//计算代码  
        private void button1\_Click(object sender, EventArgs e)  
         
{  
string[] sd = new string[dataGridView1.RowCount-5];  
double[] sdr = new double[sd.Length];   
double[] cr = new double[sd.Length ];   
double sum = 0;  
cr[0] = dmstorad(Convert.ToString(dataGridView1.Rows[0].Cells[4].Value));   
double acd = dmstorad(Convert.ToString(dataGridView1.Rows[dataGridView1.RowCount-6].Cells[4].Value));   
for (int i = 1; i < sd.Length; i++)   
{  
sd[i] = Convert.ToString(dataGridView1.Rows[i].Cells[1].Value);   
sdr[i] = dmstorad(sd[i]);  
}  
sum = fangweijiao(sdr, cr);   
dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[1].Value = radtodms(sum);   
double fd, fdx;   
fd = cr[cr.Length - 1] - acd;  
fdx = 60 \* Math.Sqrt(sd.Length - 1);  
dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[1].Value = Convert.ToString(Math.Round(fd \* 180 / Math.PI \* 3600, 2))+"″";   
dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[1].Value = Convert.ToString(Math.Round(fdx, 2))+"″";  
if (Math.Abs(fd \* 180 / Math.PI \* 3600) > fdx)  
MessageBox.Show("角度闭合差超限！");  
else  
{  
double vd = -fd / (sd.Length - 1);  
double sumvd = 0;   
for (int i = 1; i < sdr.Length; i++)   
{  
sdr[i] += vd;  
sumvd += vd;   
dataGridView1.Rows[i].Cells[2].Value = Convert.ToString(Math.Round(vd \* 180 / Math.PI \* 3600, 2))+"″";   
dataGridView1.Rows[i].Cells[3].Value = radtodms(sdr[i]);  
}  
if (Math.Round(sumvd, 8) != Math.Round(-fd , 8))   
MessageBox.Show("角度改正数分配有误！");  
else  
dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[2].Value = Convert.ToString(Math.Round(sumvd \* 180 / Math.PI \* 3600, 2)) + "″";   
sum = fangweijiao(sdr, cr);  
if (Math.Round(cr[cr.Length - 1], 8) != Math.Round(acd, 8))   
MessageBox.Show("坐标方位角推算有误！");  
else  
{  
dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[3].Value = radtodms(sum);   
for (int i = 1; i < cr.Length-1; i++)  
dataGridView1.Rows[i].Cells[4].Value = radtodms(cr[i]);  
}  
}

//坐标增量的计算和调整  
double[] jl = new double[sd.Length - 1];  
double[] dx = new double[sd.Length - 1];  
double[] dy = new double[sd.Length - 1];  
double sumjl = 0;  
double sumdx = 0;  
double sumdy = 0;  
for (int i = 1; i < jl.Length; i++)  
{  
    jl[i] = Convert.ToDouble(dataGridView1.Rows[i].Cells[5].Value);  
    //将观测距离放入数组中   
    sumjl += jl[i]; //计算距离总和   
    dx[i] = jl[i] \* Math.Cos(cr[i]); //计算坐标增量   
    dy[i] = jl[i] \* Math.Sin(cr[i]);  
    sumdx += dx[i]; //计算坐标增量总和   
    sumdy += dy[i];  
    dataGridView1.Rows[i].Cells[6].Value = Convert.ToString(Math.Round(dx[i], 3));  
    //将坐标增量放入表格   
    dataGridView1.Rows[i].Cells[7].Value = Convert.ToString(Math.Round(dy[i], 3));  
}  
dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[5].Value = Convert.ToString(suml);  
//将距离总和放入表格中   
dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[6].Value = Convert.ToString(Math.  
Round(sumdx, 3)); //将坐标增量总和放入表格中   
dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[7].Value =  
Convert.ToString(Math.Round(sumdy, 3));  
  
double x2, y2, x3, y3; //存放已知两个点的x，y坐标   
x2 = Convert.ToDouble(dataGridView1.Rows[1].Cells[12].Value);  
y2 = Convert.ToDouble(dataGridView1.Rows[1].Cells[13].Value);  
x3 = Convert.ToDouble(dataGridView1.Rows[sd.Length - 1].Cells[12].Value);  
y3 = Convert.ToDouble(dataGridView1.Rows[sd.Length - 1].Cells[13].Value);  
double fx, fy, fxy, k1;  
fx = sumdx - (x3 - x2); //计算坐标增量闭合差   
fy = sumdy - (y3 - y2);  
fxy = Math.Sqrt(fx \* fx + fy \* fy); //计算导线全长闭合差   
k1 = suml / fxy; //计算导线全长相对闭合差分母   
dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[7].Value =  
Convert.ToString(Math.Round(fx, 3));  
dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[7].Value =  
Convert.ToString(Math.Round(fy, 3));  
dataGridView1.Rows[dataGridView1.RowCount - 3].Cells[10].Value =  
Convert.ToString(Math.Round(fxy, 3));  
dataGridView1.Rows[dataGridView1.RowCount - 2].Cells[11].Value =  
Convert.ToString((int)k1);//导线全长相对闭合差分母取整 导线全长相对闭合差；1/整数   
double[] vx = new double[sd.Length - 1]; //定义数组用于存放坐标增量的改正数及总和   
double[] vy = new double[sd.Length - 1];  
double sumvx = 0;  
double sumvy = 0;  
double[] cx = new double[sd.Length - 1]; //定义数组用于存放改正后的坐标增量及总和   
double[] cy = new double[sd.Length - 1];  
double sumcx = 0;  
double sumcy = 0;  
double[] x = new double[sd.Length - 1]; //定义数组用于存放x，y坐标   
double[] y = new double[sd.Length - 1];  
x[1] = x2;  
y[1] = y2;  
  
