7 位数字”、“要检查的学号不是 7 位”
★ 匹配学生的学号错误分别是“要匹配的学号不是 7 位数字”、“要匹配的学号不是 7 位”、“检查学号是 all，匹配学号必须是 all”

(2) 文件的匹配

要求既可以是单文件，也可以全部文件

★ 除“all” （纯小写）表示所有文件外，其余均表示某个具体文件名，不需要判断文件是否存在
★ 文件名长度超过 32 字节则给出“源程序文件名超过了 32 字节”的错误

(3) 相似度设置

要求值在 60-100 间浮动

★ 如果给出的范围不正确，取缺省值 80

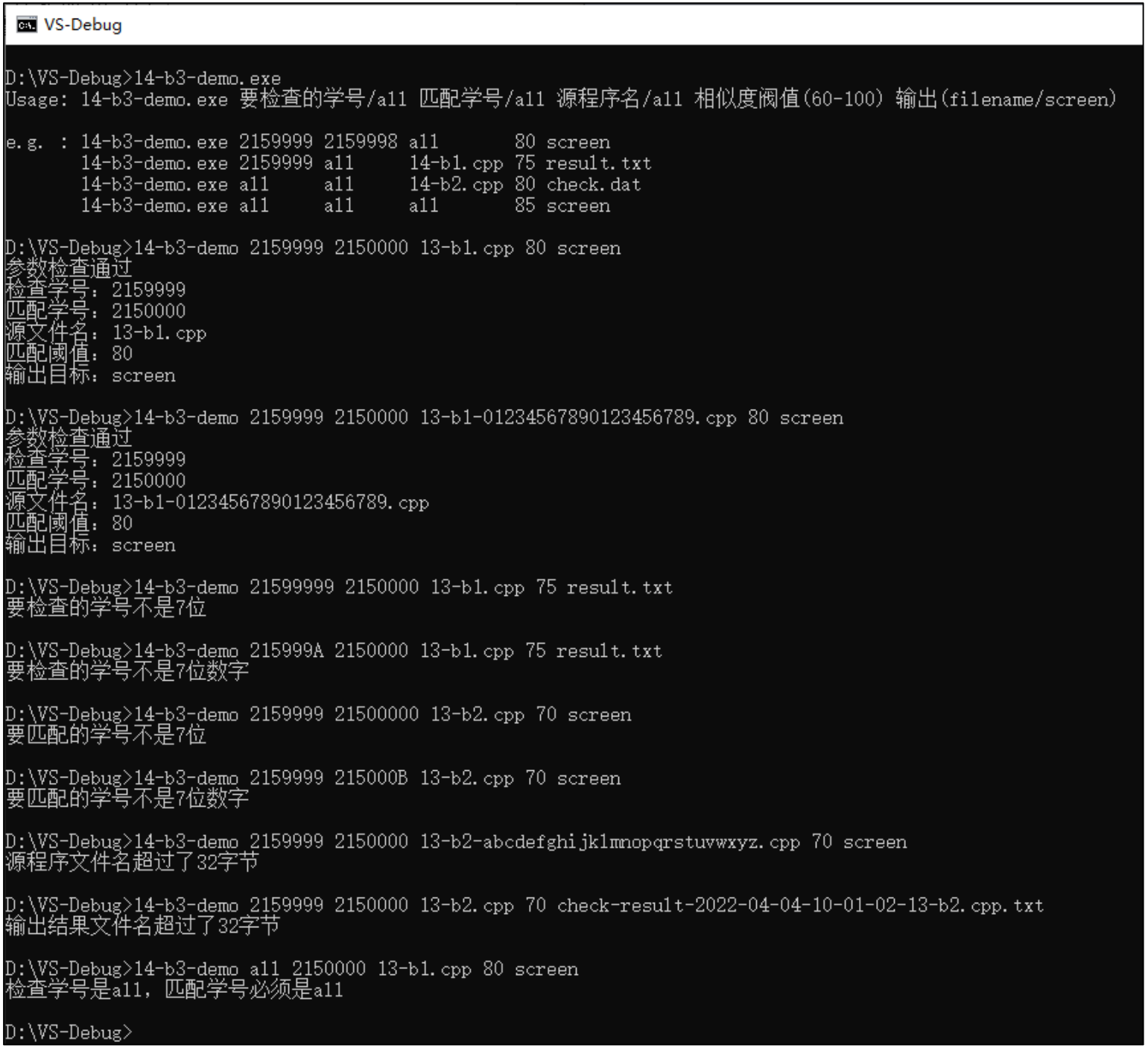
(4) 输出方式

既可以将结果输出到某个文件中，也可以直接输出到屏幕上

★ 除“screen” （纯小写）表示屏幕外，其余均表示某个具体文件名，不需要判断文件名是否合理
★ 文件名长度超过 32 字节则给出“输出结果文件名超过了 32 字节”的错误

要求：

(1) 如果给的参数不足 5 个，则调用 usage 函数给出提示即可（procname 为 argv[0]）,usage 函数见附件
(2) 给出 14-b3-demo.exe 供参考（注意：把 cmd 的当前目录设为 14-b3-demo.exe 所在目录，不要拖曳运行）
(3) 建议：本程序在 cmd 下调试比集成环境下方便（具体方法：将 cmd 快捷方式的“起始位置”设置为解决方案的 Debug 目录即可）


VS-Debug
D:\VS-Debug>14-b3-demo.exe
Usage: 14-b3-demo.exe 要检查的学号/a11 匹配学号/a11 源程序名/a11 相似度阀值(60-100) 输出(filename/screen)
e.g. : 14-b3-demo.exe 2159999 2159998 a11 80 screen
14-b3-demo.exe 2159999 a11 14-b1.cpp 75 result.txt
14-b3-demo.exe a11 a11 14-b2.cpp 80 check.dat
14-b3-demo.exe a11 a11 a11 85 screen
D:\VS-Debug>14-b3-demo 2159999 2150000 13-b1.cpp 80 screen
参数检查通过
检查学号：2159999
匹配学号：2150000
源文件名：13-b1.cpp
匹配阈值：80
输出目标：screen
D:\VS-Debug>14-b3-demo 2159999 2150000 13-b1-01234567890123456789.cpp 80 screen
参数检查通过
检查学号：2159999
匹配学号：2150000
源文件名：13-b1-01234567890123456789.cpp
匹配阈值：80
输出目标：screen
D:\VS-Debug>14-b3-demo 21599999 2150000 13-b1.cpp 75 result.txt
