同济大学 实验报告

装

订

线

二维码生成器

---C++程序设计实验报告

姓名：童佳燕

学号：1551445

班级：计算机科学与技术1班

指导教师：沈坚

完成日期：2017年5月23日

1. 实验题目：二维码生成器

1.1 实验要求

（1）从键盘输入要生成的内容，生成二维码

（2）支持的内容至少要包括文本（含中英文，汉字，标点和特殊符号）和网址两种

（3）文本内容/网址长度不超过100字节

（4）允许使用网上现成的函数实现部分要求

装

1.2 程序设计要求

（1）独立功能使用独立的函数实现，严格按照格式编写，强调注释以及函数，变量的命名可视化。

订

（2）不允许使用goto语句和string类，以及全局变量

（3）涉及申请动态空间，要求自行释放

线

2. 整体设计思路

（1）确定二维码版本范围以及纠错级别

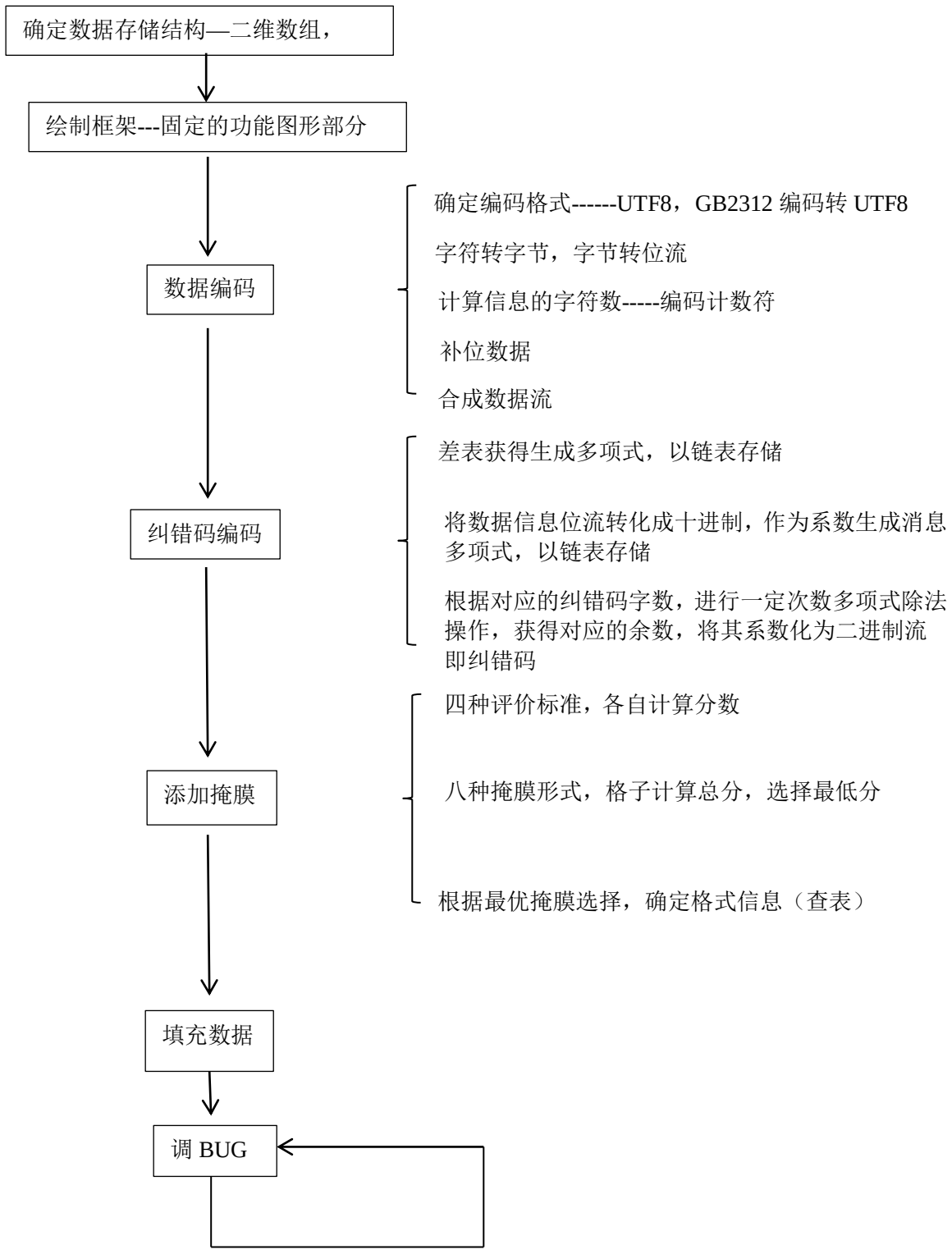
二维码根据纠错级别以及存储的信息量大的不同分为41个版本，程序设计前，需要先确定版本范围，根据最大信息量不超过100字节以及考虑程序设计的难度，最终确定的是版本1-5，纠错级别为L。

