```c
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#define round(a) ((int)(a + 0.5))

int k;
float clipWindowXMin, clipWindowYMin, clipWindowXMax, clipWindowYMax, arr[20], slope;

void clipLeft(float x1, float y1, float x2, float y2) {
    if (x2 - x1) slope = (y2 - y1) / (x2 - x1);
    if (x1 >= clipWindowXMin && x2 >= clipWindowXMin) {
        arr[k] = x2;
        arr[k + 1] = y2;
        k += 2;
    }
    if (x1 < clipWindowXMin && x2 >= clipWindowXMin) {
        arr[k] = clipWindowXMin;
        arr[k + 1] = y1 + slope * (clipWindowXMin - x1);
        arr[k + 2] = x2;
        arr[k + 3] = y2;
        k += 4;
    }
    if (x1 >= clipWindowXMin && x2 < clipWindowXMin) {
        arr[k] = clipWindowXMin;
        arr[k + 1] = y1 + slope * (clipWindowXMin - x1);
        k += 2;
    }
}

void clipTop(float x1, float y1, float x2, float y2) {
    if (y2 - y1) slope = (x2 - x1) / (y2 - y1);
    if (y1 <= clipWindowYMax && y2 <= clipWindowYMax) {
        arr[k] = x2;
        arr[k + 1] = y2;
        k += 2;
    }
    if (y1 > clipWindowYMax && y2 <= clipWindowYMax) {
        arr[k] = x1 + slope * (clipWindowYMax - y1);
        arr[k + 1] = clipWindowYMax;
        arr[k + 2] = x2;
        arr[k + 3] = y2;
        k += 4;
```

```
42          k += 4;
43      }
44      if (y1 <= clipWindowYMax && y2 > clipWindowYMax) {
45          arr[k] = x1 + slope * (clipWindowYMax - y1);
46          arr[k + 1] = clipWindowYMax;
47          k += 2;
48      }
49  }
50
51  void clipRight(float x1, float y1, float x2, float y2) {
52      if (x2 - x1) slope = (y2 - y1) / (x2 - x1);
53      if (x1 <= clipWindowXMax && x2 <= clipWindowXMax) {
54          arr[k] = x2;
55          arr[k + 1] = y2;
56          k += 2;
57      }
58      if (x1 > clipWindowXMax && x2 <= clipWindowXMax) {
59          arr[k] = clipWindowXMax;
60          arr[k + 1] = y1 + slope * (clipWindowXMax - x1);
61          arr[k + 2] = x2;
62          arr[k + 3] = y2;
63          k += 4;
64      }
65      if (x1 <= clipWindowXMax && x2 > clipWindowXMax) {
66          arr[k] = clipWindowXMax;
67          arr[k + 1] = y1 + slope * (clipWindowXMax - x1);
68          k += 2;
69      }
70  }
71
72  void clipBottom(float x1, float y1, float x2, float y2) {
73      if (y2 - y1) slope = (x2 - x1) / (y2 - y1);
74      if (y1 >= clipWindowYMin && y2 >= clipWindowYMin) {
75          arr[k] = x2;
76          arr[k + 1] = y2;
77          k += 2;
78      }
79      if (y1 < clipWindowYMin && y2 >= clipWindowYMin) {
80          arr[k] = x1 + slope * (clipWindowYMin - y1);
81          arr[k + 1] = clipWindowYMin;
82          arr[k + 2] = x2;
83          arr[k + 3] = y2;
```

```
        arr[k + 3] = y2;
        k += 4;
    }
    if (y1 >= clipWindowYMin && y2 < clipWindowYMin) {
        arr[k] = x1 + slope * (clipWindowYMin - y1);
        arr[k + 1] = clipWindowYMin;
        k += 2;
    }
}

void main() {
    int graphicsDriver = DETECT, graphicsMode, numberOfSides, polygon[20], i;
    float startX, startY, endX, endY, polygonCoordinates[20];

    clrscr();
    printf("Coordinates of rectangular clip window:\nxmin, ymin: ");