要检查的学号不是7位
D:\VS-Debug>14-b3-demo 215999A 2150000 13-b1.cpp 75 result.txt
要检查的学号不是7位数字
D:\VS-Debug>14-b3-demo 2159999 21500000 13-b2.cpp 70 screen
要匹配的学号不是7位
D:\VS-Debug>14-b3-demo 2159999 215000B 13-b2.cpp 70 screen
要匹配的学号不是7位数字
D:\VS-Debug>14-b3-demo 2159999 2150000 13-b2-abcdefghijklmnopqrstuvwxyz.cpp 70 screen
源程序文件名超过了32字节
D:\VS-Debug>14-b3-demo 2159999 2150000 13-b2.cpp 70 check-result-2022-04-04-10-01-02-13-b2.cpp.txt
输出结果文件名超过了32字节
D:\VS-Debug>14-b3-demo a11 2150000 13-b1.cpp 80 screen
检查学号是a11，匹配学号必须是a11
D:\VS-Debug>

(4) 下表为部分组合及测试结果（注：表中分析结果仅为示例，限于宽度，未列出超长文件名，具体的输出信息要求与 demo 保持一致）

命令	分析结果	检查学号	匹配学号	文件名	相似度	输出
14-b3 2159999 2159998 13-b3.cpp 80 screen	正确	2159999	2159998	13-b3.cpp	80	screen
14-b3 2159999 2159998 all 75 all.dat	正确	2159999	2159998	All	75	all.dat
14-b3 2159999 all 13-b3.cpp 80 screen	正确	2159999	All	13-b3.cpp	80	screen
14-b3 2159999 all all 70 all.txt	正确	2159999	All	All	70	all.txt
14-b3 all all all 85 final.dat	正确	All	All	All	85	final.txt
14-b3 2159999 2159998 13-b3.cpp 50 screen	正确	2159999	2159998	13-b3.cpp	80	screen
14-b3 all 2159998 all 85 final.dat	匹配学号错误					
14-b3 215abcd 2159998 13-b3.cpp 80 screen	检查学号错误					
14-b3 2159999 21599998 13-b3.cpp 80 screen	匹配学号错误					
14-b3 215abcd 21599998 13-b3.cpp 80 screen	检查学号错误					
14-b3 215abcd 21599998 13-b3.cpp 80	参数缺少					

