Qwiget的子类的对象所有的功能函数setStylesheet的部分功能：

1 Dialog::Dialog(QWidget \*parent) :

2 QDialog(parent),

3 ui(new Ui::Dialog)

4 {

5 ui->setupUi(this);

6 this->setWindowFlags(this->windowFlags()&Qt::WindowMaximizeButtonHint&Qt::WindowMinimizeButtonHint);//为对话框添加上最大化和最小化按钮

7 // layout=new QBoxLayout(this);

8 layout1=new QGridLayout(this);

9 btn1=new QPushButton(this);

10 btn1->setStyleSheet("QPushButton{color:red;background:yellow}"); //设定前景颜色，就是字体颜色

11 // btn1->setStyleSheet("QPushButton{background:yellow}");

12 btn1->setText("Button1");

13

14 btn2=new QPushButton(this);

15 btn2->setStyleSheet("QPushButton{color:red; //使用rgb来设定背景颜色

16 btn2->setText("Button2");

17

18 btn3=new QPushButton(this);

19 btn3->setStyleSheet("QPushButton{background-image:url(image/1.png);background-repeat: repeat-xy;background-position: center;background-attachment: fixed;background-attachment: fixed;background-attachment: fixed;;background-clip: padding}");

20 //设定按钮的背景图片,background-repeat可以设定背景图片的重复规则，这里设定仅在xy方向都重复，所以图片会被重复一次

21 //background-position用来设定图片的位置，是左(left)还是右(right)，还是在中间(center),是上(top)还是底部(bottom)

22 //background-attachment用来这定背景图片是否卷动或者和窗口大小相匹配，默认是卷动的

23 btn3->setText("Button3");

24

25 btn4=new QPushButton(this);

26 btn4->setStyleSheet("QPushButton{border: 3px solid red;border-radius:8px}"); //设定边框宽度以及颜色

27 //可以使用border-top，border-right，border-bottom，border-left分别设定按钮的上下左右边框，

28 //同样有border-left-color, border-left-style, border-left-width.等分别来设定他们的颜色，样式和宽度

29 //border-image用来设定边框的背景图片。

30 //border-radius用来设定边框的弧度。可以设定圆角的按钮

31 btn4->setText("Button4");

32

33 //字体设定

34 //font-family来设定字体所属家族，

35 //font-size来设定字体大小

36 //font-style来设定字体样式

37 //font-weight来设定字体深浅

38 //height用来设定其高低

39 //selection-color用来设定选中时候的颜色

40 edit1=new QLineEdit(this);

41 edit1->setStyleSheet("QLineEdit{font: bold italic large /"Times New Roman/";font-size:25px;color:rgb(55,100,255);height:50px;border:4px solid rgb(155,200,33);border-radius:15px;selection-color:pink}");

42

43 //父窗口的设定

44 //icon-size来设定图片大小

45 this->setWindowIcon(QIcon("image/1.png"));

46 this->setStyleSheet("QWidget{background:write url(image/2.png);icon-size:20px 5px}"); //设定整个对话框的背景颜色

47 // this->setStyleSheet("QWidget{icon-size:20px 5px}");

48 layout1->addWidget(btn1,0,0);

49 layout1->addWidget(btn2,0,1);

50 layout1->addWidget(btn3,1,0);

51 layout1->addWidget(btn4,1,1);

52 layout1->addWidget(edit1,2,0);

53 }

54

btn1->setStyleSheet("QPushButton{color:red}"); //设定前景颜色，就是字体颜色

2 btn1->setStyleSheet("QPushButton{background:yellow}"); //设定背景颜色为红色

和

1 btn1->setStyleSheet("QPushButton{color:red；background:yellow}");

