DP34: [Wildcard matching](https://leetcode.com/problems/wildcard-matching/)

**str1 =>** contains lowercase + '?' + '\*'

**str2** => contains only lowercase

**Recursion**: we have been given two strings we have to tell whether they are matching completely or not.

=> We will start matching from the n-1, m-1

1. If (str1[i] == str2[j] || str1[i]==’?’) **match**; try matching other characters.

*return f(i-1, j-1)*

1. If (str1[i] == ‘\*’); it can take any seq from 0 char upto entire remaining string so try matching by considering ‘\*’ can take any value.

*return f(i-1, j) || f(i, j-1)* //’\*’ is a 0 char or it is equivalent to 1 char

1. If no above condition encounters that means ith lowercase letter != jth lowercase letter

*return false;*

**=> Base case**

* if(i<0 && j<0) return true;
* if(i<0 && j>=0) return false;
* if(i>=0 && j<0){ //there should be only ‘\*’ from 0 till ith char

for(int ii= 0; ii<=i; ii++){

if(s[ii] != ‘\*’

return false;

}

return true;

}

**Tabulation:**

1. Right shift of index(1-based) to deal with i<0 and j<0 base cases: n->n+1, m->m+1
2. Use base case to make the base of the table
3. Use recurrence relation to form table i=1 to n, j=1 to m