Dynamic Programming 1 Assignment done by N S K K K Naga Jayanth

https://leetcode.com/problems/climbing-stairs/

```
class Solution {
    public int climbStairs(int n) {
        if (n <= 3) return n;

        int left = 0;
        int right = 1;
        for (int i = 0; i < n; i++) {
            int temp = right;
            right = left + right;
            left = temp;
        }

        return right;
    }
}
```

https://leetcode.com/problems/fibonacci-number/

```
class Solution {
    public int fib(int n) {
        int a = 1;
        int b = 0;
        if(n < 2){
            return n;
        }
        for(int i = 2; i<=n;i++){
            int c = a + b;
            b = a;
            a = c;
        }
```

```
        return a;


    }
}
```

https://leetcode.com/problems/perfect-squares/

```
class Solution {
    public int numSquares(int n) {
        List<Integer> squares = new ArrayList<>();

        int cur = 1;
        while (Math.pow(cur, 2) <= n) {
            squares.add((int) Math.pow(cur++, 2));
        }

        int[] dp = new int[n + 1];
        Arrays.fill(dp, Integer.MAX_VALUE);

        dp[0] = 0;
        for (int i = 1; i < n+1; i++ ) {
            for (int j = 0; j < squares.size() && squares.get(j) <= i; j++) {
                dp[i] = Math.min(dp[i], 1 + dp[i-squares.get(j)]);
            }
        }
        return dp[n];

    }
}
```

https://leetcode.com/problems/decode-ways/

```
public int numDecodings(String s) {
        int n= s.length();
```

```
    if(s==null || n==0)
        return 0;
    int dp[]= new int[n+1];
    dp[0] = 1;
    dp[1] =  s.charAt(0)!='0'?1:0;
    for(int i=2;i<=n;i++){
        int first = Integer.valueOf(s.substring(i-1,i));
        int second = Integer.valueOf(s.substring(i-2,i));
        if(first>=1  && first<=9){
            dp[i]+=dp[i-1];
        }
        if(second>=10 && second<=26){
            dp[i]+=dp[i-2];
        }
    }
    return dp[n];

}
```