（2）具体框架设计

因为二维码的基本组成分为功能图形和编码区域，根据版本的不同，设想是：先将功能图形，也就是基本固定不变的框架确定，之后在向空白区域填充数据，包括编

码的格式信息，编码数据以及纠错码等。最后，在覆盖掩膜使黑白块均匀分布。

具体框架如下图：



3.主要功能的实现

（1）数据编码

考虑到中文字符/字母/数字/特殊符号等的需求，以及编程的方便，选用了UTF8编码。而系统默认的编码模式是GB2312，所以首先需要将GB2312转换成UTF8编码格式，采用的是网友们的智慧~ 在转换编码之后，需要注意的是汉字在GB2312中是两个字节，而在UTF8中是三个字节（大多数）。接下来就按照每个字节化成八位二进制流，再在前面接上编码格式以及字符计数编码，在后面接上补位使数据编码流长度为指定长度，就形成了数据编码二进制流。

（2）多项式长除法/纠错码

纠错码的生成是通过 由数据编码位流按每八位转换成十进制后，作为多项式系数，形成消息多项式，与生成多项式进行长除法操作，余下的余项的系数转换成二进制流并拼接形成的。具体的操作学习网友的博客进行一步一步实现，需要注意把控的是余项的个数问题与纠错码字数的控制。

（3）选择最优掩膜

掩膜总共有八种供选择，每个掩码模式使用公式来确定是否改变当前位的颜色。将当前位的坐标放入公式中，如果结果为0，则使用该坐标处的相反位。而选择对于当前二维码最优的掩膜模式建立在四种评价标准上。编程过程中，对于四种评价标准分别编写程序，返回罚分值。由于覆盖掩膜的代码较简单，就写在了一个函数中，新建用于存放编码信息的二维数组的“替身”，边覆盖边评估。返回最优掩膜对应的格式信息以及掩膜代号。应用于“真身”。

4.调试过程碰到的问题

4.1 无法扫描出结果

（1）数据字符长度符bug ---理解问题

不同的资料介绍的原理有些出入，有些资料称，字符长度符是指，比如“us 我们”，有些资料称字符数是4，而有些资料则称，字符数为6.在编码过程中，一开始是按照4来理解的，由于扫码失败，才考虑另外一种情况。

（2）掩码部分，~ VS ！

在掩码部分遭遇了一个相当尴尬的情况，由于掩码具体实现是，对于一定的公式，将元素坐标I,j代入，如果结果为0，则对该元素取反，也就是0->1,1->0.一开始很自然的使用了取反符号“~”,发现加了掩膜之后，反而相同元素堆积的更加密集了。试着去找原因，首先想到的是，我是不是选了评价分数最高的？不是。选完之后 没能正常赋值？可以。后来甚至去怀疑了对规则的理解是不是又出错了。 这个bug 其实是很容易错过的。

