**Question :**

You re given a number , reduce it to 1 with minimum cost

In one operation you can do

* - 1 which cost “A”
* If it s divisible by 3 , then , it cost is “B”
* If it is divisible by 5 , then , it cost is “C”
* If it is divisible by 7 , then it cost s “D”

**Observation :**

* 1 to reach 1 is 0 , so dp[1] = 0
* Find the minimum of cost required to reduce a number “i” to 1 .
* Consider all cases , /3 ,/5 , /7 and for 1 it will be dp[ i - 1] + A

**Recurrence Relation :**

* dp[i] = min (dp[i - 1] + A , dp[i / 3] + B , dp[ i / 5] + C , dp[ i / 7] + D)

**Code :**

class Solution {

private int min(int a , int b , int c , int d){

if(a < b && a < c && a < d){

return a;

}

if(b < c && b < d){

return b;

}

if(c < d){

return c;

}

return d;

}

public int minOperation(int n , int A , int B ,int C , int D) {

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

dp[1] = 0;

for(int i = 2 ; <= n ; i++){

int v1 = dp[i - 1] + A;

int v2 , v3 , v4;

v2 = v3 = v4 = Integer.MAX\_VALUE;

if(i % 3 == 0){

v2 = dp[i / 3] + B;

}

if(i % 5 == 0){

v3 = dp[i / 5] + C;

}

if(i % 7 == 0){

v4 = dp[i / 7] + D;

}

dp[i] = min(v1 , v2 , v3 , v4);

}

return dp[n];

}

}