

大作业报告—数据库

1. 程序功能

基础功能（均已实现）

1. 存档与读档
2. 数据导入导出
3. Count(Expression)语句
4. 分组语句
5. 排序语句

附加功能

1. **登录密码**机制，支持用户“用户名+密码”登录，且支持用户注册。实现**多用户**共享程序，并保证数据信息安全。

实现方法：

通过文件重定向(fstream)，在进入程序时不直接开始读取命令，没有账号的用户，**没有**账号的用户，先输入希望**注册**的用户名，先搜索用户名对应的日志文件（文件名中对应位置为用户名）。若**不存在**，则之后软件会提示没有此用户，询问用户是否希望注册，之后按要求设置密码，即可使用。若**存在**，则提醒用户输入密码，在 5 次以内输入正确才能进入，其中密码存在于日志文件中。**拥有**账号的用户，即可通过“用户名+密码”登录，并**自动读取**之前的**存档**

即使他人接着使用此软件，也不能对用户数据造成更改（只有“用户名+密码”**成功登录后**才**有权限修改数据文件**）；注册过的用户可随时登录，访问、查询与修改。

2. **支持扩展的数据类型**，实现了日期与时间类型

由于第一阶段同学并未对各种类型进行特化，而是完全以 string 类型存储加上另外的方式存储类型以后期进行识别，时间和日期类型也随之而选择 string 类型存储。严格按 YYYY-MM-DD 方式存储日期，并以 AA:BB:CC 形式存储时间，在进行计算时将 string 拆分成整数进行计算，包括 addtime 和 adddate 函数，考虑了闰年和各个月份的区别，保证了准确性。计算后组成并返回 string 进行显示。

同时在录入和计算过程中保证数据合法，若输入不合法会返回字段“Invalid”并提醒用户重新输入。又鉴于 string 良好的兼容性，使得可以对于已有的运算符有良好的适配性。

3. 实现绝大部分常用的**数学函数**（共计 19 个）

所有数学函数在 select 命令后实现，是属于 database 级别的命令，因为并未指明 current 是哪个。并且，依赖于 parser 函数，所有的数学函数都实现了对于大小写不敏感。绝大部分数学函数依赖已经存在的 c++ 库中函数。

4. **LIKE 语句**搜索(需要按照 Mysql 语法标准进行查询)

为某字段在对应列内容中是否存在，并且 '%' 字符规定了其出现的位置，'%s' 代表 s 在末尾，'s%' 代表 s 在起始，'%s%' 代表 s 在任何位置。以 string 存储所有数据的优势在此显现，

因为 string 自带与子串有关的函数，如 substr,length,c_str 等。

5. 用户可以**自定义**设置数据库的**显示颜色**

使用方法：在进入程序后会有选择菜单，根据颜色输入编号即可改变字体颜色。

字体背景的颜色同样是可调的，这些改变得益于 Windows.h 库。简化版的函数也存在，通过 stdio.h 库实现。

6. **UNION**(包括 UNION ALL)

鉴于第一阶段同学 parser 函数巨大的优势以及一些不便（在此处是将分号完全处理掉）使得分号不再是句子的分隔符，而以回车作为分隔符。因而在此制作的是将多个 select 语句合为一行。通过 string 下标访问 map 类型 columns 找到 expression 的位置，然后遍历，放入一个临时的 vector，同时保证如果读到 ALL，就不与已存在的重复。union 是在 database 层面实现，临时 vector 承装来自两个 table 的内容，然后针对这个 vector 处理；

鉴于 database 和 table 无继承关系，在 table 层面可以访问的私有成员在 database 无法访问，因而增加了一些可以访问到私有成员的 table 类公有函数：

```
int pri() { return Table::primary; }
int show_loca(string a) { return Table::columns[a].order; }
int cnum_f(){ return cnum; }
int rnum_f(){ return rnum; }
int primary_f(){ return primary; }
string pri_type_f(){ return pri_type; }
vector <string> record_f(){ return record; }
vector <string> col_name_f(){ return col_name; }
map<string, col_info> columns_f(){ return columns; }
```