4、模拟课件中 Windows 下 ping 命令的参数解析

假设 ping 命令的基本语法格式为：**ping [-l 大小] [-n 数量] [-t] IP 地址**

说明：(1) [***]表示该参数为可选项，若不带参数或参数超过范围，则使用缺省值，

★ 可选项必须以-开头，否则给出错误信息“不是以-开头的合法参数”

★ 每个可选项后可以带 1 个 int 型的额外参数，额外参数可以指定范围及默认值

★ -l 后参数的合理范围是[32..64000]，默认值为 64

★ -n 后参数的合理范围是[1..1024]，缺省值为 4

★ -t 后面不带参数，打印时，带参数为 1，不带为 0 即可

★ -l 后面的参数，如果再是-开头（含给出负整数），则给出错误信息“参数-l 没有后续参数”（-n 同样处理）

★ 出现非“-l/-n/-t”的参数，例如-x，则给出错误信息“参数-x 不存在”

★ **-t 和 -n 数量** 在实际 ping 命令中是互斥的，分析中不用管

(2) IP 地址的基本格式为点分十进制 ***.***.***.***，其中每个数字都在 0-255 之间，要求 IP 地址必须是 ping 命令的最后一项

★ IP 地址检查不正确，给出错误信息“IP 地址错误”

★ 首先检查 IP 地址，再检查其它参数

【注:】实际的 ping 操作支持 www.sohu.com 形式的 DNS 解析，作业中认为错误（IP 地址格式不正确）即可

(3) 如果参数出现重复，如 “-l 64 -t -l 200”，则后者（200）覆盖前者（64）即可

(4) 建议：本程序在 cmd 下调试比集成环境下方便

要求：(1) 在命令行下带参数执行，分析执行时所带的参数，并给出分析结果（不需要具体实现 ping）。
(2) 未带任何参数，则给出 Usage 提示
(3) 给出 14-b4-demo.exe 供参考

```
VS-Debug

D:\VS-Debug>14-b4-demo.exe
Usage: 14-b4-demo.exe [-l 大小] [-n 数量] [-t] IP地址
       ================================
        参数 附加参数 范围        默认值
       ================================
        -l   1        [32..64000] 64
        -n   1        [1..1024]   4
        -t   0        [0..1]      0
       ================================

D:\VS-Debug>14-b4-demo 192.168.80
IP地址错误

D:\VS-Debug>14-b4-demo 192.168..230
IP地址错误

D:\VS-Debug>14-b4-demo 192.168.80.230
参数检查通过
-l 参数：64
-n 参数：4
-t 参数：0
IP地址：192.168.80.230

D:\VS-Debug>14-b4-demo 192.168.80.260
IP地址错误

D:\VS-Debug>14-b4-demo -w 192.168.80.230
参数-w不存在

D:\VS-Debug>14-b4-demo -l 192.168.80.230
参数-l没有后续参数

D:\VS-Debug>14-b4-demo -l -t 192.168.80.230
参数-l没有后续参数

D:\VS-Debug>14-b4-demo -l 10 -t 192.168.80.230
参数检查通过
-l 参数：64
-n 参数：4
-t 参数：1
IP地址：192.168.80.230

D:\VS-Debug>14-b4-demo -n 70000 -l 10 -t -l 128 192.168.80.230
参数检查通过
-l 参数：128
-n 参数：4
-t 参数：1
IP地址：192.168.80.230

D:\VS-Debug>
```

