1000 на 1000
10000 на 10000

	7	8	9	10	
	6	1	2	11	
	5	4	3	12	
			14	13	

```
void paint_vector_segment(int x1, int y1, int x2, int y2, struct point shade)
{
   int v1 = x2 - x1;
   int v2 = y2 - y1;
   double d = sqrt(v1 * v1 + v2 * v2);
   cout << d << endl;
   int param = 5;
   double v1_ed = v1 / (d * param);
   double v2_ed = v2 /(d * param);
   double d_ed = 1 / param;
   double i = 0;
   double j = 0;
   int k = 0;
   while(d_ed < d)
   {
      i += v1_ed;
      j += v2_ed;
      paint_point((int)(i + x1),(int)(j + y1), shade);
      d_ed = sqrt(i * i + j * j);
      k++;
   }
   cout << k;
}
```

```
void paint_segment(int x1, int y1, int x2, int y2, struct point shade)
{
   double dx = x2 - x1;
   double dy = y2 - y1;
   double coef = dy / dx;
   cout << "coef =" << coef << endl;
   int t;
   point red_shade;
   red_shade.red = 255;
   red_shade.blue = 255;
   red_shade.green = 255;
   //cout << "x1 = " << x1  << " " << "y1 = " << y1 << endl;
   //cout << "x2 = " << x2  << " " << "y2 = " << y2 << endl;
   paint_point(x1, y1, red_shade);
   int k = 0;
   if(x1 < x2)
   {
      for(int j = x1 + 1; j < x2; j++)
      {
         //cout << coef * j << endl;
         t = (int)(coef * (j - x1));
         paint_point(j, t + y1, shade);
         //cout << j << " " << t + y1 << endl;
         k++;
      }
   }
   if(x1 > x2)
   {
      for(int j = x1 - 1; j > x2; j--)
      {
         //cout << coef * j << endl;
         t = (int)(coef * j);
         paint_point(j, t + y1, shade);
         cout << j << " " << t + y1 << endl;
         k++;
      }
   }
   cout << k;
   paint_point(x2, y2, red_shade);
}
```

```
void paint_rectangle(int x1, int y1, int x2, int y2, struct point shade)
{
   int maxx, minx, maxy, miny;
   if(x1 > x2)
   {
      maxx = x1;
      minx = x2;
   }
   else
   {
      maxx = x2;
      minx = x1;
   }
   if(y1 > y2)
   {
      maxy = y1;
      miny = y2;
   }
   else
   {
      maxy = y2;
      miny = y1;
   }

   for(int i = minx; i <= maxx; i++)
   {
      for(int j = miny; j <= maxy; j++)
      {

         truecolor[i][j].blue = shade.blue;
         truecolor[i][j].green = shade.green;
         truecolor[i][j].red = shade.red;

      }
   }
}
```

```
#include <iostream>
#include <cmath>
#include <cstdlib>
using namespace std;
#define HEIGHT 1000
#define WIDTH 1000
struct point
{
   char blue;
   char green;
   char red;
};
struct point truecolor[WIDTH][HEIGHT];
char color[HEIGHT*WIDTH*3];
int main()
{
   FILE *ty;
   ty=fopen("test.bmp","rb");
   char mass[54];
   fread(mass, sizeof(char), 54, ty);
   FILE *ty2;
   ty2=fopen("result.bmp","wb");
   fwrite(mass, sizeof(char),54,ty2);
   int i;
   int g;
   for(i=0;i<WIDTH;i++)
   {
      for(g=0;g<HEIGHT;g++)
      {
         truecolor[i][g].blue=128;
         truecolor[i][g].green=128;
         truecolor[i][g].red= 0;
      }
   }
   struct point mypoint;
   mypoint.blue=0;
   mypoint.green=0;
   mypoint.red=0;
  //paint_point(150, 100, mypoint);
  // paint_rectangle(250, 300, 100, 200, mypoint);
  //paint_segment(200, 10, 550, 350, mypoint);
  //paint_vector_segment(200, 10, 550, 350, mypoint);
   int ind_i = WIDTH / 2 - 1;
   //cout << ind_i << endl;
   int ind_j = ind_i;
   point blue_shade;
   blue_shade.red = 0;
   blue_shade.blue = 255;
   blue_shade.green = 0;
   paint_point(ind_i, ind_j, blue_shade);
   int storona = 1,nomer = 2;
   while (nomer <= WIDTH * HEIGHT && nomer < 100)
   {
      paint_spiral( WIDTH, ind_i, ind_j, nomer, storona, -1, 1);
      paint_spiral( WIDTH, ind_i, ind_j, nomer, storona, 1, 1);
      storona++;
      //cout << storona << endl;
      paint_spiral(WIDTH, ind_i, ind_j, nomer, storona, -1, -1);
      paint_spiral(WIDTH, ind_i, ind_j, nomer, storona, 1, -1);
      storona++;
   }
   int a=0;
   for(g=0;g<HEIGHT;g++)
   {
      for(i=0;i<WIDTH;i++)
      {
         color[a]=truecolor[i][g].blue;
         color[a+1]=truecolor[i][g].green;
         color[a+2]=truecolor[i][g].red;
         a+=3;
      }
   }
   fwrite(color,sizeof(char),HEIGHT*WIDTH*3,ty2);
   fclose(ty);
   fclose(ty2);
}
```

```
void paint_spiral(int razmer, int &ind_i, int &ind_j, int &nomer, int storona, int status, int direction)
{
   point white_shade;
   white_shade.red = 255;
   white_shade.blue = 255;
   white_shade.green = 255;
   point red_shade;
   red_shade.red = 255;
   red_shade.blue = 0;
   red_shade.green = 0;
   int shag = 0;
   while (shag < storona && nomer <= razmer * razmer)
   {
      if (status == 1)
      {
         ind_i += direction;
      }
      else
      {
         ind_j += direction;
      }
      cout << "nomer=" << nomer << " ";
      if(prostoe_chislo(nomer) == true)
      {
         paint_point(ind_i, ind_j, white_shade);
         cout << "w" << endl;
         //matrix[ind_i][ind_j] = 0;
      }
      else
      {
         paint_point(ind_i, ind_j, red_shade);
         cout << "r" << endl;
      }
      nomer++;
      shag++;
   }
}

bool prostoe_chislo(int n)
{
   int status = 0;
   for(int i = 2; i <=sqrt(n); i++)
   {
      if(n%i == 0)
      {
         return false;
      }
   }
   if(status == 0)
   {
      return true;
   }
}

void paint_point(int x, int y, struct point shade)
{
   truecolor[x][y].blue=shade.blue;
   truecolor[x][y].green=shade.green;
   truecolor[x][y].red=shade.red;
}
```