**实验13 C与LUA交互之数组操作与字符串处理**

**实验目的**

（1）熟悉用C函数操作数组

（2）熟悉用C函数操作字符串

（3）掌握C函数对字符串和缓冲处理。

**实验内容**

1. 打开vs2019，创建控制台项目。创建test.lua文件，并在其中定义一个数组array = {"Hello", 1, "World", 23.2}。编写C++程序，使用lua\_rawgeti函数实现对数组array的调用。
2. 在lua文件中定义一个空数组array1 = { }，在C++中使用lua\_rawseti函数实现向数组array1中写入整数数据。

（3）在test.lua中添加如下语句。

print ("数组变换测试...")

t1 = {1,2,3,4,5}

f = function(v) --以1个数组元素为参数

return v \* 10 --返回变换后的1个值

end

print("变换前的数组为:")

for i,v in ipairs(t1) do

print(v)

end

arrayTransformation(t1,f) --由C提供

print("变换后的数组为:")

for i,v in ipairs(t1) do

print(v)

end

print("======================================")

print("准备进行平均数测试，正在初始化数组...")

s = os.time()

t2={ }

for i=1,10000000 do

t2[i] = i

end

print("初始化用时：" .. os.time()-s .."秒")

s = os.time()

print(avege(t2)) --由C提供

print("AVG函数用时："..os.time()-s .."秒")

print("======================================")

s = os.time()

print(avegeraw(t2)) --由C提供

print("AVGRAW函数用时："..os.time()-s .."秒")

print("======================================")

print("字符串分割测试...")

s = "The quick brown fox jumps over the lazy dog"

print("测试字符串为:" .. s)

t = split(s, " ") --由C提供

print("分割结果为:")

for i,v in ipairs(t) do

print(v)

end

print("======================================")

print("转换字符串为大写:")

print(upper(s)) --由C提供

print("======================================")

s2= "LUA"

print("转换字符串为小写:")

print(lower(s2)) --由C提供

（4）补充下面空缺的代码，并添加到C++程序中。

//lua数组变换

int CforLUA\_arrayTransformation(lua\_State\* L)

{

//第一个参数必须是table(t)

luaL\_checktype(L, 1, LUA\_TTABLE); //若类型不匹配，则抛出错误

//第二个参数必须是函数(f)

luaL\_checktype(L, 2, LUA\_TFUNCTION);

//获取table 的大小

int n = luaL\_len(L, 1);

for (int i = 1; i <= n; i++)

{

lua\_pushvalue(L, 2);//压入f ==> 将要调用的函数入栈

lua\_rawgeti(L, 1, i);//压入t[i] ==> 将要使用的实参入栈

lua\_call(L, 1, 1); //调用f(t[i]) ==> 函数调用

lua\_rawseti(L, 1, i); //t[i] = 结果 ==> 修改数组元素的值

}

return 0;//无返回值

}

//求lua数组平均值

static int CforLUA\_AVG(lua\_State\* L)

{

luaL\_checktype(L, 1, LUA\_TTABLE);

int n = luaL\_len(L, 1);

double sum = 0;

for (int i = 1; i <= n; i++)

{

lua\_pushinteger(L, i);

lua\_gettable(L, -2);

sum += lua\_tonumber(L, -1);

lua\_pop(L, 1);

}

lua\_pushnumber(L, sum / n);

return 1;

}

//求lua数组平均值

static int CforLUA\_AVGRAW(lua\_State\* L)

{

//参考CforLUA\_AVG函数，使用lua\_rawgeti(L,t,key) 方法实现求解lua数组的平均值

return 1;

}

//字符串分割

int CforLUA\_split(lua\_State\* L)

{

// 检查传入的2个参数的合法性

const char\* pSrc = luaL\_checkstring(L, 1); //原串

const char\* pSep = luaL\_checkstring(L, 2); //分隔符

const char\* pSepLocation = NULL; //分隔符位置

lua\_newtable(L); //在虚拟栈创建空表，用于存放结果

int i = 1; //结果表(数组)起始元素下标

/\* repeat for each separator \*/

while ((pSepLocation = strchr(pSrc, \*pSep)) != NULL) {

//strchr()是C 库函数，在参数 s 所指向的字符串中搜索第一次出现字符 \*sep的位置

lua\_pushlstring(L, pSrc, pSepLocation - pSrc); //压入子串

lua\_rawseti(L, -2, i++); //设置结果表(数组)，添加数组元素t[i]= 子串

pSrc = pSepLocation + 1; //跳过分隔符

}

lua\_pushstring(L, pSrc); //最后一个子串入栈

lua\_rawseti(L, -2, i); //添加数组元素

return 1; //推送1个返回值，即整个数组

}

//转换为大写字母

int CforLUA\_upper(lua\_State\* L)

