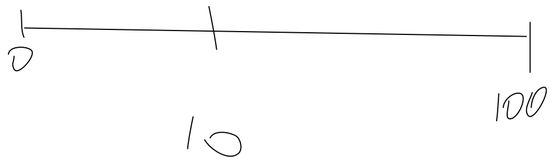
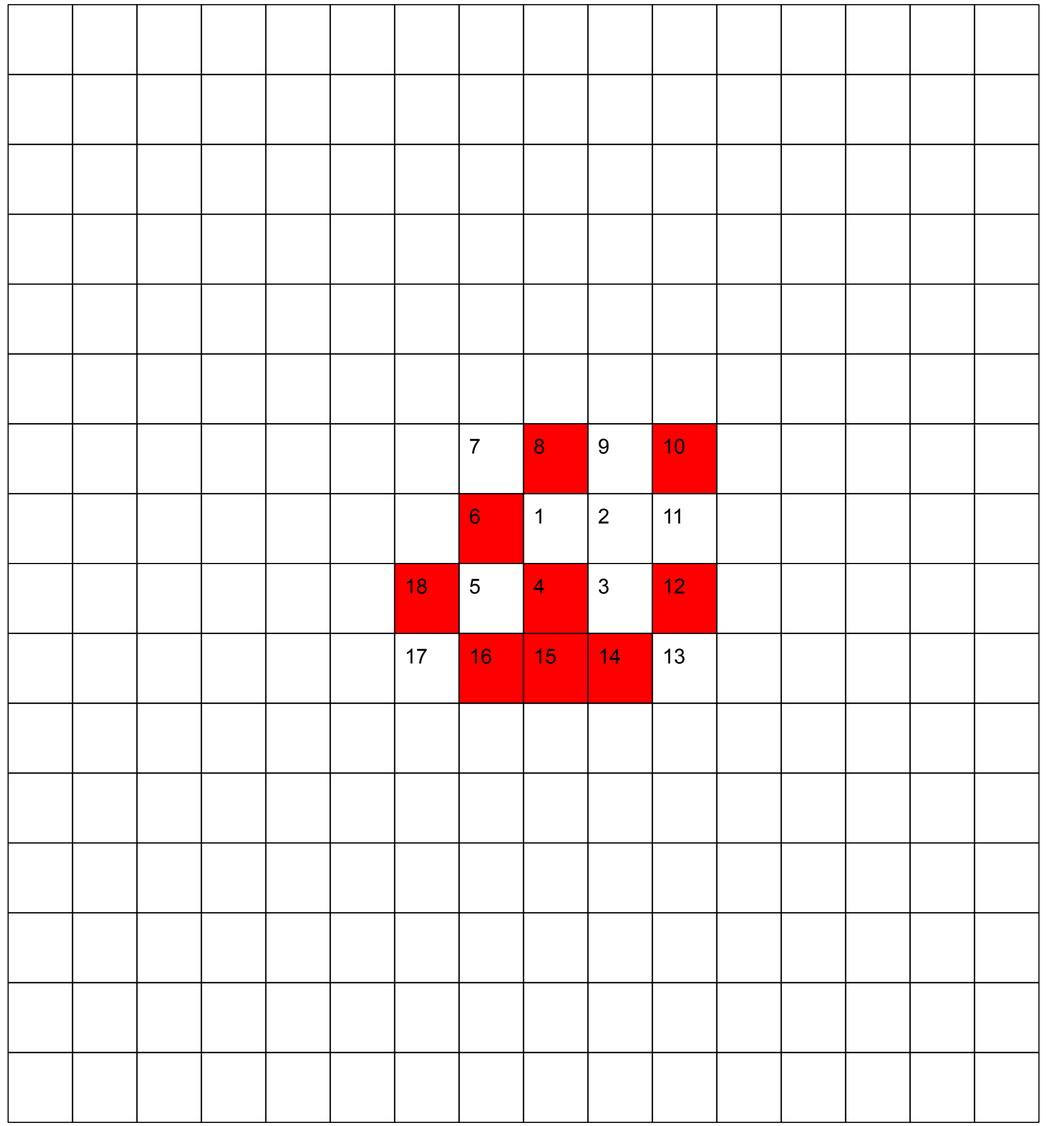



```
void gg()
{
        int c_x=WIDTH/2;
        int c_y=HEIGHT/2;
        int i;
        for(i=0;i<100;i=i+2)
        {
                c_x+=1+i;
                printf("x=%d y=%d\n", c_x, c_y);
                c_y-=1+i;
                printf("x=%d y=%d\n", c_x, c_y);
                c_x-=2+i;
                printf("x=%d y=%d\n", c_x, c_y);
                c_y+=2+i;
                printf("x=%d y=%d\n", c_x, c_y);
        }
}
```




```
void spiral_v2(struct point shade)
{
        int c_x=WIDTH/2;
        int c_y=HEIGHT/2;
        truecolor[c_x][c_y].blue=0;
    truecolor[c_x][c_y].green=255;
    truecolor[c_x][c_y].red=0;
    int i;
    int k;
    int g=1;
    for(i=0;i<10000;i=i+2)
    {
        for(k=0;k<i+1;k++)
        {
            c_x+=1;
            //printf("x=%d y=%d\n", c_x, c_y);
            g++;
            if(proverka(g)==0)
            {
                truecolor[c_x][c_y].blue=255;
                truecolor[c_x][c_y].green=255;
                truecolor[c_x][c_y].red=255;
            }
            else
            {
                truecolor[c_x][c_y].blue=shade.blue;
                truecolor[c_x][c_y].green=shade.green;
                truecolor[c_x][c_y].red=shade.red;
            }
        }
        for(k=0;k<i+1;k++)
        {
            c_y-=1;
            //printf("x=%d y=%d\n", c_x, c_y);
            g++;
            if(proverka(g)==0)
            {
                truecolor[c_x][c_y].blue=255;
                truecolor[c_x][c_y].green=255;
                truecolor[c_x][c_y].red=255;
            }
            else
            {
                truecolor[c_x][c_y].blue=shade.blue;
                truecolor[c_x][c_y].green=shade.green;
                truecolor[c_x][c_y].red=shade.red;
            }
        }
        for(k=0;k<i+2;k++)
        {
            c_x-=1;
            //printf("x=%d y=%d\n", c_x, c_y);
            g++;
            if(proverka(g)==0)
            {
                truecolor[c_x][c_y].blue=255;
                truecolor[c_x][c_y].green=255;
                truecolor[c_x][c_y].red=255;
            }
            else
            {
                truecolor[c_x][c_y].blue=shade.blue;
                truecolor[c_x][c_y].green=shade.green;
                truecolor[c_x][c_y].red=shade.red;
            }
        }
        for(k=0;k<i+2;k++)
        {
            c_y+=1;
            //printf("x=%d y=%d\n", c_x, c_y);
            g++;
            if(proverka(g)==0)
            {
                truecolor[c_x][c_y].blue=255;
                truecolor[c_x][c_y].green=255;
                truecolor[c_x][c_y].red=255;
            }
            else
            {
                truecolor[c_x][c_y].blue=shade.blue;
                truecolor[c_x][c_y].green=shade.green;
                truecolor[c_x][c_y].red=shade.red;
            }
        }
    }
}
```

```
int proverka(int n)
{
    int x;
    int g=0;
    for(x=2;x<=sqrt(n);x++)
    {
        if(n%x==0)
        {
            g=1;
            break;
            //printf("ne prostoe")
        }
    }
    if(g==1)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```