Qt常用类：

QDataStream 为QIODevice提供了一串的二进制的数据

QDataWidgetMapper 提供了从数据模型到部件的映射

QData 日期的功能

QDateEdit 提供了日期的部件用来编辑日期，是基于QDateTimeEdit 部件

QDateTime 提供日期与时间的功能

QDateTimeEdit 提供了编辑日期与实践的部件

QDBusAbstractAdaptor 提供了 基于D-BUS适配器的基础累

QDBusAbstractInterface 提供了所有QtDBus 绑定以及能够访问的D-BUS 界面

QDebug 提供了输出调试信息流， 从后台输出

QDecoration 提供了Window 与嵌入式linux 的装饰支持

QDecorationDefault 提供了默认窗口的装饰

QDesktopSerivces 提供了访问常用桌面服务的方法

QDesktopWidget 提供了访问多头的屏幕信息的功能

QDial 提供了滑盘控制的支持

QDialog 提供了窗口系统的基类

QDialogButtonBox 提供了使用于当前部件类型的按钮对话框

QDir 提供了访问目录的结构与内容的支持

QDirectPainter 提供了直接用户绘画的硬件支持， 使用在嵌入式linux 当中

QDirModel 提供了用于访问本地文件系统的模型支持

QDockWidget 提供了在一个QMainWindow 中的窗口或者作为桌面的顶级窗口出现

Effect 用来传输音频流

EffectParameter 用来描述一个effect 的参数

EffectWidget 提供了一个用来 控制effect 参数的部件

QErrorMessage 通过一个对话框来显示错误信息

QEvent 提供了所有事件累的积累，事件对象包含了时间的参数

QEventLoop 提供了进入和离开事件循环的功能

Exception 提供了在线程之间传输的异常

QFile 提供了用于读写文件的界面

QFileDialog 提供了一个对话框使得用户可以选择文件或者目录

QFileIconProvider 为QDirModel 类提供的文件图标

QFileInfo 提供基于平台依赖的文件信息

QFileOpenEvent 当打开一个文件或者打开一个url 地址的时候抛出一个事件

QFileSystemModel 提供了本地文件系统的一个数据模型

QFileSystemWatcher 提供了坚实本地文件与目录的监视功能

QFocusEvent 当一个部件获得焦点则发出事件

QFocusFrame 提供了当前具有绘画的区域的焦点

QFont 为绘画文本提供制定的字体

QFontComboBox 提供了选择字体族的下拉列表框

QFontDatabase 提供了可以在Window 操作系统下能够使用的字体

QFontDialog 用来选择一个指定的字体

QFontEngineInfo 用来描述一个指定的字体引擎插件

QFontEnginePlugin 累提供了字体仓库插件的基础累， 用于嵌入式linux

QFontInfo 包含了一个字体的常规信息

QFontDialog 用来弹出对话框 来获得指定的字体

QFtp 提供了ftp通讯的客户端程序的实现功能

QGesture 包含了用户输入的信息描述

QGestureEvent 当一个用户输入来临触发此事件

QGLColormap 提供了用于QGLWidget 的自定义颜色图

QGLContext 包含了OpenGL 的渲染内容

QGLFormat 提供了OpenGL 渲染内容的格式

QGLFramebufferObject 包含了OpenGL 的帧缓冲对象

QGLFramebufferObjectFormat 指定了当前OpenGL 帧缓冲的对象格式

QGLPixelBuffer 包含了OpenGL pbuffer

QGLWidget 提供了用来渲染 OpenGL的图形部件

QGraphicsScale 提供了缩放的翻译

QGraphicsScene 提供了2D 图形项目的管理界面

QGraphicsView 他提供了QGraphicsScene 内容的显示部件

QGraphicsWebView 将web 内容添加到了GraphicsView 中

QGridLayout 提供了网状布局

QGroupBox 提供了拥有相同类型的组框， 并带有标题

QHash 模板类， 提供了哈希表字典

QHashIterator 提供了java 类型的迭代器， 主要是给QHash 与QMultiHash 来使用

QHBoxLayout 部件横向布局

QHideEvent 当前部件隐藏时发出的事件

QHostAddress 提供ip 地址

QHostInfo 提供了静态的函数来对主机进行查询

QHoverEvent 提供描述鼠标的事件，当鼠标徘徊在某一处的时候

QIcon 提供了可以缩放的图标， 可用在不同模式下不同状态中

QIconEngine 绘画QIcon 的抽象基础类

QImage

QImageReader 提供了从一个文件或者其他设备读取指定文件格式的界面

QImageWriter 提供了写入一个文件或者其他设备所指定的文件格式的界面

QInputContext 作为一种输入法 ， 他依赖于具体的数据与状态

