```
// C++ program for point clipping Algorithm
#include <bits/stdc++.h>
using namespace std;

// Function for point clipping
void pointClip(int XY[][2], int n, int Xmin, int Ymin,
                                int Xmax, int Ymax)
{
    /*************** Code for graphics view
    // initialize graphics mode
    detectgraph(&gm,&gr);
    initgraph(&gm,&gr,"d:\\tc\\BGI");
    for (int i=0; i<n; i++)
    {
    if ( (XY[i][0] >= Xmin) && (XY[i][0] <= Xmax))
    {
            if ( (XY[i][1] >= Ymin) && (XY[i][1] <= Ymax))
        putpixel(XY[i][0],XY[i][1],3);
    }
    }
    **********************/
    /**** Arithmetic view ****/
    cout << "Point inside the viewing pane:" << endl;
    for (int i = 0; i < n; i++)
    {
        if ((XY[i][0] >= Xmin) && (XY[i][0] <= Xmax))
        {
            if ((XY[i][1] >= Ymin) && (XY[i][1] <= Ymax))
                cout <<"[" << XY[i][0] <<","<<XY[i][1]<<"] ";
        }
    }

    // print point coordinate outside viewing pane
    cout<<"\n"<< endl;
    cout << "Point outside the viewing pane:"<<endl;
    for (int i = 0; i < n; i++)
    {
        if ((XY[i][0] < Xmin) || (XY[i][0] > Xmax))
            cout << "[" << XY[i][0] << "," << XY[i][1] << "] ";
        if ((XY[i][1] < Ymin) || (XY[i][1] > Ymax))
            cout << "[" << XY[i][0] << "," << XY[i][1] << "] ";
    }
}

// Driver code
int main()
{
    int XY[6][2] = {{10, 10}, {-10, 10}, {400, 100},
                    {100, 400}, {400, 400}, {100, 40}};

    // getmaxx() & getmaxy() will return Xmax, Ymax
    // value if graphics.h is included
    int Xmin = 0;
    int Xmax = 350;
    int Ymin = 0;
    int Ymax = 350;
    pointClip(XY, 6, Xmin, Ymin, Xmax, Ymax);
    return 0;
}
```