事实上，对于int值来说，1的二进制0x00 0x00 0x00 0x01 取反之后，则为0Xff,0xff,0xff,0xfe,显然不是我想要的0. 最终，将~改成！后，掩码部分正常显示。

(3)纠错码部分---尾零问题

很惭愧的说,这次的作业,遇到的若干几乎无从下手的bug,都是尾零惹的祸。比如在数据流和纠错码流的连接时，发现中间多了几个乱码，当时第一反应是自己长度没有控制好，但事实上并不是；再比如纠错码的数组屁股后面没有加尾零，关键是，测试时出来的并不是乱码!!而是，四个空格键！强行加了*才发现这四个隐藏的乱码。

4.2 仅1,2版本可以用，以上版本都无法扫描

尚未解决

4.3 部分中文扫不出来，比如“傻”字

5.经验总结&经验教训

5.1 拆分大程序为小程序

这次大作业与之前有一个不同之处就是，思路正确与否，需要最终结果出来之后才知分晓，也就是说，很难做到边写边验证的效果。所以，这次与上一次作业一样，选择开一个拆分问题的，写函数，验证函数的项目~将验证正确的程序加入到大程序中。

同时，面对大程序要有备份的意识，没准下一次改动就把程序改坏了，还回不去就太糟心了。

5.2 提升文字理解能力以及耐心

仔细回想这次大作业的经历，四周时间，前两周属于原理充电期（两周抽了三天大致浏览了一下），大致准备构思大作业。但是在版本的选择以及编码方式的选择上，一开始，真难~

换种说法，我觉得这次大作业主要在考察我查资料，从0开始去理解一个对象的原理等的学习能力，理解能力。而真正在编程的时间，差不多2天，二维码就基本成型了（忽略若干不完善的bug，仅指能扫出东西）。

5.3 面向对象编程初涉水

这次心血来潮，尝试使用类来进行编程，初衷一是为了让自己对类更加熟悉，二是为了尝试“分工”编程的模式，让自己的代码简洁明了。

但事实上，很遗憾的是这次的类，在数据成员和函数成员的设置上有些凌乱。有些可以通过一个简单的公式从一个数据成员得到另外一个数据成员，这样会让我

觉得很冗余~ 而对于成员函数来说，我甚至都怀疑，只要一个public函数，让用户直接调用，生成二维码。私有函数全由该共有函数进行调用，让该共有函数作为所有私有函数的接口。不知道这样是否合适。