这些基本上是在第一阶段基础上增加的所有接口，因为多表连接的需求出现在第二阶段。我们保证了原成员的封闭性不受到影响。

7. **存档功能**

实现方法：与登陆功能相结合。首先使用语句“std::stringstream* oldbuf = std::cout.rdbuf(testout.rdbuf());”进行流重载，使用 ofstream 在用户所对应的操作日志文件中（命名为 code+id+.txt），记录出去显示的函数（select 和 show）之外的所有操作，原份拷贝并且在下次用户登录后会直接阅读并重复日志中每一步操作，实现存档功能。最后使用“std::cout.rdbuf(oldbuf);”语句使流安全的回到控制台。

8. **更便捷的检查方法**

实现方法：开启#define TEST 后将跳过颜色选择，登陆等步骤，输入测试编号直接在控制台输入命令并得到 TestPoint+id 为文件名的输出文件，方便比对。我们编写了随机生成命令的代码，省去拷贝的麻烦，加上小组所编写的比对程序，在已得到标准答案的情况下更不需要人眼检查。回到原来的模式也只需将#define TEST 行去掉即可。

2. 优势分析

1. 数据库由 2 位同学合作完成，2 人完成了 3 人的工作量
2. 支持多用户使用，并且有密码保护机制，保障数据安全 (1 账户对应 1 密码对应 1 存档)
3. 扩展功能全面，并且均遵循 MySQL 标准语法
4. 尽可能不更改第一阶留有的接口和其属性，保证安全性。
5. 充分利用了第一阶段代码中的优点，如 parser 函数。并且在其基础上尽可能增加鲁棒性，在不符合程序进程时候增加提示语。
6. 尽可能少的完全复制粘贴，向 oop 思想靠拢。
7. 通过函数重载等多种方式避免重定向带来的只读空串使得程序直接退出的问题。

3. 存在的问题

1. 受编程语言限制，存储密码的文件未能进行加密处理，因此在交付用户前，可以对此进行进一步优化（比如再写一个存档文件加密程序）。
2. 由于第一阶段的 Parser 已经自动除去每行末尾的“;”，所以在 UNION 语句部分，“UNION”关键字的语法要求其前后的两个分语在同一行出现（因为第一阶段处理丢失了“，”分隔符）

4. 经验总结

1. 完成作业过程中遇到的问题：
文件重定向部分出现了输入流无法切换回控制台的问题（已解决）

经验：注意文件输入流若进行了切换，则在使用结束后，需要切换回控制台，否则会出现“读档结束后程序自动结束”的问题

总结：每次切换输入流之后注意切换回默认状态，输出流亦是如此。

5. 实现分析（基础功能）

1. 数据导出功能

“std::streambuf* oldbuf = std::cout.rdbuf(testout.rdbuf());”进行流重载，使用 ofstream 在用户所对应的所输出文件中（文件名从 select 命令行中提取），将 select 所输出的内容导出到 outfile 中。最后使用“std::cout.rdbuf(oldbuf);”语句使流安全的回到控制台。

2. 数据导入功能

在 table 层面新增的 Table::load 函数。首先检验所导入的文件是否存在，若否则输出提示。读文件时使用 eof()函数，读到结尾则跳出。首先使用 vector<string>将所有内容原封不动搬入，然后检查文件中各列的顺序和 table 中各列顺序如何对应，产生联系后根据对应关系将 vector 内容推入表中。为保证以一维数组存储的表的稳定性，在未读到内容的时候推入 NULL 防止 vector 爆掉。

3. Count(expression)功能

通过 columns 找到 expression 的位置，然后遍历，放入一个临时的 vector，同时保证不与已存在的重复，并且不录入 null。

6. 测试截图

1. Like 函数

小规模人工生成（并且数据库支持 DATE 与 TIME 类型：表头第一个）

```

**      2      GREEN      **
**      3      BLUE      **
**      4      YELLOW    **
**      5      PURPLE     **
**      6      CYAN       **
**      7      DEFAULT    **
*****
Enter the corresponding ID for the color :
6
Please enter user's name:lza
Mysql found. Enter password:123
CREATE DATABASE OOP;
USE OOP;
CREATE TABLE poi(lkj DATE, mnjb INT, bvc INT, dfj CHAR, PRIMARY KEY(lkj));
INSERT INTO poi(lkj, mnjb, bvc, dfj) VALUES (2028-01-02, 2, 3, "absdsf");
INSERT INTO poi(lkj, mnjb, bvc, dfj) VALUES (2038-09-02, 3, 4, "qfqwfzsds");
INSERT INTO poi(lkj, mnjb, bvc, dfj) VALUES (2048-08-02, 4, 5, "zvbzbbeef");
INSERT INTO poi(lkj, mnjb, bvc, dfj) VALUES (2058-07-02, 5, 6, "jrtjdrksser");
INSERT INTO poi(lkj, mnjb, bvc, dfj) VALUES (2068-06-02, 6, 7, "sjsexfbxdf");
SELECT * from poi;
lkj      mnjb      bvc      dfj
2028-01-02      2      3      absdsf
2038-09-02      3      4      qfqwfzsds
2048-08-02      4      5      zvbzbbeef
2058-07-02      5      6      jrtjdrksser
2068-06-02      6      7      sjsexfbxdf
SELECT * from poi WHERE dfj LIKE '%ser';
lkj      mnjb      bvc      dfj
2058-07-02      5      6      jrtjdrksser
```

