Source Code:

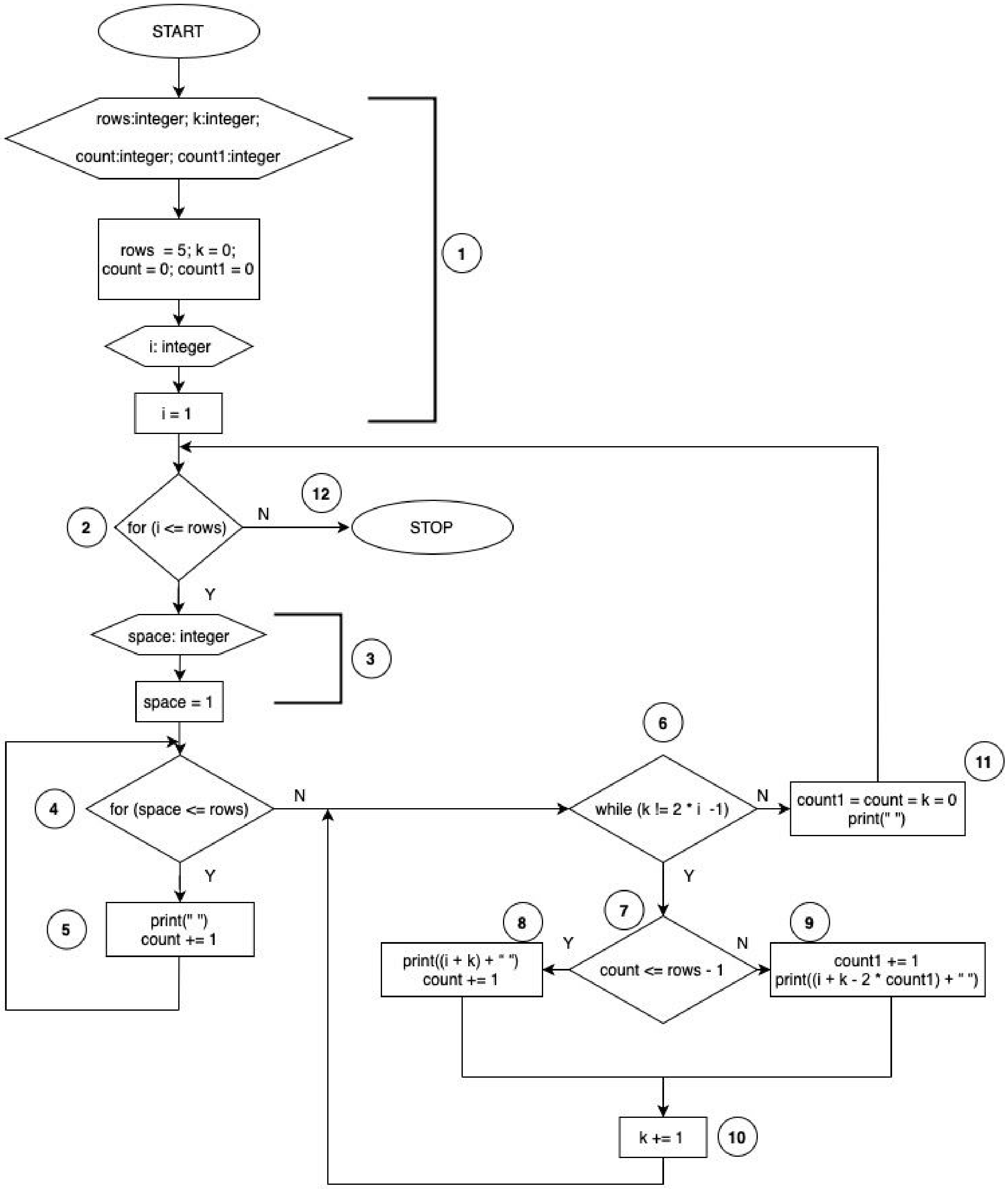
```
public class Pattern {
    public static void main(String[] args) {
        int rows = 5, k = 0, count = 0, count1 = 0;
        for(int i = 1; i <= rows; ++i) {
            for(int space = 1; space <= rows - i; ++space) {
                System.out.print(" ");
                ++count;
            }
            while(k != 2 * i - 1) {
                if (count <= rows - 1) {
                    System.out.print((i + k) + " ");
                    ++count;
                 }
                else {
                    ++count1;
                    System.out.print((i + k - 2 * count1) + " ");
                }
                ++k;
            }
            count1 = count = k = 0;
            System.out.println();
        }
    }
}
```