在我看来，面对对象编程，也许是我没有get到精髓，我觉得某种程度上来说，是不是就是对于普通的编程，方便了参数的调用，整理了对于一个对象而言的函数打个包？

6. 附件：源程序

```
struct QRcodeNode
{
    int value;
    int isFunctionGraph;1
};
typedef struct Polymon
{
    int coef;//系数
    int expn;//指数
    Polymon *next;
}Polymon,*PolyLink;
class QRcode
{
private:
    int version;//版本
    char *info;//信息
    char *utf8;
    char *dataStream;//数据位流
    char *errorCodeStream;//纠错码位流
    char *codeStream;
    char *formatInfo;//格式信息
    int maskId;//掩码形式代码
    int NumBitOfData;
    QRcodeNode codeArr[N][N];//记录0 1 值
    void setSingleLocateValue(int I,int J,int
singleSize);
    void addBits();//添加补位码
    inline int nToAn(int n);
    inline int AnToN(int An);
    void setGLink(PolyLink *GPolyLhead, int
*generatorPolyExpn)
    int maskEvaluate1(QRcodeNode tmp[N][N]);
    int maskEvaluate2(QRcodeNode tmp[N][N]);
    int maskEvaluate3(QRcodeNode tmp[N][N]);
    int maskEvaluate4(QRcodeNode tmp[N][N]);
    void initCodeFunctionPart();
    void printCode();
    void dataCharToByteArr();//数据编码
    void setInfoPolymon(PolyLink
*InfoPLhead);//建立消息多项式链表
```

```
void GBKToUTF8(const char* strGBK, char* utf8)//
将编码转换成UTF8模式
{
    int len = MultiByteToWideChar(CP_ACP, 0,
strGBK, -1, NULL, 0);
    wchar_t* wstr = new wchar_t[len + 1];
    memset(wstr, 0, len + 1);
    MultiByteToWideChar(CP_ACP, 0, strGBK, -1,
wstr, len);
    len = WideCharToMultiByte(CP_UTF8, 0, wstr,
-1, NULL, 0, NULL, NULL);
    char* str = new char[len + 1];
    memset(str, 0, len + 1);
    WideCharToMultiByte(CP_UTF8, 0, wstr, -1,
str, len, NULL, NULL);
    strcpy(utf8, str);
    strcat(utf8, "\0");
    if (wstr) delete[] wstr;
    if (str) delete[] str;
}
int main()
{
    char s[100];
    cout << "请输入要输入的内容【不超过100字
节】:" << endl;
    cin.getline(s,100,'\n');//为了扫到空格，不
能用cin
    QRcode code(s);
    code.creatQRcode();
    return 0;
}
/*编码数据=模式指示符+计数指示符+编码后原串+补
位*/
void QRcode::dataCharToByteArr()
{
    void addBits(char *dataStream, int
maxNumBit);
    int len = strlen(info);
    char *utf8 = (char *)malloc(sizeof(char)*len
* 2);
```

```
    void setGeneratorPolymon(PolyLink *GPolyLhead);
    void errorCode();
    void fillCodeArr()
    void chooseMask();
    void setFormatInfo(QRcodeNode tmp[N][N], char *formatInfo);//在格式信息区域填充信息
    void setCodeArrMask();
public:
    QRcode(const char *s);
    ~QRcode();
    void creatQRcode();
};
void GBKToUTF8(const char* strGBK, char* utf8);
    char *binary = (char *)malloc(sizeof(char)*len * 16);
if (binary == NULL)
        exit(LOVERFLOW);
    char a;
    int bitCount = 8;
    for (int i = 0; utf8[i]; i++)
    {
        a = utf8[i];
        for (int j = 0; j < 8; j++)
        {
            binary[--bitCount] = char((a & 0x0001)+'0');
            a >>= 1;
        }
        bitCount += 16;
    }
    binary[bitCount - 8] = '\0';
    free(utf8);

    char modeIndicator[5] = "0100";
    strcpy(dataStream, modeIndicator);
    strcat(dataStream, charCountIndicator);
    strcat(dataStream, binary);
    QRcode::addBits();
}
void QRcode::setInfoPolymon(PolyLink *InfoPolyLhead)//建立消息多项式
{
    int expn[5] = { 7,10,15,20,26 };
    int numByteOfData = NumBitOfData /8;
    int expnValue = numByteOfData + expn[version - 1];
    PolyLink p =*InfoPolyLhead;
    for (int i = 0; i < NumBitOfData+8* expn[version - 1]; i += 8)
    {
        PolyLink s = (PolyLink)malloc(sizeof(Polymon));
        if (s == NULL)
            exit(LOVERFLOW);
    if (utf8 == NULL)
        exit(LOVERFLOW);
    GBKToUTF8(info, utf8);//GB->UTF8
    int length = strlen(utf8);
    char charCountIndicator[CHARCOUNTLENGTH + 1];//1-9版本的计数指示符为八位长
    charCountIndicator[CHARCOUNTLENGTH] = '\0';