利用测试程序随机生成较大规模（以下为 100 条样例数据的测试截图）

类型 1 以 x 结尾

```
tcu      xcb      bik      sdj
tfp      bwa      pxg      myv
thv      vlv      lnf      vtk
tij      wtd      ruy      yfq
tlt      wue      vxo      udx
tny      efq      bjg      xsm
tsc      eqc      bdv      brt
vml      fff      odb      pfw
vpj      jkk      udf      ypm
vqb      bld      tfp      ogj
wig      tvn      nwm      anu
wqo      jcl      thh      jup
xig      goi      nim      hct
xit      ayp      ibw      cjs
xji      ipf      wja      gjg
xlc      hte      psq      trx
xyq      hrf      hzd      rkk
yay      sek      mxu      pcy
yih      zdc      ibt      sua
ywo      zxr      ikm      uhl
yza      dij      jxa      chc
zgm      kbc      rix      but
zgt      dti      rge      afc
zib      bev      gcx      qqc
zqd      smj      fky      ssl
SELECT * from poi WHERE dfj LIKE '%x';
lkj      mnv      bvc      dfj
tlt      wue      vxo      udx
xlc      hte      psq      trx
```

类型 2 以 x 开头

tny	efq	bjg	xsm
tsc	eqc	bdv	brt
vml	fff	odb	pfw
vpj	jkk	udf	ypm
vqb	bld	tfp	ogj
wig	tvn	nwm	anu
wqo	jcl	thh	jup
xig	goi	nim	het
xit	ayp	ibw	cjs
xji	ipf	wja	gjj
xlk	hte	psq	trx
xyq	hrf	hzd	rkk
yay	sek	mxu	pcy
yih	zdc	ibt	sua
ywo	zxr	ikm	uhl
yya	dij	jxa	chc
zgm	kbc	rix	but
zgt	dti	rge	afc
zib	bev	gcx	qqc
zqd	smj	fky	ssl

SELECT * from poi WHERE dfj LIKE 'x%';

lkj	mnb	bvc	dfj
clr	mip	kwp	xoq
cny	zhc	yov	xwv
hah	ddv	qfc	xac
htt	xre	wqx	xzb
psj	fbr	cyc	xfr
qjw	cti	wjc	xsx
tny	efq	bjg	xsm

类型 3 中间出现 x 字段(% x %)

子样例 1 (单字符)

vpj	jkk	udf	ypm
vqb	bld	tfp	ogj
wig	tvn	nwm	anu
wqo	jcl	thh	jup
xig	goi	nim	het
xit	ayp	ibw	cjs
xji	ipf	wja	gjj
xlk	hte	psq	trx
xyq	hrf	hzd	rkk
yay	sek	mxu	pcy
yih	zdc	ibt	sua
ywo	zxr	ikm	uhl
yya	dij	jxa	chc
zgm	kbc	rix	but
zgt	dti	rge	afc
zib	bev	gcx	qqc
zqd	smj	fky	ssl

SELECT * from poi WHERE dfj LIKE '%x%';

lkj	mnb	bvc	dfj
clr	mip	kwp	xoq
cny	zhc	yov	xwv
hah	ddv	qfc	xac
htt	xre	wqx	xzb
jnp	cvy	zsa	pxv
psj	fbr	cyc	xfr
qjw	cti	wjc	xsx
tlr	wue	vxo	udx
tny	efq	bjg	xsm
xlk	hte	psq	trx

