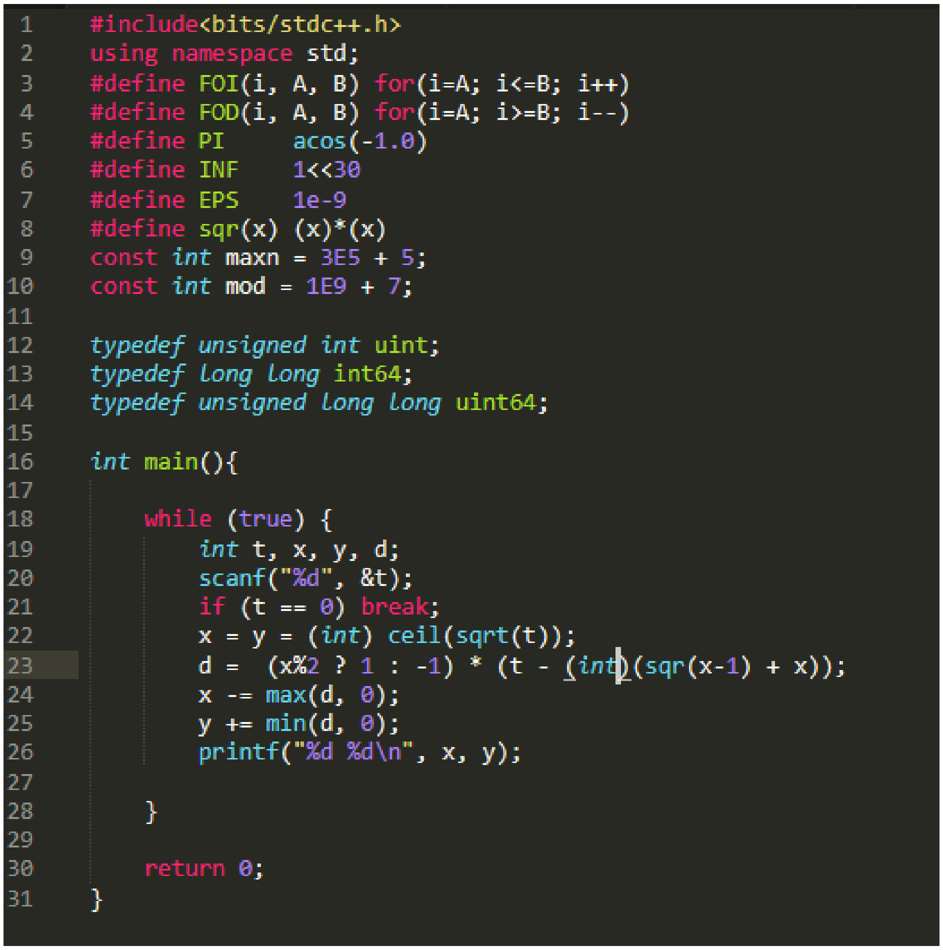
UNIT – 6

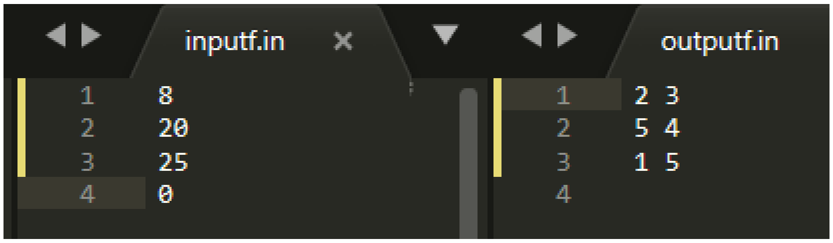
1. Ant on a Chessboard

Program:



```cpp
#include<bits/stdc++.h>
using namespace std;
#define FOI(i, A, B) for(i=A; i<=B; i++)
#define FOD(i, A, B) for(i=A; i>=B; i--)
#define PI     acos(-1.0)
#define INF    1<<30
#define EPS    1e-9
#define sqr(x) (x)*(x)
const int maxn = 3E5 + 5;
const int mod = 1E9 + 7;

typedef unsigned int uint;
typedef long long int64;
typedef unsigned long long uint64;

int main(){

    while (true) {
        int t, x, y, d;
        scanf("%d", &t);
        if (t == 0) break;
        x = y = (int) ceil(sqrt(t));
        d =  (x%2 ? 1 : -1) * (t - (int)(sqr(x-1) + x));
        x -= max(d, 0);
        y += min(d, 0);
        printf("%d %d\n", x, y);

    }

    return 0;
}
```

Output:

