```
#include <stdio.h>

int N;

struct point{
    int x, y;
};

int min(int a, int b){
    return a < b ? a : b;
}

int max(int a, int b){
    return a > b ? a : b;
}

// Does this point exist in the list?
int check(struct point a, struct point *P){
    for(int i=0; i<N; i++){
        if(a.x == P[i].x && a.y == P[i].y){
            return 1;
        }
    }
    return 0;
}

int main(){
    scanf("%d", &N);

    struct point P[N];
    for(int i = 0 ; i < N; i++)
        scanf("%d %d", &(P[i].x), &(P[i].y));

    int MAX_VAL = -1;
    int area = MAX_VAL;

    // Take every pair of points and try to construct a rectangle
    // With those two points as the diagonal points of the rectangle
    // To see if this can be done, see if the other two diagonal points
    // are present in the list or not
    for(int i = 0; i < N; i++){
        for(int j = i+1; j < N; j++){

            // The four corners of the rectangle formed out of
            // the two points P[i] and P[j]
            struct point a, b, c, d;

            a.x = min(P[i].x, P[j].x);
            a.y = min(P[i].y, P[j].y);

            b.x = min(P[i].x, P[j].x);
            b.y = max(P[i].y, P[j].y);

            c.x = max(P[i].x, P[j].x);
            c.y = min(P[i].y, P[j].y);

            d.x = max(P[i].x, P[j].x);
            d.y = max(P[i].y, P[j].y);

            // Good - all four points are available
            if(check(a, P) && check(b, P) && check(c, P) && check(d, P)){
                int X = max(P[i].x, P[j].x) - min(P[i].x, P[j].x);
                int Y = max(P[i].y, P[j].y) - min(P[i].y, P[j].y);
                if(X*Y > 0)
                    area = max(area, X*Y);
            }
        }
    }
    printf("%d\n", area);
}
```