子样例 2 (多字符)

```

ttjks xitxa ynzdt jywgn
uexxp uisrg umovn endon
ugfcs pbdkd sevcj usmhp
ukaxa orxdt cddfb btjwp
urqbl owmmw iuugh aqvxd
uxnlb flhmj kwydu cjtut
vfohq ksrou qfuom hevjr
vtkxc nrpet znhvu oxcud
waqjj ymzzn ggwdm cunrs
wjwlc xtxel etpkc iubsc
wmkkd tuprb usoam stmox
wrzsw giwkb ulfli cxwyl
wvvhk qbbdb hltok zizyi
wvvnv yexwj yrzxm cyumz
wzcuo upgnk ehxda vjbws
xiezb piwsh ylowa tvbhk
xtbfz vuxum nmfjk zwugo
xvtsv evmbh kvwmb apngg
xybdd uwphg xwhno nwcx
yhhws cnrxm gkqsg cpzju
yibng dnseu peowf pmkcq
yrvzj jpass iumbs tmnsj
ysrve wjcsa jwqdz cwtqy
zhtgj ejwas vqddw ebzsg
zpkad zrhbt agaem osajm
zuslg bzgvi omvmv fnzwb
SELECT * from poi WHERE dfj LIKE '%jg%';
lkj mnb bvc dfj
coeqy qthvn jckyr jjgjr

```

2. 账号注册

(类似于 windows 下的多个 administrators 模式，可实现多个用户共享软件)

注册用户

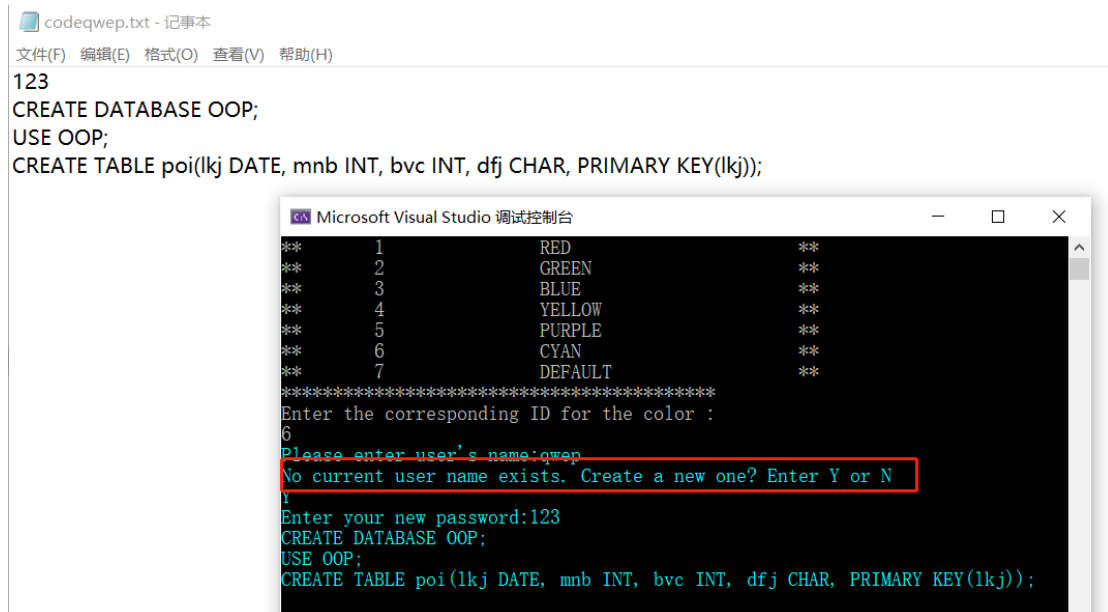
示例 1. 注册一个“new”用户名

```

Hello, Please select the textcolor you like :
*****
**      1          RED          **
**      2          GREEN        **
**      3          BLUE         **
**      4          YELLOW       **
**      5          PURPLE       **
**      6          CYAN         **
**      7          DEFAULT      **
*****
Enter the corresponding ID for the color :
5
Please enter user's name:new
No current user name exists. Create a new one? Enter Y or N
Y
Enter your new password:114514

