luckyNumbers.java

```java
// In number theory, a lucky number is a natural number in a set which is generated by a certain "sieve".
// This sieve is similar to the Sieve of Eratosthenes that generates the primes, but it eliminates numbers based on their position in the remaining set,
// instead of their value (or position in the initial set of natural numbers).
// Starting from 2, start eliminating numbers till the count is big enough that all remaining numbers stay

import java.util.Scanner;

public class luckyNumbers {
    int N;
    int[] arr;

    public static void main(String[] args) {
        luckyNumbers obj = new luckyNumbers();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter value of N");
        obj.N = sc.nextInt();
        sc.close();
        obj.arr = new int[obj.N];
        for (int i = 0; i < obj.N; i++) {
            obj.arr[i] = i + 1;
        }
        obj.print();
        int steps = obj.numberOfSteps();
        for (int i = 1; i <= steps; i++) {
            obj.eliminate(++i);
        }
        System.out.println("Lucky numbers: ");
        obj.print();
    }

    void eliminate(int count) {
        int a = 0;
        for (int i = 0; i < N; i++) {
            if (arr[i] == 0) continue;
            if ((a + 1) % count == 0) arr[a] = 0;
            a += 1;
        }
    }

    int numberOfSteps() {
        // Find number of steps of elimination required to find lucky numbers for a given value of N
        int N = this.N;
        int steps = 0;
        int i = 2;
        while (i <= N) {
            int R = (int) (Math.floor(N / (i + 0.0)));
            N -= R;
            i += 1;
            steps += 1;
        }
        return steps;
    }

    void print() {
        for (int i = 0; i < N; i++) {
            if (arr[i] != 0) System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
}
```