```
void shadow()
{
    int i,j,r,count=0,avg,sum=0;
    int px,py,p1x,p1y;
    r=257;
    do
    {
        for(i=0; i<width; i++)
        {
            for(j=0; j<height; j++)
            {
                if(b[i][j]==r)
                    sum=sum+output[i][j];
                    count=count+1;
            }
        }
        avg=sum/count;
        if(avg>= 245 && avg <=255)  //black region
        {
            px=i;
            py=j;
            for(i=0; i<width; i++)
            {
                for(j=0; j<height; j++)
                {
                    if(b[i][j]==r)
                    {
                        if (j<py)
                        {
                            px=i;
                            py=j;  //left most pixel
                        }
                        else
                             break;
                    }
                }
            }
            if(b[i-1][j]==0)
            {
                p1x=i-1;
                p1y=j;
                flag[i-1][j]=1;
            }
            else if(b[i+1][j]==0)
            {
                p1x=i+1;
                p1y=j;
                flag[i+1][j]=1;
            }
            else if (b[i][j-1]==0)
            {
                p1x=i;
                p1y=j-1;
                flag[i][j-1]=1;
            }
            else if (b[i][j+1]==0)
            {
```

```
            p1x=i;
            p1y=j+1;
            flag[i][j+1]=1;
        }
        dv=(((p1x-px)^2)+(p1y-py)^2))^1/2;
        do
        {
            f=0;
            if(b[i-1][j]==0 && flag[i][j]==0)
            {
                pp1x=i-1;
                pp1y=j;
            }
            else if(b[i+1][j]==0 && flag[i][j]==0)
            {
                pp1x=i+1;
                pp1y=j;
            }
            else if (b[i][j-1]==0 && flag[i][j]==0)
            {
                pp1x=i;
                pp1y=j-1;
            }
            else if (b[i][j+1]==0 && flag[i][j]==0)
            {
                pp1x=i;
                pp1y=j+1;
            }


            if(b[i-1][j]==0)
            {
                pp2x=i-1;
                pp2y=j;
                flag[i-1][j]=1;
            }
            else if(b[i+1][j]==0)
            {
                pp2x=i+1;
                pp2y=j;
                flag[i+1][j]=1;
            }
            else if (b[i][j-1]==0)
            {
                pp2x=i;
                pp2y=j-1;
                flag[i][j-1]=1;
            }
            else if (b[i][j+1]==0)
            {
                pp2x=i;
                pp2y=j+1;
                flag[i][j+1]=1;
            }

            dv1=(((pp1x-ppx)^2)+(pp1y-ppy)^2))^1/2;
```

```
            if (dv1>dv)     //vertival length
            {   dv=dv1;
                f=1;
            }
        }while(f=1);
    }

    sum=0;
    count=0;
    for(i=0; i<width; i++)
    {
        for(j=0; j<height; j++)
        {
            if(b[i][j]==r)
                sum=sum+output[i][j];
                count=count+1;
        }
    }
    avg=sum/count;
    if(avg>= 245 && avg <=255)  //black region
    {
        px=i;
        py=j;
        for(i=0; i<width; i++)
        {
            for(j=0; j<height; j++)
            {
                if(b[i][j]==r)
                {
                    if (i<px)
                    {
                        px=i;
                        py=j;  //left most pixel
                    }
                    else
                         break;
                }
            }
        }
        if(b[i-1][j]==0)
        {
            p1x=i-1;
            p1y=j;
            flag[i-1][j]=1;
        }
        else if(b[i+1][j]==0)
        {
            p1x=i+1;
            p1y=j;
            flag[i+1][j]=1;
        }
        else if (b[i][j-1]==0)
        {
            p1x=i;
            p1y=j-1;
            flag[i][j-1]=1;
        }
```

```
        else if (b[i][j+1]==0)
        {
            p1x=i;
            p1y=j+1;
            flag[i][j+1]=1;
        }
        dh=(((p1x-px)^2)+(p1y-py)^2))^1/2;
        do
        {
            f=0;
            if(b[i-1][j]==0 && flag[i][j]==0)
            {
                pp1x=i-1;
                pp1y=j;
            }
            else if(b[i+1][j]==0 && flag[i][j]==0)
            {
                pp1x=i+1;
                pp1y=j;
            }
            else if (b[i][j-1]==0 && flag[i][j]==0)
            {
                pp1x=i;
                pp1y=j-1;
            }
            else if (b[i][j+1]==0 && flag[i][j]==0)
            {
                pp1x=i;
                pp1y=j+1;
            }


            if(b[i-1][j]==0)
            {
                pp2x=i-1;
                pp2y=j;
                flag[i-1][j]=1;
            }
            else if(b[i+1][j]==0)
            {
                pp2x=i+1;
                pp2y=j;
                flag[i+1][j]=1;
            }
            else if (b[i][j-1]==0)
            {
                pp2x=i;
                pp2y=j-1;
                flag[i][j-1]=1;
            }
            else if (b[i][j+1]==0)
            {
                pp2x=i;
                pp2y=j+1;
                flag[i][j+1]=1;
            }
```

```
            dh1=(((pp1x-ppx)^2)+(pp1y-ppy)^2))^1/2;

            if (dh1>dh)     //vertival length
            {   dh=dh1;
                f=1;
            }
        }while(f=1);
    }
```