if (k1 < 2000) //判断导线全长相对闭合差是否超限   
    MessageBox.Show("导线全长相对闭合差超限！");  
else

{  
    for (int j = 1; j < vx.Length; j++)  
    {  
        vx[j] = -fx \* sl[j] / suml; //计算坐标增量改正数   
        vy[j] = -fy \* sl[j] / suml;  
        sumvx += vx[j]; //计算坐标增量改正数总和   
        sumvy += vy[j];  
        dataGridView1.Rows[j].Cells[8].Value = Convert.ToString(Math.Round(vx[j], 4));  
        //将坐标增量改正数放入表格   
        dataGridView1.Rows[j].Cells[9].Value = Convert.ToString(Math.Round(vy[j], 4));  
        cx[j] = dx[j] + vx[j]; //计算改正后坐标增量   
        cy[j] = dy[j] + vy[j];  
        sumcx += cx[j]; //计算改正后坐标增量总和   
        sumcy += cy[j];  
        dataGridView1.Rows[j].Cells[10].Value = Convert.ToString(Math.Round(cx[j], 3));  
        //将改正后坐标增量放入表格   
        dataGridView1.Rows[j].Cells[11].Value = Convert.ToString(Math.Round(cy[j], 3));  
    }  
    //C#中 1/2000 = 0 两个整数相除，可以写 成1.0/2000 保留小数位数   
    if (Math.Round(sumvx, 4) != Math.Round(-fx, 4) || Math.Round(sumvy, 4) !=  
    Math.Round(-fy, 4))  
        MessageBox.Show("坐标增量分配有误！");  
    if (Math.Round(sumcx, 4) != Math.Round(x3 - x2, 4) || Math.Round(sumcy, 4) != Math.Round(y3 - y2, 4))  
        MessageBox.Show("改正后坐标增量计算有误！");  
    dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[8].Value =  
    Convert.ToString(Math.Round(sumvx, 3)); //将坐标增量改正数总和放入表格中   
    dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[9].Value =  
    Convert.ToString(Math.Round(sumvy, 3));  
    dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[10].Value =  
    Convert.ToString(Math.Round(sumcx, 3)); //将改正后坐标增量总和放入表格中   
    dataGridView1.Rows[dataGridView1.RowCount - 4].Cells[11].Value =  
    Convert.ToString(Math.Round(sumcy, 3));  
    for (int j = 1; j < x.Length - 1; j++)  
    {  
        x[j + 1] = x[j] + cx[j]; //计算x,y坐标   
        y[j + 1] = y[j] + cy[j];  
        dataGridView1.Rows[j + 1].Cells[12].Value = Convert.ToString(Math.Round(x[j + 1], 3));  
        //将x,y坐标放入表格   
        dataGridView1.Rows[j + 1].Cells[13].Value = Convert.ToString(Math.Round(y[j + 1], 3));  
    }  
    if (Math.Round(x[x.Length - 1] + cx[cx.Length - 1], 3) != Math.Round(x3, 3) || Math.Round  
        (y[y.Length - 1] + cy[cy.Length - 1], 3) != Math.Round(y3, 3))  
        MessageBox.Show("坐标计算有误！");  
}   
  
        }  
  
        private void excelToolStripMenuItem\_Click(object sender, EventArgs e)  
        {    //Excel输入代码  
            dataGridView1.DataSource = null;  
            dataGridView1.Rows.Clear();  
            dataGridView1.Columns.Clear();  
            OpenFileDialog file = new OpenFileDialog();  
            file.Filter = "Excel文件|\*.xls|Excel文件|\*.xlsx";  
            if (file.ShowDialog() == DialogResult.OK)  
            {  
                string fname = file.FileName;  
                string strSource = "provider=Microsoft.ACE.OLEDB.12.0;" + "Data Source=" + fname + ";Extended Properties=&apos;Excel 8.0;    HDR=Yes;IMEX=1&apos;";  
  
                OleDbConnection conn = new OleDbConnection(strSource);  
                string sqlstring = "SELECT \* FROM [Sheet1$]";  
                OleDbDataAdapter adapter = new OleDbDataAdapter(sqlstring, conn);  
                DataSet da = new DataSet();  
                adapter.Fill(da);  
                dataGridView1.DataSource = da.Tables[0];  
            }  
            else  
                return;  
        }  
    }  
}