参考要求(4)，Usage 的打印也要求能灵活适应，不能写死

(4) 要求可以很方便的变更参数的名称、附加参数的个数、附加参数的上下限、默认值等，应该如何设计程序的存储结构并实现？

（例：将-l 变更为-s，带 1 个附加参数，范围[128..32000]，默认 128，除了初始化外，不改动其它位置）

(5) 下表为部分组合及测试结果**（注：表中分析结果仅为示例，具体的输出信息要求与 demo 保持一致）**

命令	分析结果	l 的值	n 的值	t 的值
14-b4	Usage: 14-b4 [-l 大小] [-n 数量] [-t] IP 地址			
14-b4 www.sohu.com	IP 地址错误			
14-b4 192.168.1.256	IP 地址错误			
14-b4 .168.1.230	IP 地址错误			
14-b4 192.168..230	IP 地址错误			
14-b4 192.168.1	IP 地址错误			
14-b4 192.168.1.	IP 地址错误			
14-b4 -n	IP 地址错误			
14-b4 192.168.1.10	正确	64	4	0
14-b4 -x 192.168.1.10	参数-x 不存在			
14-b4 n 192.168.1.10	不是以-开头的合法参数			
14-b4 -l 192.168.1.10	参数-l 没有后续参数			
14-b4 -t -l 192.168.1.10	参数-l 没有后续参数			
14-b4 -l 31 192.168.1.10	正确	64	4	0
14-b4 -l 1024 192.168.1.10	正确	1024	4	0
14-b4 -l abc 192.168.1.10	正确	64	4	0
14-b4 -l 1024 -t 192.168.1.10	正确	1024	4	1
14-b4 -t -l 1024 192.168.1.10	正确	1024	4	1
14-b4 -t -n 192.168.1.10	参数-n 没有后续参数			
14-b4 -t -n 2048 192.168.1.10	正确	64	4	1
14-b4 -n -12 192.168.1.10 (-12 是负 12)	参数-n 没有后续参数（-12 被识别为-开头）			
14-b4 -l 256 -n 20 -l 512 192.168.1.10	正确	512	20	0
14-b4 -t -n 20 -l 256 192.168.1.10	正确	256	20	1
14-b4 -t -n -l 256 192.168.1.10	参数-n 没有后续参数			
14-b4 -t -n 20 -l 192.168.1.10	参数-l 没有后续参数			
14-b4 -n 20 -l 256 -t -n 10 192.168.1.10	正确	256	10	1

