

数据结构实验报告

1. 实验序号与名称

实验 4.2.1：文本串的加密。

2. 学号，姓名，专业，实验时间

学号：2016141223037 姓名：宋运翔 专业：计算金融（方向） 实验时间：第 12~15 周。

3. 实验内容与目标

一个文本串可以用事先给定字母映射表进行加密，字母映射表如下所示：

*abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
NgzQTCobmUHelkPdAwxfYIvrsJGnZqtcOBMuhELKpaDWXFyiVRjS*

未被映射的字符不加以改变，例如，字符串“e*ncrypt”被加密成“T*kzwsdf”，写一个程序要求采用菜单方式实现相应功能，其选项及功能要求如下：

- 1.加密——将已经输入的文本串进行加密后输出。
- 2.解密——将已经输入的已加密的文本解密后输出。
- 3.退出——退出运行。

4. 实验工具

Microsoft Visual Studio 2013

5.实验分析

5.1 基础算法分析

在该实验中定义了加密类 ENCRYPT, ENCRYPT 类的成员及功能如表 5.1 所示。

表 5.1ENCRYPT 类的成员及功能

类型	成员	功能
数据成员	String letters	大小写字母表
	String map	字母映射表
方法	String Encode(const String &str)	返回加密后的文本串
	String Decode(const String &str)	返回解密后的文本串

其中 Encode()用于加密，具体方法是对文本串中的每个字符，通过查找字母表 letters，确定在字母表中的位置，然后通过字母映射表得到字母的映射。

而 Decode()用于解密，具体方法是对加密文本串中的每个字符，通过查找字母映射表 map，确定在字母映射表中的位置，然后通过字母表得到加密前的字母。

5.2 成员函数代码

5.2.1 Encode(const String &str)

LinkedList<char>MO;//临时线性链表，用于存储加密后的文件串

```
for (int i = 0; i < str.Length(); i++)
{
    int pos;
    for ( pos = 0; pos < 52; pos++)
    {
        if (str[i] == letters[pos])
        {
            MO.Insert(MO.Length() + 1, map[pos]);
            break;
        }
    }
    if (pos == 52)
    {
        MO.Insert(MO.Length() + 1, str[i]);
    }
}
String result(MO);//生成加密文本串并返回
return result;
```

