```
/******  Monotone Convex Hull Algo by Nafee *****/
struct point
{
    long long x, y;
    point(){}
    point(long long x, long long y)
    {
        this->x = x;
        this->y = y;
    }
    point operator - (point B)
    {
        point ret;
        ret.x = x - B.x;
        ret.y = y - B.y;
        return ret;
    }
    const bool operator < (point B) const
    {
        if ( x == B.x )
        {
            return y < B.y;
        }
        return x < B.x;
    }
};
bool cmpPoint(point A, point B)
{
    if ( A.x == B.x )
    {
        return A.y < B.y;
    }
    return A.x < B.x;
}
long long crossPr(point A, point B)
{
    long long ret = A.x*B.y - A.y*B.x;
    return ret;
}
vector<point> findConHull(vector<point> givenPoints)
{
    long long a, b, c, d, e, f, t, len = givenPoints.size(), siz;
    vector<point> ret;
    sort(givenPoints.begin(), givenPoints.end());
    for (a = 0;a < len; a++)
    {
        siz = ret.size();
        while( siz >= 2 && crossPr(ret[siz-1]-ret[siz-2], givenPoints[a]-ret[siz-2]) <= 0 )
        {
            ret.pop_back();
            siz = ret.size();
        }
        ret.push_back( givenPoints[a] );
    }
    ret.pop_back();
    t = ret.size();
    len = givenPoints.size();
    for (a = len-1; a >= 0; a--)
    {
        siz = ret.size();
        while( siz >= 2+t && crossPr(ret[siz-1]-ret[siz-2], givenPoints[a]-ret[siz-2]) <= 0 )
        {
            ret.pop_back();
            siz = ret.size();
        }
        ret.push_back( givenPoints[a] );
    }
    ret.pop_back();
    return ret;
}
```