【编译器要求:】

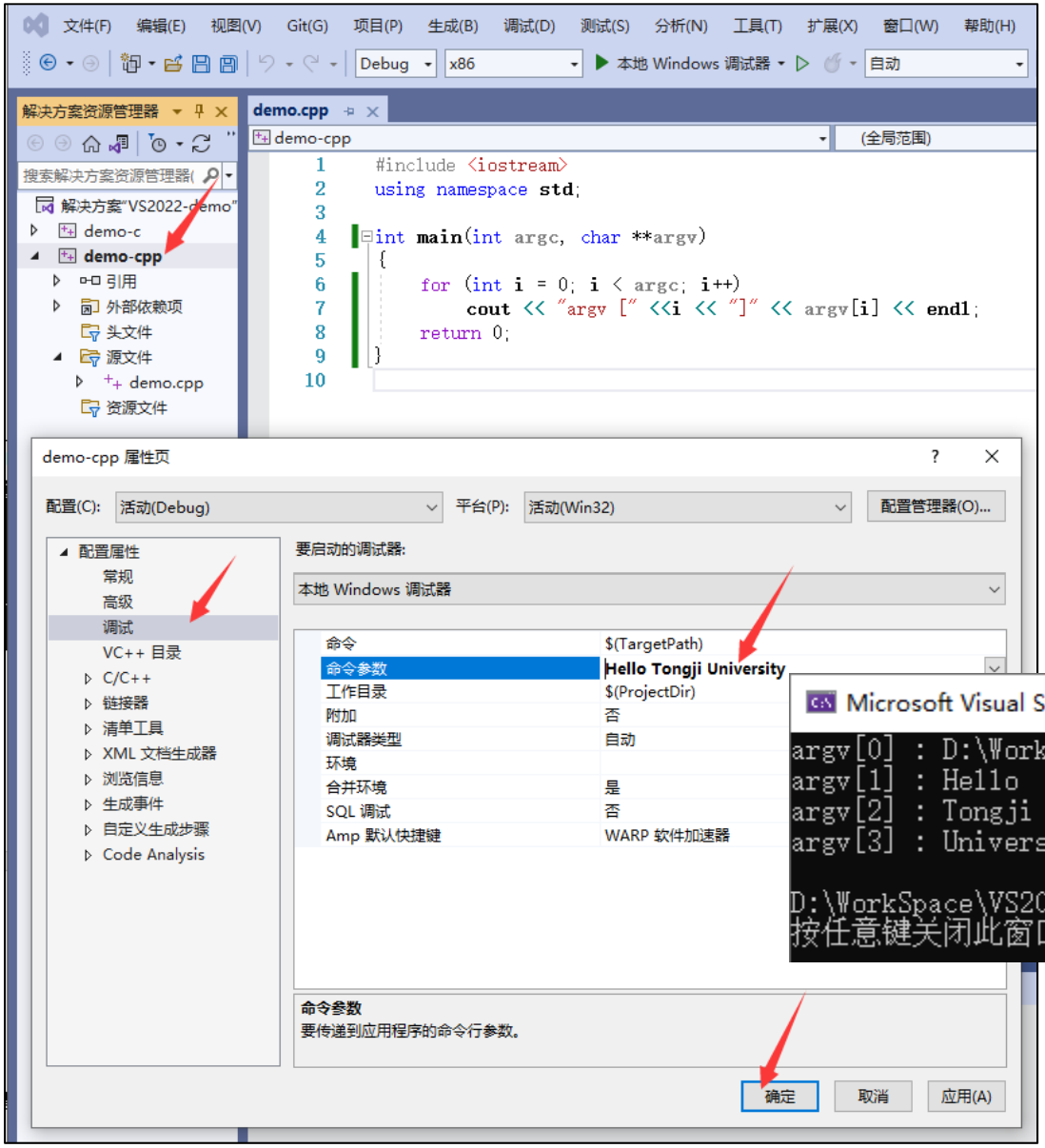
		编译器VS	编译器Dev	编译器Linux
14-b1.cpp	位运算模拟多开关	Y	Y	Y
14-b2.cpp	斗地主发牌	Y	Y	Y
14-b3.cpp	作业相似度匹配参数解析	Y	Y	Y
14-b4.cpp	模拟ping的参数解析	Y	Y	Y

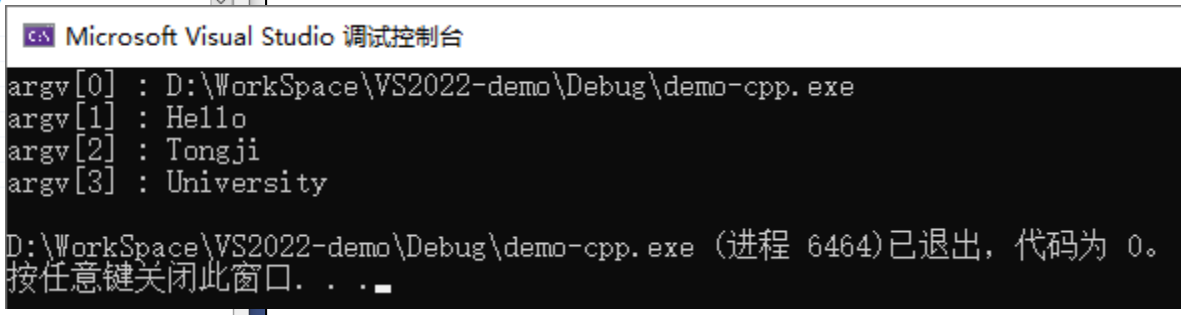
【作业要求:】

1、10 月 24 日前网上提交本次作业
2、每题所占平时成绩的具体分值见网页
3、超过截止时间提交作业则不得分

补充:

说明：如何在 VS2022 的集成环境下设置 main 函数带参数





1、 选中相应的项目
2、 鼠标右键 – 属性 – 打开属性页 – 选调试
3、 右侧命令参数中依次输入若干字符串（空格分隔）
4、 按确定
5、 如果所示，则每次运行时 argc 为 4，argv[0]~[3]打印如下
6、 可根据需要更改命令参数