~
    for (int i = CHARCOUNTLENGTH - 1; i >= 0; i--)
    {
        charCountIndicator[i] = (char)((length & 0x0001) + '0');//转换成二进制流
        length >>= 1;
    }
        p->next = s;
        p = s;
    }
    p->next = NULL;
}
void QRcode::errorCode()
{
    PolyLink GPolyLhead = (PolyLink)malloc(sizeof(Polymon));
    if (GPolyLhead == NULL)
        exit(LOVERFLOW);
    setGeneratorPolymon(&GPolyLhead);

    PolyLink InfoPLhead = (PolyLink)malloc(sizeof(Polymon));
    if (InfoPLhead == NULL)
        exit(LOVERFLOW);
    setInfoPolymon(&InfoPLhead);

    PolyLink p = InfoPLhead->next;
    PolyLink q = GPolyLhead->next;
    int infoHExpn = p->expn;
    int GHExpn = q->expn;
    for (int i = 0; i<=infoHExpn - GHExpn ; i++)//一直进行多项式相除，直到剩余项数为 生成多项式的次数
    {
        if (p->coef == 0)
        {
            p = p->next;
            continue;//如果系数已经是0，就直接过
        }
        int addExpn = AnToN(p->coef);
        q = GPolyLhead->next;
        PolyLink s = p;
        for (int j = 0; j <= GHExpn; j++)
        {
            int ceofTmp = q->coef + addExpn;
            if (ceofTmp > 255)
                ceofTmp = ceofTmp % 255;

            s->coef = (s->coef)
```

```
        if (i < NumBitOfData)
        {
            int coefValue = 0;
            for (int j = i; j < i + 8; j++)
            {
                coefValue <<= 1;
                coefValue += (dataStream[j] -
'0');
            }
            s->coef = coefValue;
        }
        else
            s->coef = 0;
        s->expn = --expnValue;

        {
            errorCodeStream[--index] =
char((errorCodeOct & 0x01) + '0');
            errorCodeOct >>= 1;
        }
        index += 16;
        p = p->next;
    }

    /*释放生成多项式的链表*/
    PolyLink w = GPolyLhead, ww;
    while (w)
    {
        ww = w->next;
        free(w);
        w = ww;
    }
    /*释放消息多项式的链表*/
    w = InfoPLhead;
    while (w)
    {
        ww = w->next;
        free(w);
        w = ww;
    }
}
inline int QRcode::nToAn(int n)//函数篇幅较短，
内置提高运行效率
{
    int An = 1;
    for (int i = 0; i < n; i++)
    {
        An*=2;
        if (An > 255)
            An = An ^ 285;
    }
    return An;
}

inline int QRcode::AnToN(int An)
{
^( nToAn(ceofTmp));
            s = s->next;
            q = q->next;
        }
        InfoPLhead->next = p->next;
        free(p);//随时释放消息多项式结点
        p = InfoPLhead->next;
    }
    p = InfoPLhead->next;
    int index = 8;
while (p)
    {
        int errorCodeOct = p->coef;
        for (int i=0;i<8;i++)

    int size = 17 + 4 * version;
    for (int j = size - 1; j > 0;j-=2 )
    {
        if ((j / 2) % 2 == 0)
            for (int i = size; i > 0; i--)//从
下到上
            {
                if (codeArr[i][j +
1].isFunctionGraph == 0)//越过功能区域
                {
                    if (*p == '1')
                codeArr[i][j + 1].value = 1;
                    p++;                }
                if
(codeArr[i][j].isFunctionGraph == 0)
                {
                    if (*p == '1')
                        codeArr[i][j].value
= 1;
                    p++;
                }
            }
        else if ((j / 2) % 2 == 1)//从上到下
            for (int i = 1; i <= size; i++)
            {
                if (codeArr[i][j +
1].isFunctionGraph == 0)
                {
                    if (*p == '1')
                        codeArr[i][j +
1].value = 1;
                    p++;
                }
                if
(codeArr[i][j].isFunctionGraph == 0)
                {
                    if (*p == '1')
                        codeArr[i][j].value
= 1;
                    p++;
                }
            }
```