```

示例 2. 注册一个“qwep”用户名（注意文档名称）



用户登录

用户登录账号使用，对其余的数据与记录不造成影响（下图为用户名为“user_default”再次登录，读取其之前的存档）

```
Hello, Please select the textcolor you like :
*****
**      1      RED      **
**      2      GREEN    **
**      3      BLUE     **
**      4      YELLOW   **
**      5      PURPLE   **
**      6      CYAN     **
**      7      DEFAULT  **
*****
Enter the corresponding ID for the color :
6
Please enter user's name:user_default
Mysql found. Enter password:123
Wrong code. You have 4 times left. Enter again:123
Wrong code. You have 3 times left. Enter again:123456
SHOW DATABASES;
Database
oop
```

3. 密码登录

(安全密码机制，连续输入密码错误 5 次，程序就会强制关闭输入保证各用户数据安全)

以下为登陆失败样例

```
Hello, Please select the textcolor you like :
*****
**      1          RED          **
**      2          GREEN        **
**      3          BLUE         **
**      4          YELLOW       **
**      5          PURPLE       **
**      6          CYAN         **
**      7          DEFAULT      **
*****
Enter the corresponding ID for the color :
3
Please enter user's name:lza
Mysql found. Enter password:aef
Wrong code. You have 4 times left. Enter again:reg
Wrong code. You have 3 times left. Enter again:r
Wrong code. You have 2 times left. Enter again:v
Wrong code. You have 1 times left. Enter again:th
Wrong code. Quitting.....
C:\Users\lizia\Desktop\BigProject\x64\Debug\BigProject.exe (进程 17916) 已退出，返回代码为: 0。
若要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口...
```

登录成功即可使用与读取以前的数据存档

```
Hello, Please select the textcolor you like :
*****
**      1          RED          **
**      2          GREEN        **
**      3          BLUE         **
**      4          YELLOW       **
**      5          PURPLE       **
**      6          CYAN         **
**      7          DEFAULT      **
*****
Enter the corresponding ID for the color :
6
Please enter user's name:lza
Mysql found. Enter password:123
```

4. 自动读档

用户登录成功后即可自动读取之前的存档（用户注册过，则会提示“MySQL found”，并输入密码，密码正确则会自动读档）

```

Hello, Please select the textcolor you like :
*****
**      1      RED      **
**      2      GREEN    **
**      3      BLUE     **
**      4      YELLOW   **
**      5      PURPLE    **
**      6      CYAN     **
**      7      DEFAULT  **
*****
Enter the corresponding ID for the color :
4
Please enter user's name: USER
Mysql found. Enter password: 123
USE OOP;
SELECT * FROM poi;
lkj   mnk   bvc   dfj
aad   hju   ycp   awh
acv   jbu   gte   mnt
ahg   hbc   qts   eft
bdb   wyg   mby   fwn
bfb   kra   lzb   ynt
bpb   wtk   mcp   oog
bun   qax   lol   orh
bvy   jhr   bwq   waw
ccx   abz   jcm   ely
cdy   owz   zon   jrj
clr   mip   kwp   xoq
cny   zhc   yov   xwv
ctx   itx   nai   rww

```

密码需要用户正确输入才会读档

5. UNION

```

Please enter user's name:lza
Mysql found. Enter password:123
CREATE DATABASE OOP;
USE OOP;
CREATE TABLE poi(lkj DATE, mnk INT, bvc INT, dfj CHAR, PRIMARY KEY(lkj));
INSERT INTO poi(lkj, mnk, bvc, dfj) VALUES (2028-01-02, 2, 3, "a");
INSERT INTO poi(lkj, mnk, bvc, dfj) VALUES (2018-01-02, 3, 4, "a");
INSERT INTO poi(lkj, mnk, bvc, dfj) VALUES (2018-01-04, 4, 5, "a");
INSERT INTO poi(lkj, mnk, bvc, dfj) VALUES (2018-01-03, 5, 6, "a");
INSERT INTO poi(lkj, mnk, bvc, dfj) VALUES (2018-05-02, 6, 7, "a");
CREATE TABLE poq(lkj DATE, mnk INT, bvc INT, dfj CHAR, PRIMARY KEY(lkj));
INSERT INTO poq(lkj, mnk, bvc, dfj) VALUES (2038-01-02, 2, 3, "a");
INSERT INTO poq(lkj, mnk, bvc, dfj) VALUES (2048-01-02, 3, 4, "a");
INSERT INTO poq(lkj, mnk, bvc, dfj) VALUES (2058-01-04, 4, 5, "a");
INSERT INTO poq(lkj, mnk, bvc, dfj) VALUES (2068-01-03, 5, 6, "a");
INSERT INTO poq(lkj, mnk, bvc, dfj) VALUES (2078-05-02, 6, 7, "a");
SELECT lkj FROM poi UNION ALL SELECT lkj FROM poq ORDER BY lkj;
lkj
2018-01-02
2018-01-03
2018-01-04
2018-05-02
2028-01-02
2038-01-02
2048-01-02
2058-01-04
2068-01-03
2078-05-02