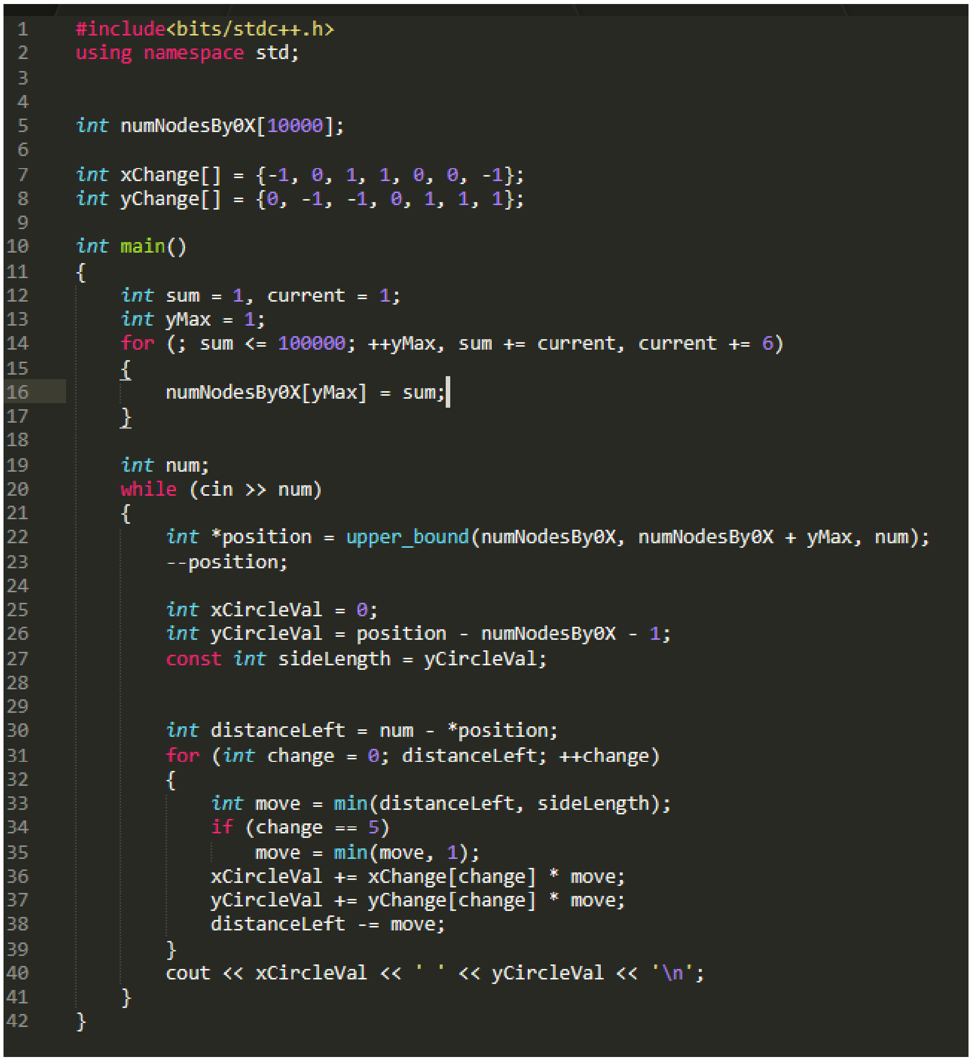


```
8
20
25
0
```

```
2 3
5 4
1 5
```

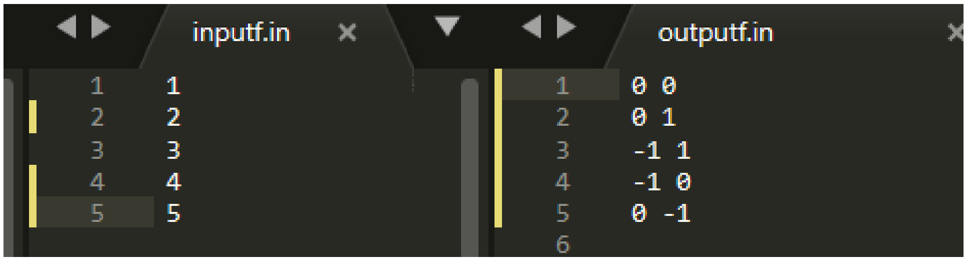
2. Bee Maja

Program:



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


int numNodesBy0X[10000];

int xChange[] = {-1, 0, 1, 1, 0, 0, -1};
int yChange[] = {0, -1, -1, 0, 1, 1, 1};

int main()
{
    int sum = 1, current = 1;
    int yMax = 1;
    for (; sum <= 100000; ++yMax, sum += current, current += 6)
    {
        numNodesBy0X[yMax] = sum;
    }

    int num;
    while (cin >> num)
    {
        int *position = upper_bound(numNodesBy0X, numNodesBy0X + yMax, num);
        --position;

        int xCircleVal = 0;
        int yCircleVal = position - numNodesBy0X - 1;
        const int sideLength = yCircleVal;


        int distanceLeft = num - *position;
        for (int change = 0; distanceLeft; ++change)
        {
            int move = min(distanceLeft, sideLength);
            if (change == 5)
                move = min(move, 1);
            xCircleVal += xChange[change] * move;
            yCircleVal += yChange[change] * move;
            distanceLeft -= move;
        }
        cout << xCircleVal << ' ' << yCircleVal << '\n';
    }
}
```

Output:



```
1
2
3
4
5
```

```
0 0
0 1
-1 1
-1 0
0 -1
```