So let’s discuss a delicious problem today! We are given a number ‘x’, we want to reduce it to ‘1’ in minimum number of steps possible!!

At each step, we are allowed to :

1)Decrement the given number by ‘1’……..(i)

2)Decrement the given number by ‘2’ ………..(ii)

3)Divide the given number by ‘7’ only if it is divisible by ‘7’……..(iii)

So how do we solve this problem(find the minimum number of steps to reduce any given ‘x’ to ‘1’) ?

Observation :

* Operation for 1 to reach 1 : Zero 🡪 dp[1] = 0
* Operation for 2 toreach 1 : One 🡪 dp[2] = 1
* Operation for 3 to reach 1 : one 🡪 3- 2

Now for 4 to x , we can calculate easily

Dp[4] = min (dp[3] , dp[2]) + 1

But when the number is divisible by 7 , then min (dp[i/7] , dp[I – 1] , dp[I – 2]) + 1

import java.util.\*;

public class MinStepsToOne {

public static int mini(int a, int b, int c) {

return Math.min(a, Math.min(b, c));

}

public static int solve(int x) {

int[] dp = new int[x + 1];

dp[1] = 0;

dp[2] = 1;

int i = 3;

while (i <= x) {

if (i % 7 == 0) {

dp[i] = mini(dp[i - 1] + 1, dp[i - 2] + 1, dp[i / 7] + 1);

} else {

dp[i] = Math.min(dp[i - 1] + 1, dp[i - 2] + 1);

}

i++;

}

return dp[x];

}

}