```

```

CREATE DATABASE OOP;
USE OOP;
CREATE TABLE poi(1kj DATE, mnb INT, bvc INT, dfj CHAR, PRIMARY KEY(1kj));
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (2028-01-02, 2, 3, "a");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (2028-01-02, 3, 4, "a");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (2028-01-02, 4, 5, "a");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (2028-01-02, 5, 6, "a");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (2028-01-02, 6, 7, "a");
CREATE TABLE poq(1kj DATE, mnb INT, bvc INT, dfj CHAR, PRIMARY KEY(1kj));
INSERT INTO poq(1kj, mnb, bvc, dfj) VALUES (2038-01-02, 2, 3, "a");
INSERT INTO poq(1kj, mnb, bvc, dfj) VALUES (2028-01-02, 3, 4, "a");
INSERT INTO poq(1kj, mnb, bvc, dfj) VALUES (2058-01-04, 4, 5, "a");
INSERT INTO poq(1kj, mnb, bvc, dfj) VALUES (2028-01-02, 5, 6, "a");
INSERT INTO poq(1kj, mnb, bvc, dfj) VALUES (2078-05-02, 6, 7, "a");
SELECT 1kj FROM poi;
1kj
2028-01-02
SELECT 1kj FROM poq;
1kj
2028-01-02
2038-01-02
2058-01-04
2078-05-02
SELECT 1kj FROM poi UNION SELECT 1kj FROM poq ORDER BY 1kj;
1kj
2028-01-02
2038-01-02
2058-01-04
2078-05-02

```

6. 支持扩展的类型

(DATE & TIME)

```

Hello, Please select the textcolor you like :
*****
**      1      RED      **
**      2      GREEN    **
**      3      BLUE     **
**      4      YELLOW   **
**      5      PURPLE   **
**      6      CYAN     **
**      7      DEFAULT  **
*****
Enter the corresponding ID for the color :
6
Please enter user's name:lza
Mysql found. Enter password:123
SELECT ADDTIME('11:11:11', 5)
11:11:16
SELECT ADDDATE("2019-06-21", INTERVAL 11 DAY);
2019-7-2

```

7. 数学函数

(共计 19 个, 请注意这是 TEST 模式)

TestPoint1 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1 C:\Users\lizia\Desktop\BigProject\x64\Debug\BigProject.exe
1.31812 Enter Test point #1
0.304693 select abs(-1);
1.19029 select acos(0.25);
2 select asin(0.3);
0.267499 select atan(2.5);
-7.01525 select ceiling(1.5);
286.479 select cos(1.3);
2 select cot(3.0);
20.0855 select degrees(5.0);
1 select 10 DIV 5;
0.693147 select EXP(3);
3.14159 select FLOOR(1.43);
16 select LN(2);
1 select SPACE(5);
-0.756802 select PI();
4 select POWER(2,4);
7.24461 select ROUND(1.234);
select SIN(4);
select SQRT(16);
select tan(14);
Enter Test point #
```

8. 数据生成程序

(生成大规模数据，测试数据库时使用)