Start

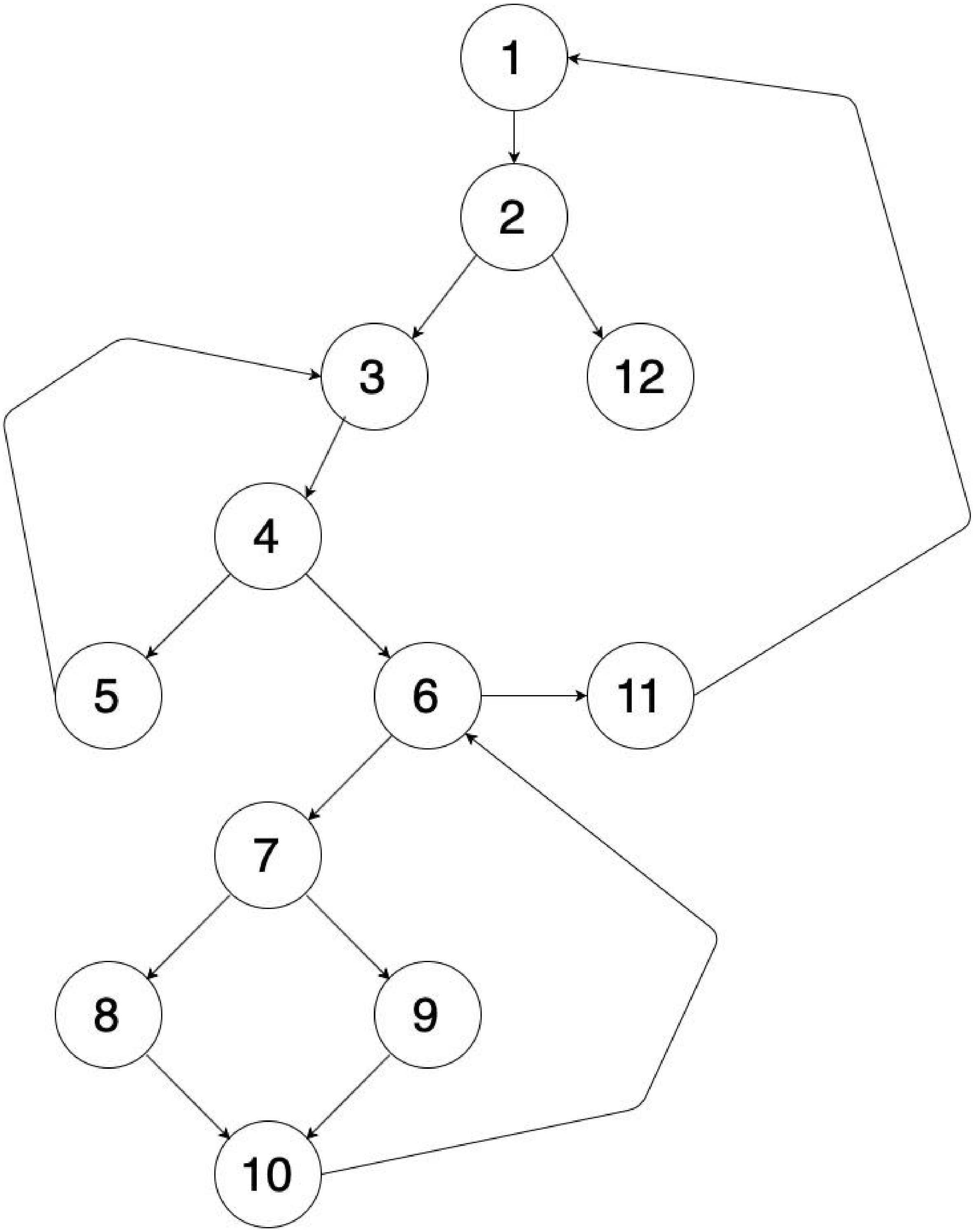
```
1.  rows : integer, k : integer, count : integer, count1 : integer
2.  rows = 5, k = 0, count = 0, count1 = 0
3.  i : integer
4.  i = 1
5.  for i <= rows
6.  space : integer
7.  Space = 1

8.  for space <= rows
9.  print “ “
10. count + 1
    (Balik ke 9)
11. while k != 2 * i - 1
12. If count <= rows - 1
13. print (i + k) + “ “
14. Count + 1
15. Else
16. Count1 + 1
17. Print (i + k - 2 * count1) + “ “
18. k + 1
    (Balik ke 12)
19. count1 = count = k = 0
20. Print “ “
    (Balik ke 6)
```

FLOWCHART


START
rows:integer; k:integer;
count:integer; count1:integer
rows = 5; k = 0;
count = 0; count1 = 0
i: integer
i = 1
1
12
2
for (i <= rows)
N
STOP
Y
space: integer
space = 1
3
6
11
4
for (space <= rows)
N
while (k != 2 * i -1)
N
count1 = count = k = 0
print(" ")
Y
Y
5
print(" ")
count += 1
7
8
9
Y
N
print((i + k) + " ")
count += 1
count <= rows - 1
count1 += 1
print((i + k - 2 * count1) + " ")
k += 1
10

BASIS PATH



R1 = 3, 4, 5
R2 = 7, 8, 9, 10
R3 = 6, 7, 9, 10
R4 = 1, 2, 3, 4, 6, 11
R5 = diluar graph
Region ada 5

Cyclomatic complexity

V(G) = jumlah region
= 5

Independent Path

Path 1 : 1 - 2 - 12
Path 2 : 1 - 2 - 3 - 4 - 6 - 11 - 1 - 2 - 12
Path 3 : 1 - 2 - 3 - 4 - 6 - 7 - 8 - 10 - 6 - 11 - 1 - 2 - 12
Path 5 : 1 - 2 - 3 - 4 - 6 - 7 - 9 - 10 - 6 - 11 - 1 - 2 - 12

Test Case

Path 1 : rows = -1 and rows = 5
Path 2 : k = -1 and k = 0
Path 3 : count = 5 and count = 0
Path 4 : count = 0 and count = 5

Source Code JavaScript

```
function pattern(){
  var rows = 5, k = 0, count = 0, count1 = 0; //1
  for (var i = 1; i <= rows; ++i) { //2
    {
      for (var space = 1; space <= rows - i; ++space) { //3, 4
        {
          console.info(" "); //5
          ++count;
        };
      }
      while ((k !== 2 * i - 1)) { //6
        {
          if (count <= rows - 1) { //7
            console.info((i + k) + " "); //8
            ++count;
          }
          else { //9
            ++count1;
            console.info((i + k - 2 * count1) + " ");
          }
          ++k; //10
        }
      };
      count1 = count = k = 0; //11
      console.info();
    };
  };
  return 0;
}
```