QInputDialog 提供了方便的对话框让用户来输入单个数值

QInputEvent 当用户输入的时候发出信息

QInputMethodEvent 输入发生时抛出事件

QIODevice 他是Qt 中所有输入输出界面的基类

QKeyEvent 一个键盘输入事件发生

QLabel 在一个部件显示文字或者图片

QLayout 提供了几何位置管理的基础类

QLCDNumber 提供了一个部件用于类似LCD 的现实数字

QLibrary 在程序运行时加载共享库文件

QLibaryInfo 提供Qt 库的信息

QLine 两个坐标点提供一条具有芳香的直线

QLineEdit 提供了一行文本输入的编辑器

QList 模板类， 提供链表的基本操作功能

QListIterator 提供了java 类型的链表迭代器

QListView 提供列表或者提供图标的试图， 要求显示在一个模型上

QListWidget 提供了基于一个项目的列表部件

QListWidgetItem 创建项目用于QListWidget视图的显示

QLocalServer 本地套接字的基础服务器端

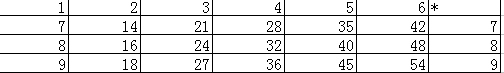
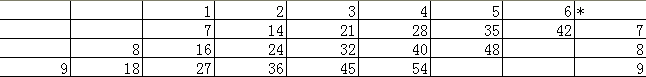
QLocalSocket 提供了一个本地的套接字

大整数乘法-分治法

http://simg.sinajs.cn/blog7style/images/common/sg_trans.gif (2015-10-26 12:46:10)

[[http://simg.sinajs.cn/blog7style/images/common/sg_trans.gif](javascript:;)转载▼](javascript:;)

|  |  |
| --- | --- |
| 标签：  [佛学](http://search.sina.com.cn/?c=blog&q=%B7%F0%D1%A7&by=tag)    [股票](http://search.sina.com.cn/?c=blog&q=%B9%C9%C6%B1&by=tag)    [军事](http://search.sina.com.cn/?c=blog&q=%BE%FC%CA%C2&by=tag)    [教育](http://search.sina.com.cn/?c=blog&q=%BD%CC%D3%FD&by=tag) |  |

一、两个整数相乘，需要考虑计算的容量，把大整数划分为多个部分（通常两个部分），则问题规模变小，存储容量变小。  
 先介绍123456\*789问题，按照数学常规思考问题，可以写成下列表格形式：  
表一  
        [](http://photo.blog.sina.com.cn/showpic.html#blogid=52f623240102vvme&url=http://album.sina.com.cn/pic/001wc72cgy6Wv7jQcCde1)  
表二：  
[](http://photo.blog.sina.com.cn/showpic.html#blogid=52f623240102vvme&url=http://album.sina.com.cn/pic/001wc72cgy6Wv7JSAU525)  
表三  
[](http://photo.blog.sina.com.cn/showpic.html#blogid=52f623240102vvme&url=http://album.sina.com.cn/pic/001wc72cgy6Wv83sUPt67)  
  
由表三得到：  
1.m位整数\*n位整数乘积m+n-1位，也可能进位乘积为m+n位  
2.进位是表中最后一列和除以进制得到的商，保留位为表中和除以进制得到的余数  
3.因为要取得每一位相乘，故采用字符数组存放整数，可以定义三个字符数组分别存储乘数、被乘数、乘积。  
算法代码  
void   f（char \*a,char \*b,char  \* &c）{  
int carry=0;//定义进位  
  
for（int i=0;a[i]!='\0';i++）{  
a[i]-='0';  
}  
for（int i=0;b[i]!='\0';i++）{  
b[i]-='0';  
}  
//计算乘积，乘积个位算起  
for(int k=i+j;k>0;k--){  
   sum=carry;  
    //乘数\*被乘数首先是乘数个位乘被乘数高位  
    
   for(j=0;str[j]!='\0';j++){  
   sum+=a[i-j][j];  
   c[i+1]=sum+'0';  
carry=sum/10;  
}  
 if((c[0]=carry+'0')=='0'){  
c[0]='\040';//设置为空格  
}}  
}  
效率分析：需要进行m\*n次乘法运算，接着在进行m+n次加法即m+n次的取摸运算。  
二、接下来改进算法  
我们可以将大整数对拆为两个部分。  
即a和b相乘就可以写为：a \* b = {  a1 \* 10^(n1/2)   +  a0  }   \*  {  b1 \* 10^(n2/2)  +  b0 }  
  