    scanf("%f %f", &clipWindowXMin, &clipWindowYMin);
    printf("Coordinates of rectangular clip window:\nxmax, ymax: ");
    scanf("%f %f", &clipWindowXMax, &clipWindowYMax);

    printf("\n\nPolygon to be clipped:\nNumber of sides: ");
    scanf("%d", &numberOfSides);
    printf("Enter the coordinates: ");
    for (i = 0; i < 2 * numberOfSides; i++)
        scanf("%f", &polygonCoordinates[i]);

    polygonCoordinates[i] = polygonCoordinates[0];
    polygonCoordinates[i + 1] = polygonCoordinates[1];

    for (i = 0; i < 2 * numberOfSides + 2; i++)
        polygon[i] = round(polygonCoordinates[i]);

    initgraph(&graphicsDriver, &graphicsMode, "C:\\TC\\BGI");
    rectangle(clipWindowXMin, clipWindowYMax, clipWindowXMax, clipWindowYMin);

    printf("\tUNCLIPPED POLYGON");
    fillpoly(numberOfSides, polygon);
    getch();
    cleardevice();

    k = 0;
    for (i = 0; i < 2 * numberOfSides; i += 2)
```

```
124     for (i = 0; i < 2 * numberOfSides; i += 2)
125         clipLeft(polygonCoordinates[i], polygonCoordinates[i + 1], polygonCoordinates[i + 2], polygonCoordinates[i + 3]);
126
127     numberOfSides = k / 2;
128     for (i = 0; i < k; i++)
129         polygonCoordinates[i] = arr[i];
130
131     polygonCoordinates[i] = polygonCoordinates[0];
132     polygonCoordinates[i + 1] = polygonCoordinates[1];
133     k = 0;
134
135     for (i = 0; i < 2 * numberOfSides; i += 2)
136         clipTop(polygonCoordinates[i], polygonCoordinates[i + 1], polygonCoordinates[i + 2], polygonCoordinates[i + 3]);
137
138     numberOfSides = k / 2;
139     for (i = 0; i < k; i++)
140         polygonCoordinates[i] = arr[i];
141
142     polygonCoordinates[i] = polygonCoordinates[0];
143     polygonCoordinates[i + 1] = polygonCoordinates[1];
144     k = 0;
145
146     for (i = 0; i < 2 * numberOfSides; i += 2)
147         clipRight(polygonCoordinates[i], polygonCoordinates[i + 1], polygonCoordinates[i + 2], polygonCoordinates[i + 3]);
148
149     numberOfSides = k / 2;
150     for (i = 0; i < k; i++)
151         polygonCoordinates[i] = arr[i];
152
153     polygonCoordinates[i] = polygonCoordinates[0];
154     polygonCoordinates[i + 1] = polygonCoordinates[1];
155     k = 0;
156
157     for (i = 0; i < 2 * numberOfSides; i += 2)
158         clipBottom(polygonCoordinates[i], polygonCoordinates[i + 1], polygonCoordinates[i + 2], polygonCoordinates[i + 3]);
159
160     for (i = 0; i < k; i++)
161         polygon[i] = round(arr[i]);
162
163     if (k) {
164         fillpoly(k / 2, polygon);
165     }
```

```
157         for (i = 0; i < 2 * numberOfSides; i += 2)
158             clipBottom(polygonCoordinates[i], polygonCoordinates[i + 1], polygonCoordinates[i + 2], polygonCoordinates[i + 3]);
159
160         for (i = 0; i < k; i++)
161             polygon[i] = round(arr[i]);
162
163         if (k) {
164             fillpoly(k / 2, polygon);
165         }
166
167         rectangle(clipWindowXMin, clipWindowYMax, clipWindowXMax, clipWindowYMin);
168         printf("\tCLIPPED POLYGON");
169         getch();
170         closegraph();
171     }
```