5.2.2 Decode(const String &str)

LinkedList<char>MO;//临时线性链表，用于存储加密后的文件串

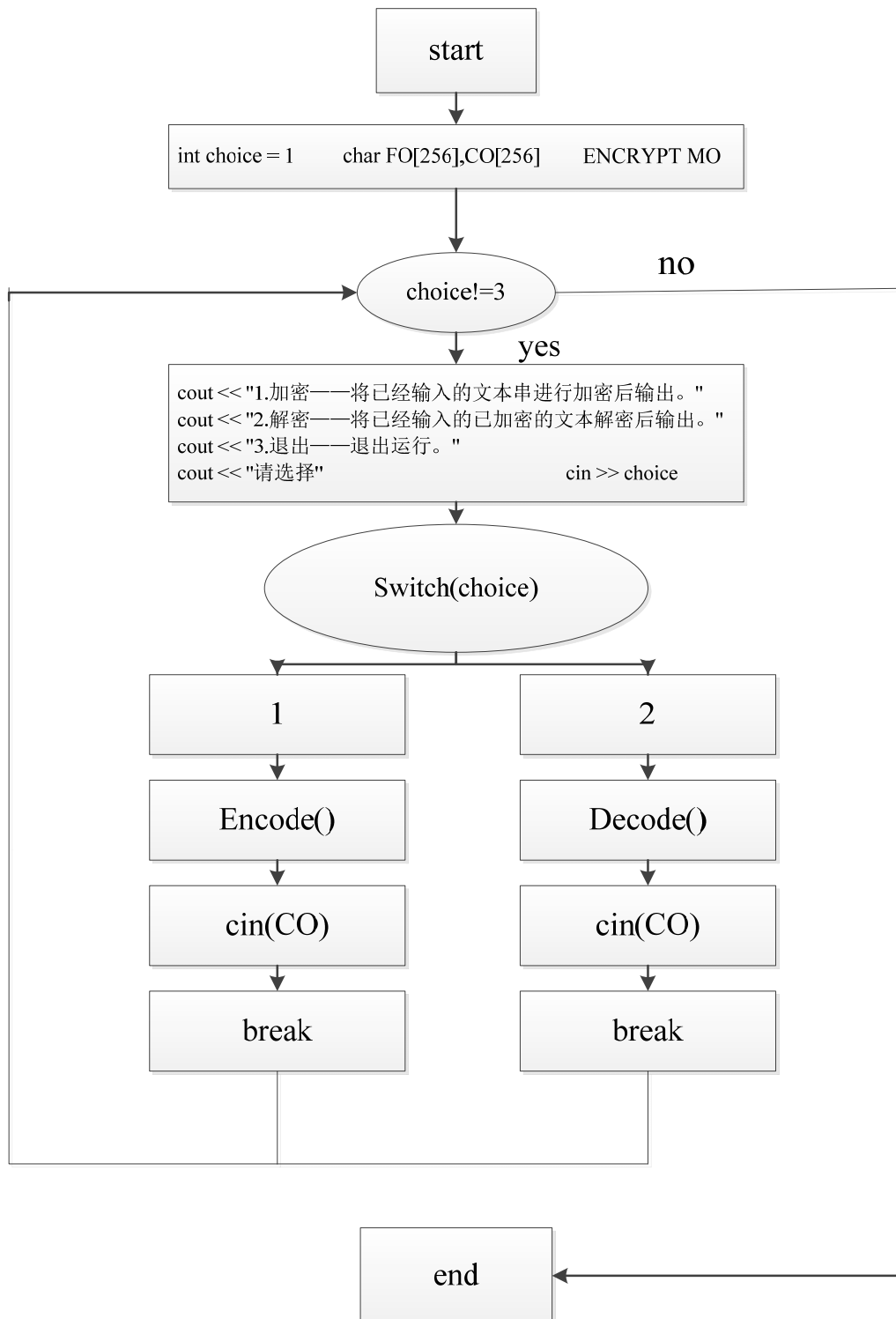
```
for (int i = 0; i < str.Length(); i++)
{
    int pos;
    for (pos = 0; pos < 52; pos++)
    {
        if (str[i] == map[pos])
        {
            MO.Insert(MO.Length() + 1, letters[pos]);
            break;
        }
    }
    if (pos == 52)
    {
```

```

        MO.Insert(MO.Length() + 1, str[i]);
    }
}
String result(MO);
return result;

```

5.3 主函数流程图



6. 实验步骤

1. 建立工程 encrypt。
2. 将软件包中 utility.h 复制到 encrypt 文件夹中，并将 utility.h 加入到工程中。
3. 将串需要的的头文件 string.h, node.h 和 lk_list.h 复制到 encrypt 文件夹中，并将 string.h, node.h 和 lk_list.h 加入到工程中。
4. 建立头文件 encrypt.h, 声明并实现加密类 ENCRYPT, 建立源程序文件 main.cpp, 实现 main() 函数。

7. 测试与结论

主界面如下：

```
1.加密——将已经输入的文本串进行加密后输出。
2.解密——将已经输入的已加密的文本解密后输出。
3.退出——退出运行。
请选择
```

选择第一项功能，并输入测试串，输出如下：

```
1.加密——将已经输入的文本串进行加密后输出。
2.解密——将已经输入的已加密的文本解密后输出。
3.退出——退出运行。
请选择1

请输入文本串：abc()efg
加密串：Ngz()TCo

1.加密——将已经输入的文本串进行加密后输出。
2.解密——将已经输入的已加密的文本解密后输出。
3.退出——退出运行。
请选择
```

选择第一项功能，并输入测试串，输出如下：

```
1.加密——将已经输入的文本串进行加密后输出。
2.解密——将已经输入的已加密的文本解密后输出。
3.退出——退出运行。
请选择2

请输入加密串Ngz()TCo
解密串：abc()efg

1.加密——将已经输入的文本串进行加密后输出。
2.解密——将已经输入的已加密的文本解密后输出。
3.退出——退出运行。
请选择
```

对照字母表与字母映射表可知两次随机输入均正确，本程序满足实验目标与要求。

8. 思考与感悟

本程序存在着不足，在选择功能输入时，若输入不是数字则会使程序出现死循环。本程序通过查找字母表或字母映射表来进行加密或者解密，效率较低，还可以通过定义字母映射表和逆字母映射表，可以直接得到字符的编码或者解码字符；同时该程序还能从对文本串加密与解密推广为对文件的加密与解密。

在本次的实验学习中，开始时觉得它很难，是需用花费了大量的时间编写出来的。现在我真正的明白了串这种结构的应用，体会到每个程序都有一些共同点，通用的结构，相似的格式。只要努力去学习，就会灵活的去应用它。