

CONVEX HULL

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct Point
```

```
{
```

```
    int x, y;
```

```
};
```

```
int orientation(struct Point p, struct Point q, struct Point r)
```

```
{
```

```
    int val = (q.y - p.y) * (r.x - q.x) - (q.x - p.x) * (r.y - q.y);
```

```
    if (val == 0)
```

```
    {
```

```
        return 0;
```

```
    }
```

```
    return (val > 0) ? 1 : 2;
```

```
}
```

```
void convexHull(struct Point points[], int n)
```

```
{
```

```
    if (n < 3)
```

```
    {
```

```
        return;
```

```
    }
```

```
    struct Point *hull = malloc(n * sizeof(struct Point));
```

```
    int hull_size = 0;
```

```
    int l = 0;
```

```
    for (int i = 1; i < n; i++)
```

```

{
    if (points[i].x < points[l].x)
    {
        l = i;
    }
}

int p = l, q;

do
{
    hull[hull_size++] = points[p];

    q = (p + 1) % n;
    for (int i = 0; i < n; i++)
    {
        if (orientation(points[p], points[i], points[q]) == 2)
        {
            q = i;
        }
    }

    p = q;
} while (p != l);

for (int i = 0; i < hull_size; i++)
{
    printf("(%d, %d)\n", hull[i].x, hull[i].y);
}

free(hull);
}

int main()
{

```

```

int n;

printf("Enter the number of points: ");

scanf("%d", &n);

struct Point *points = (struct Point *)malloc(n * sizeof(struct Point));

printf("Enter the coordinates of each point (x, y):\n");

for (int i = 0; i < n; i++)
{
    printf("Point %d: ", i + 1);

    scanf("%d %d", &points[i].x, &points[i].y);
}

convexHull(points, n);

return 0;
}

```

OUTPUT

```

Enter the number of points: 6
Enter the coordinates of each point (x, y):
Point 1: 0
0
Point 2: 0
4
Point 3: -4
0
Point 4: 5
0
Point 5: 0
-6
Point 6: 1
0
(-4, 0)
(0, -6)
(5, 0)
(0, 4)

```