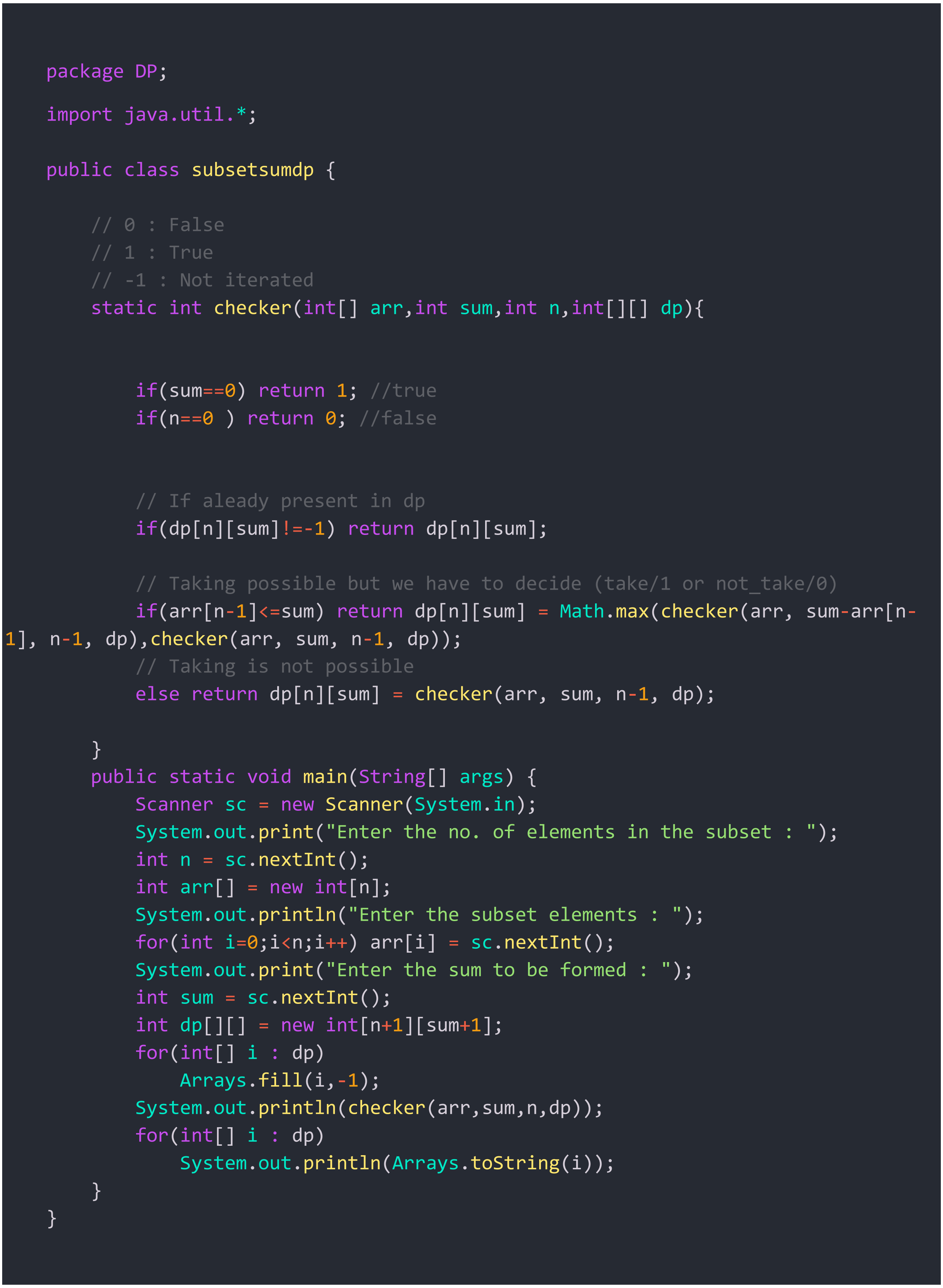
SUBSET SUM PROBLEM

USING DP

CODE:



```
    package DP;
    import java.util.*;

    public class subsetsumdp {

        // 0 : False
        // 1 : True
        // -1 : Not iterated
        static int checker(int[] arr,int sum,int n,int[][] dp){


            if(sum==0) return 1; //true
            if(n==0 ) return 0; //false


            // If aleady present in dp
            if(dp[n][sum]!=-1) return dp[n][sum];

            // Taking possible but we have to decide (take/1 or not_take/0)
            if(arr[n-1]<=sum) return dp[n][sum] = Math.max(checker(arr, sum-arr[n-
1], n-1, dp),checker(arr, sum, n-1, dp));
            // Taking is not possible
            else return dp[n][sum] = checker(arr, sum, n-1, dp);

        }
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter the no. of elements in the subset : ");
            int n = sc.nextInt();
            int arr[] = new int[n];
            System.out.println("Enter the subset elements : ");
            for(int i=0;i<n;i++) arr[i] = sc.nextInt();
            System.out.print("Enter the sum to be formed : ");
            int sum = sc.nextInt();
            int dp[][] = new int[n+1][sum+1];
            for(int[] i : dp)
                Arrays.fill(i,-1);
            System.out.println(checker(arr,sum,n,dp));
            for(int[] i : dp)
                System.out.println(Arrays.toString(i));
        }
    }
```