{

size\_t len;

luaL\_Buffer buf; //①声明缓冲区变量buf

const char\* s = luaL\_checklstring(L, 1, &len);

luaL\_buffinit(L, &buf); //②初始化缓冲区变量buf

for (size\_t i = 0; i < len; i++)

//③调用luaL\_add\*系列函数向缓冲区添加字符串

luaL\_addchar(&buf, toupper((unsigned char)(s[i])));

luaL\_pushresult(&buf); //④更新缓冲区，将最终的结果字符串留在栈顶

return 1; //推送1个返回值

}

//转换为小写字母

int CforLUA\_lower(lua\_State\* L)

{

//将代码补充完成，实现所有字母转换成小写

}

int main()

{

lua\_State\* pluaState = luaL\_newstate(); /\*Creates a new Lua state\*/

luaL\_openlibs(pluaState); /\*Opens all standard Lua libraries into the given state\*/

lua\_register(pluaState, "arrayTransformation", CforLUA\_arrayTransformation);

lua\_register(pluaState, " ", CforLUA\_AVG);

lua\_register(pluaState, "avegeraw", CforLUA\_AVGRAW);

lua\_register(pluaState, "split", CforLUA\_split);

lua\_register(pluaState, "upper", CforLUA\_upper);

lua\_register(pluaState, "lower", CforLUA\_lower);

luaL\_dofile(pluaState, "test.lua");

readLuaArray(pluaState, "array");

readLuaArray(pluaState, "array0");

writeLuaArray(pluaState, "array0");

readLuaArray(pluaState, "array0");

writeLuaArray(pluaState, "array1");

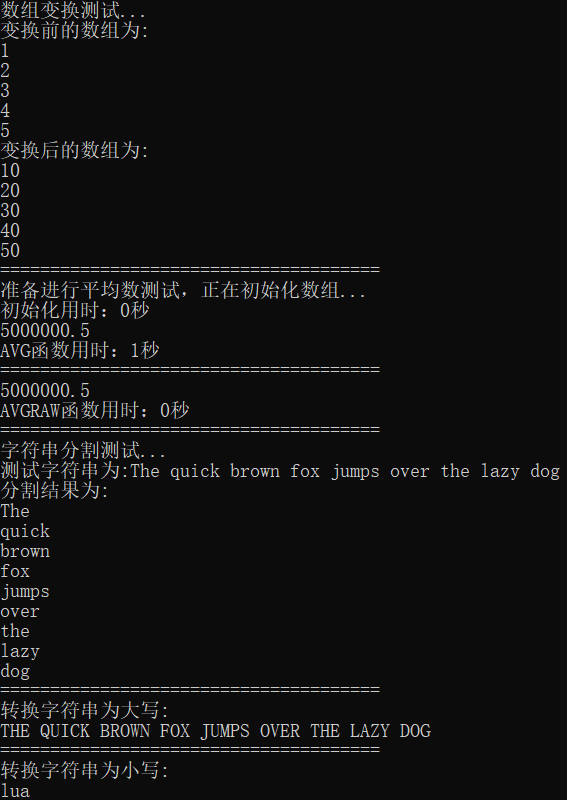
readLuaArray(pluaState, "array1");

lua\_close(pluaState);

return 0;

}

运行结果如下：



（5）在项目中用C++语言为lua脚本编写一个函数实现字符串大小写切换的功能。若lua中此功能的函数取名为 toggle ，请实现此函数并进行验证。

（6）编写一个简单的统计库(Statist).实现对一个数字型数组进行以下统计功能:[可以使用定义C模块的方法，也可以使用以上实验中方法 单独定义多个函数实现不同功能]

a) 平均数(Avg)

b) 计数(Count)

c)求和(Sum)

d) 最大值(Max)

e) 最小值(Min)

f) 方差(Varp)

g) 标准差(StdDevP)

提示：

标准差是方差开方后的结果(即方差的算术平方根)

假设这组数据的平均值是m

方差公式s^2=1/n[(x1-m)^2+(x2-m)^2+...+(xn-m)^2]

**实验要求**：

（1）独立完成；

（2）win7以上兼容系统，lua解释器，VS2010以上版本；

（3）记录并分析实验结果。

**实验报告要求**：

（1）按照学院的实验报告格式编写实验报告(注意有封面的)，不能直接使用本指导书修改；

（2）指导书截图不用复制到实验报告，实验报告要附上自己的关键代码。最主要是要有合理的实验分析和总结；

（3）以“学号+姓名+实验\*”命名word文档，提交到ftp对应位置。