

## UNIT – 6

### 1. Ant on a Chessboard

Program:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  #define FOI(i, A, B) for(i=A; i<=B; i++)
4  #define FOD(i, A, B) for(i=A; i>=B; i--)
5  #define PI      acos(-1.0)
6  #define INF      1<<30
7  #define EPS      1e-9
8  #define sqr(x) (x)*(x)
9  const int maxn = 3E5 + 5;
10 const int mod = 1E9 + 7;
11
12 typedef unsigned int uint;
13 typedef long long int64;
14 typedef unsigned long long uint64;
15
16 int main(){
17     while (true) {
18         int t, x, y, d;
19         scanf("%d", &t);
20         if (t == 0) break;
21         x = y = (int) ceil(sqrt(t));
22         d = (x%2 ? 1 : -1) * (t - (int)(sqr(x-1) + x));
23         x -= max(d, 0);
24         y += min(d, 0);
25         printf("%d %d\n", x, y);
26     }
27     return 0;
28 }
29
30
31 }
```

Output:

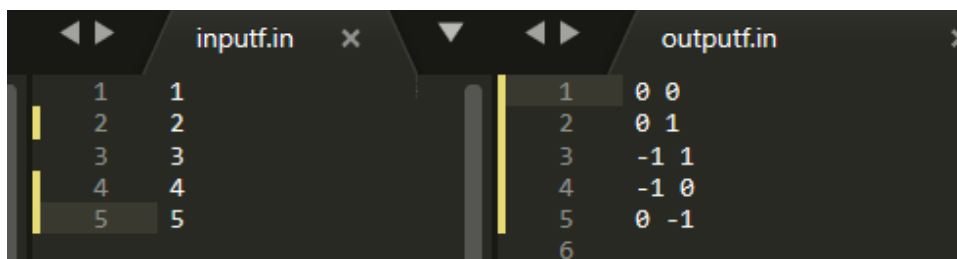
inputf.in		outputf.in	
1	8	1	2 3
2	20	2	5 4
3	25	3	1 5
4	0	4	

### 2. Bee Maja

## Program:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4
5  int numNodesByOX[10000];
6
7  int xChange[] = {-1, 0, 1, 1, 0, 0, -1};
8  int yChange[] = {0, -1, -1, 0, 1, 1, 1};
9
10 int main()
11 {
12     int sum = 1, current = 1;
13     int yMax = 1;
14     for (; sum <= 10000; ++yMax, sum += current, current += 6)
15     {
16         numNodesByOX[yMax] = sum;
17     }
18
19     int num;
20     while (cin >> num)
21     {
22         int *position = upper_bound(numNodesByOX, numNodesByOX + yMax, num);
23         --position;
24
25         int xCircleVal = 0;
26         int yCircleVal = position - numNodesByOX - 1;
27         const int sideLength = yCircleVal;
28
29
30         int distanceLeft = num - *position;
31         for (int change = 0; distanceLeft; ++change)
32         {
33             int move = min(distanceLeft, sideLength);
34             if (change == 5)
35                 move = min(move, 1);
36             xCircleVal += xChange[change] * move;
37             yCircleVal += yChange[change] * move;
38             distanceLeft -= move;
39         }
40         cout << xCircleVal << ' ' << yCircleVal << '\n';
41     }
42 }
```

## Output:



The screenshot shows a terminal window with two panes. The left pane, titled 'inputf.in', contains a list of numbers from 1 to 5. The right pane, titled 'outputf.in', contains the corresponding output coordinates for each input number. The output for input 1 is '0 0', for 2 is '0 1', for 3 is '-1 1', for 4 is '-1 0', and for 5 is '0 -1'. There is a line number 6 at the bottom of the right pane, but no output is shown for it.

Input	Output (x y)
1	0 0
2	0 1
3	-1 1
4	-1 0
5	0 -1