展开后整理得： a  \*  b  =  a1\*b1 \* 10^[ (n1+n2)/2 ]   +  a1\*b0 \* 10^(n1/2)   +   a0\*b1 \* 10^(n2/2)  + a0\*b0　；  
**这样就很容易递归的来求a \* b，如果你嫌分解后的数还太大，就可以继续分解**  
实现方法：我们定义一个支持方法f(char \*a,char \*b)，用于在结束递归时（在本例中，我定义有一个数是1位时结束递归，直接用普通乘法）计算两个字符串的乘积（为了表示大数，用字符串来接受参数）。有 了这个支持方法，分治递归实现两个大数乘法的实现如下：

int f(char \*a,char \*b)//用字符串读入2个大整数  
{  
long result = 0;  
if(len(a) == 1 || len == 1) //递归结束的条件  
result = f(a,b);  
else //如果2个字符串的长度都 >= 2  
{  
  char \*a1 =a;  
\*(a1+(len(a)/2))='\0'; //截取前一半的字符串(较短的一半)  
  char \*a0 = a+len(a)/2; //截取后一半的字符串  
  \*(a0+len(a)/2)='\0';  
char \*b1=b;  
\*(b1+(len(a)/2))='\0'; //截取前一半的字符串(较短的一半)  
  char \*b0 = a+len(a)/2; //截取后一半的字符串  
  \*(b0+len(b)/2)='\0';  
  
//分治的思想将整数写成这样： a = a1 \* 10^(n1/2) + a0, b = b1 \* 10^(n2/2)，相乘展开得到以下四项  
//其中n1，n2为2个整数a，b的位数  
result = (f(a1,b1) \* pow(10,len( a0) + len(b0))  
+ f(a1,b0) \* pow(10, len(a0) +f(a0,b1) \* pow(10,len(b0))  
+ fy(a0,b0));  
}  
  
return result;  
}

分

**input sample**

2  
2  
0 0  
8  
2 2

**output sample**

CASE:1  
0  1    
1  1    
CASE:2  
3  3  4  4  8  8  9  9    
3  2  2  4  8  7  7  9    
5  2  0  6  10 10 7  11   
5  5  6  6  1  10 11 11   
13 13 14 1  1  18 19 19   
13 12 14 14 18 18 17 19   
15 12 12 16 20 17 17 21   
15 15 16 16 20 20 21 21

**代码如下：**

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include<string.h>  #define N 10000  int board[N][N];//棋盘的布局  int team;//用方块覆盖，相当于分组  int loc;//key(x,y)的方位（返回值为1、2、3、4分别指1、2、3、4象限）  int witk(int m,int x,int y,int a,int b){//where is the key(x,y)?  m为棋盘的边长，x，y为黑点 的坐标，a，b为棋盘左上角的坐标，      int n;      n=m/2;      if(x<n+a&&y>=n+b)          return 1;      else if(x<n+a&&y<n+b)          return 2;      else if(x>=n+a&&y<n+b)          return 3;      else if(x>=n+a&&y>=n+b)          return 4;  }  void cover(int m,int a,int b){      int i;      int j;      int x,y;      int n;      if(m==2){//若m==2，直接覆盖          for(i=a;i<m+a;i++)              for(j=b;j<m+b;j++)                  if(board[i][j]==-1)                      board[i][j]=team;          team++;      }      else{//m>2          for(i=a;i<m+a;i++){//找出黑点的位置x，y              for(j=b;j<m+b;j++)                  if(board[i][j]!=-1){                      x=i;                      y=j;                  }          }          n=m/2;          loc=witk(m,x,y,a,b);//where is the key(x,y)?          for(i=a+n-1;i<=a+n;i++)//遍历棋盘中部四个格，判断其方位，若不和黑点在同一个方位则将其覆盖；              for(j=b+n-1;j<=b+n;j++)                  if(witk(m,i,j,a,b)!=loc)                      board[i][j]=team;          team++;          cover(n,a,b);//覆盖四个分区域，递归调用          cover(n,a,b+n);          cover(n,a+n,b);          cover(n,a+n,b+n);      }  }  int main(){      int i,j,k;      int n,m;      int x,y;      freopen("in.txt","r",stdin);      freopen("out.txt","w",stdout);      scanf("%d",&n);      for(i=0;i<n;i++){          team=1;          memset(board,-1,sizeof(board));//将棋盘全部置为-1          scanf("%d%d%d",&m,&x,&y);          board[x+1][y+1]=0;          cover(m,1,1);          printf("CASE:%d\n",i+1);          for(j=1;j<=m;j++){              for(k=1;k<=m;k++)                  printf("%-3d",board[j][k]);              printf("\n");          }      }      return 0;  } |