```
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (jstir, zwmxo, nruew, qrlrf, ghyut, "lcnw");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (pfftx, imdem, uaxim, rqrqv, psfja, "alygi");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (dgguk, ryulq, birlw, ajodr, oikwi, "acyba");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (swvwm, ixivp, godpy, jlkrr, gpagk, "sdbkv");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (badyd, vxmxy, wtyyt, qhbxn, gznjg, "njpna");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (diats, ssecf, irflo, sllmf, cnsq, "yhrzq");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (akaeb, prabf, epdzp, wkorl, isuti, "slttr");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (tquht, mahrs, mqzwx, ychzi, mswyb, "wjbcv");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (dwevl, tqjmt, efoht, bobgr, gqywh, "sfhjn");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (qtigz, kemnc, gjakg, eomdg, alryw, "shzmd");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (biuju, cwcpf, npzzu, netle, vyegr, "mozul");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (apyah, onylm, zeudl, yvvzq, zisbu, "azjix");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (nryou, cwqak, nlfav, lyqeq, qowja, "yvujr");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (rorjn, rmqli, ggiyq, skgja, mulrc, "ootog");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (imltk, kvxad, wkcam, icsfj, jmtgk, "tnfvn");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (vipec, ihqya, vipfh, uiarp, lhche, "rouxs");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (sgura, bzubo, bbefr, wlaum, zwkrz, "zvdge");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (barxl, sjlhc, stgxl, ebiml, zngit, "bnucz");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (nxyji, mkxng, mfjnv, jreji, pjbyr, "qsekj");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (xlspg, cxgoi, hwywr, chmxu, tgdli, "yqata");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (pdltf, wzket, kqvrt, ustym, nigti, "ktptz");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (pddig, kchce, fvsbs, ycmpr, ucxhi, "phaob");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (yryyo, likua, eueac, ytlqm, beoie, "xdkpj");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (ncxia, kiaan, ipunz, kvvxx, uzuxw, "ltyzx");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (iwrvb, gsjch, nlmuw, dgwnw, yzhaa, "qffbo");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (fvdab, cszuf, falzt, telrh, edeeu, "ecznh");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (nvakt, rggbl, yzste, xeeaz, kxfam, "fyyxk");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (cwvq, seqkv, tkdta, mbyfd, wgnpq, "sveub");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (wwhjb, gjwex, pygcz, adteg, bropp, "tzpcy");
INSERT INTO poi(1kj, mnb, bvc, dfj) VALUES (pkmr, dclzq, ugpyz, xfgvu, ngusm, "zinto");
```

9. 测试程序

数据输出：为便于助教测试，我们的测试程序支持连续测试 -- 先提示用户输入测试号 #x，之后根据用户输入的测试号来动态确定输出文件名，每次测试会输出一个文件 "MyPointx.txt"

数据比对：比对程序，帮助检查我们程序的输出是否与标准答案有区别

如果比对的两个文件出现不一致，会提示错误位置与正确率：

```
Please enter the TestPoint Id : 1
TestPoint # 1 succeed!
MY Output:          werwr
Standard Output:    werwr

Difference found! TestPoint # 2 failed
Difference found! TestPoint # 3 failed

TestPoint # 4 succeed!
MY Output:          freg
Standard Output:    freg

TestPoint # 5 succeed!
MY Output:          freg
Standard Output:    freg

Test Report:
*****
**      Total Test Times 5      **
**      Succeed Times 3      **
**      Total Test Times 2      **
**      Correct Rate 60%      **
*****
```

下面是我们程序的输出与标准答案的比对（这是一个 5 条数据的比对截图，正确率为 100%）

```
Please enter the TestPoint Id : 1

TestPoint # 1 succeed!
MY Output:      werwr
Standard Output: werwr

TestPoint # 2 succeed!
MY Output:      qwegawe
Standard Output: qwegawe

TestPoint # 3 succeed!
MY Output:      agxbfbd
Standard Output: agxbfbd

TestPoint # 4 succeed!
MY Output:      freg
Standard Output: freg

TestPoint # 5 succeed!
MY Output:      freg
Standard Output: freg

Test Report:
*****
**      Total Test Times 5      **
**      Succeed Times    5      **
**      Total Test Times 0      **
**      Correct Rate     100%    **
*****
```

```
Hello, Please select the textcolor you like :
*****
**      1      RED      **
**      2      GREEN   **
**      3      BLUE    **
**      4      YELLOW  **
**      5      PURPLE   **
**      6      CYAN     **
**      7      DEFAULT  **
*****
Enter the corresponding ID for the color :
3
Please enter user's name:lza
Mysql found. Enter password:aef
Wrong code. You have 4 times left. Enter again:reg
Wrong code. You have 3 times left. Enter again:r
Wrong code. You have 2 times left. Enter again:v
Wrong code. You have 1 times left. Enter again:th
Wrong code. Quitting.....
```

```
Hello, Please select the textcolor you like :
*****
**      1      RED      **
**      2      GREEN    **
**      3      BLUE     **
**      4      YELLOW   **
**      5      PURPLE    **
**      6      CYAN     **
**      7      DEFAULT   **
*****
Enter the corresponding ID for the color :
5
Please enter user's name:new
No current user name exists. Create a new one? Enter Y or N
Y
Enter your new password:114514
```