装 订 线

```
    if (An < 1 || An>255)
        return LERROR;//-2
    for (int i = 1; i < 256; i++)
        if (nToAn(i) == An)
            return i;
    return RIGHT;//-1
}
void QRcode::fillCodeArr()
{
    strcpy(codeStream, dataStream);
    strcat(codeStream, errorCodeStream);
    if (version != 1)
        strcat(codeStream, "0000000");
    char *p = codeStream;
(codeArr[i][j].isFunctionGraph == 0 && (i + j) %
2 == 0)
                    codeArr[i][j].value
= !(codeArr[i][j].value);//((i + j) % 2)^
        break;

    case 1:
        for (int i = 0; i<size + 2; i++)
            for (int j = 0; j < size + 2; j++)
                if
(codeArr[i][j].isFunctionGraph == 0 &&(i-1)%2==0)
                    codeArr[i][j].value
= !(codeArr[i][j].value);//(i % 2)^
        break;

    case 2:
        for (int i = 0; i<size + 2; i++)
            for (int j = 0; j < size + 2; j++)
                if
(codeArr[i][j].isFunctionGraph == 0 &&(j-1)%3==0)
                    codeArr[i][j].value
=!(codeArr[i][j].value);// (j % 3)^
        break;

    case 3:
        for (int i = 0; i<size + 2; i++)
            for (int j = 0; j < size + 2; j++)
                if
(codeArr[i][j].isFunctionGraph == 0 &&
(i+j-2)%3==0)
                    codeArr[i][j].value
= !(codeArr[i][j].value);//((i + j) % 3)^
        break;

    case 4:
        for (int i = 0; i<size + 2; i++)
            for (int j = 0; j < size + 2; j++)
            {
                int v = (((i - 1) % 2) ? ((i -
1) / 2 + 1) : ((i - 1) / 2)) + (((j - 1) % 3) ? ((j
- 1) / 3 + 1) : ((j - 1) / 3));
                if
(codeArr[i][j].isFunctionGraph == 0 && v%2==0)
    }
    return;
}
void QRcode::setCodeArrMask()
{
    int size = 17 + 4 * version;//公式
    setFormatInfo(codeArr, formatInfo);
    switch (maskId)
    {
    case 0:
        for (int i = 1; i < size + 2; i++)
            for (int j = 1; j < size + 2; j++)
                if
 ^
        break;

    case 6:
        for (int i = 0; i<size + 2; i++)
            for (int j = 0; j < size + 2; j++)
                if
(codeArr[i][j].isFunctionGraph == 0 &&
((((i-1)*(j-1)) % 3 + ((i-1)*(j-1)) % 2) % 2)==0)
                    codeArr[i][j].value
=!(codeArr[i][j].value);//(((i*j) % 3 + (i*j) %
2) % 2)^
        break;

    case 7:
        for (int i = 0; i<size + 2; i++)
            for (int j = 0; j < size + 2; j++)
                if
(codeArr[i][j].isFunctionGraph == 0 && (((i +
j-2) % 3 + (i + j-2) % 2) % 2)==0)
                    codeArr[i][j].value
= !(codeArr[i][j].value);//(((i + j) % 3 + (i + j) %
2) % 2)^
        break;

    default:break;
    }
}
void QRcode::creatQRcode()
{
    initCodeFunctionPart();
    dataCharToByteArr();
    errorCode();
    fillCodeArr();
    chooseMask();
    setCodeArrMask();
    printCode();
}
```

```
                        codeArr[i][j].value
= !(codeArr[i][j].value);//(v % 2)^
                }
            break;

        case 5:
            for (int i = 0; i<size + 2; i++)
                for (int j = 0; j < size + 2; j++)
                    if
(codeArr[i][j].isFunctionGraph == 0 &&
(((i-1)*(j-1)) % 3 + ((i-1)*(j-1)) % 2)==0)
                        codeArr[i][j].value
=!(codeArr[i][j].value); //((i*j) % 3 + (i*j) % 2)
```

装 订 线