Assignment No. 4

Name – Manish Namdev Barage

PRN – 22520007

Batch – T7

Problem Statement – To Implement Convex hull problem using divide and conquer method

Code –

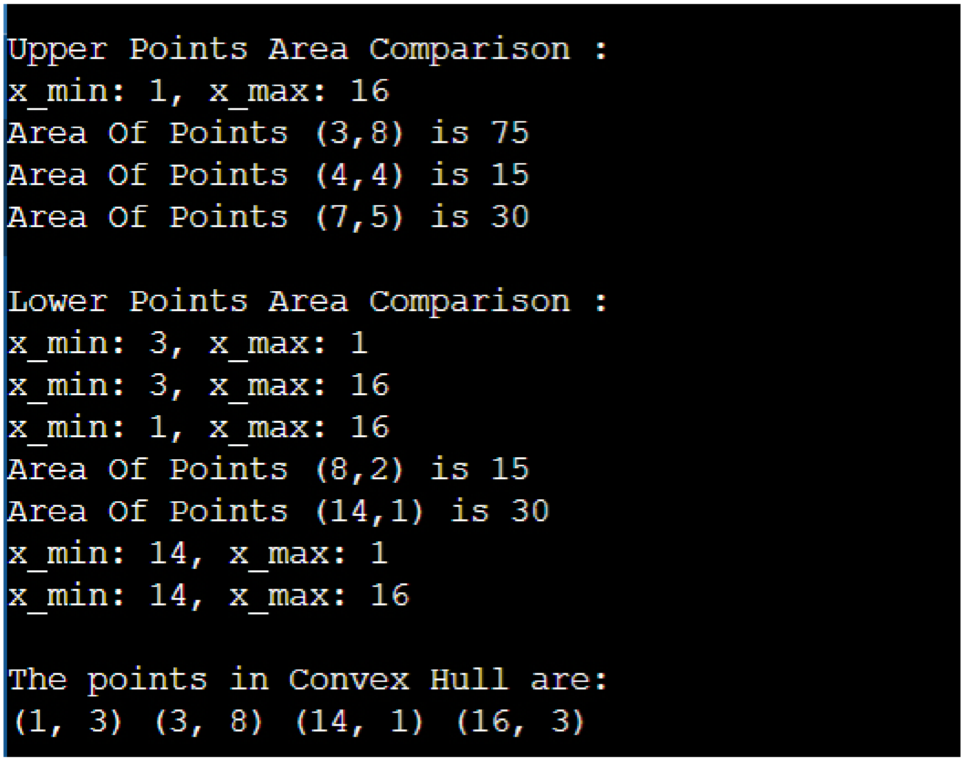
```
#include<bits/stdc++.h>
using namespace std;


set<pair<int, int>> hull;


bool flaga=true,flagb=true;
// Checking if Point P Line Ke Upar hein Ya Niche
int findSide(pair<int, int> p1, pair<int, int> p2, pair<int, int> p)
{
        int val = (p.second - p1.second) * (p2.first - p1.first) -
                        (p2.second - p1.second) * (p.first - p1.first);


        if (val > 0)
                return 1;
        if (val < 0)
                return -1;
        return 0;
}


// Finding point Having Maximum Area
int lineDist(pair<int, int> p1, pair<int, int> p2, pair<int, int> p)
{
        return abs ((p.second - p1.second) * (p2.first - p1.first) -
                        (p2.second - p1.second) * (p.first - p1.first));
```

```
}



void quickHull(pair<int, int> a[], int n, pair<int, int> p1, pair<int, int> p2, int side)
{
        int ind = -1;
        int max_dist = 0;

        // Max Point Choose Karra hu
   if(side==1 && flaga){
      cout<<endl<<"Upper Points Area Comparison :"<<endl;
      flaga=false;
   }
   if(side==-1 && flagb){
      cout<<endl<<"Lower Points Area Comparison :"<<endl;
      flagb=false;
   }

        // Print x_min and x_max in each iteration
   cout << "x_min: " << p1.first << ", x_max: " << p2.first << endl;

        for (int i=0; i<n; i++)
        {
                int temp = lineDist(p1, p2, a[i]);
      if(findSide(p1, p2, a[i]) == side){
         cout<<"Area Of Points "<<"("<<a[i].first<<","<<a[i].second<<") is "<<temp<<endl;
      }

                if (findSide(p1, p2, a[i]) == side && temp > max_dist)
                {
                        ind = i;
```

```
            max_dist = temp;
        }
    }

    // Agar koi point nhi mila toh p1,p2 daaldo
    if (ind == -1)
    {
        hull.insert(p1);
        hull.insert(p2);
        return;
    }

    // Recur for the two parts divided by a[ind]
    quickHull(a, n, a[ind], p1, -findSide(a[ind], p1, p2));
    quickHull(a, n, a[ind], p2, -findSide(a[ind], p2, p1));
}

void printHull(pair<int, int> a[], int n)
{
    // Base Case
    if (n < 3){
        cout << "Convex hull not possible\n";
        return ;
    }

    // Finding (minx,miny) and (maxx,maxy)
    int min_x = 0, max_x = 0;
    for (int i=1; i<n; i++){
        if (a[i].first < a[min_x].first)
            min_x = i;
        if (a[i].first > a[max_x].first)
```

```
            max_x = i;
    }

    // Finding Hull Points On Upper Side
    quickHull(a, n, a[min_x], a[max_x], 1);

    // Finding Hull Points On Lower Side
    quickHull(a, n, a[min_x], a[max_x], -1);



    cout << "\nThe points in Convex Hull are:\n";
    while (!hull.empty())
    {
        cout << "(" <<( *hull.begin()).first << ", "
            << (*hull.begin()).second << ") ";
        hull.erase(hull.begin());
    }
}

int main()
{
    pair<int, int> a[10] = {{1, 3}, {3, 8}, {4, 4}, {7, 5},{8, 2}, {8, 3}, {13, 3}, {14, 1} , {14,
3} , {16, 3}};
    int n = 10;
    printHull(a, n);
    return 0;
}
```

Output –



```
Upper Points Area Comparison :
x_min: 1, x_max: 16
Area Of Points (3,8) is 75
Area Of Points (4,4) is 15
Area Of Points (7,5) is 30

Lower Points Area Comparison :
x_min: 3, x_max: 1
x_min: 3, x_max: 16
x_min: 1, x_max: 16
Area Of Points (8,2) is 15
Area Of Points (14,1) is 30
x_min: 14, x_max: 1
x_min: 14, x_max: 16

The points in Convex Hull are:
(1, 3) (3, 8) (14, 1) (16, 3)
```