Recursion: <https://sites.google.com/site/smilitude/recursion_and_dp>

Dynamic Programming: <http://www.shafaetsplanet.com/?p=1022>

int fibonacci( int n ) {  
    if( n == 0 || n == 1 ) return 1;  
    else return fibonacci( n-1 ) + fibonacci( n-2 );  
}

dp:

int array[20];  
int fibonacci( int n ) {  
    if( n == 0 || n == 1 ) return 1;  
    else if( array[n] == 0 ) // ami oke ekbaro process kori nai - or khub mon kharap! :(  
        array[n] = fibonacci( n-1 ) + fibonacci( n-2 ); // ye! o khushi! :D  
     
    return array[n];  
}

**Forward:**

**for loop:**

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

// do whatever needed

}

Equivalent recursion:

void FOR(int i, int n) {

if(i==n) return; // terminates

// do whatever needed

FOR(i+1, n); // go to next step

}

Backward:

for loop:

for(int i = n-1; i >= 0; i -= 1) {

// do whatever needed

}

Equivalent recursion:

void ROF(int i, int n) {

if(i==n) return; // terminates

ROF(i+1, n); // keep going to the last

// do whatever needed when returning from prev steps

}

1.You will be given an array of integers, write a recursive solution to print it in reverse order.

void rev(int i, int n, int \*a)

{

if(i==n){

}

else {

rev(i+1, n, a);

printf("%d ", a[i]);

}

}

2. Write a recursive function to print an array in the following order.  
[0] [n-1]  
[1] [n-2]  
.........  
[(n-1)/2] [n/2]  
  
Input:  
5  
1 5 7 8 9  
Output:  
1 9  
5 8  
7 7

void print(int i, int j, int \*a)

{

    if(i<=j)

    {

        printf("%d %d\n", a[i], a[j]);

        print(i+1, j-1, a);

    }

}

3. Write a recursive solution to print the polynomial series for any input n:  
1 + x + x2 + ................. + xn-1

void print\_poly(int i, int n)

{

if(i<n)

{

if(i==0) printf("1");

else

{

printf(" + x");

if(i>1) printf("^%d", i);

}

print\_poly(i+1, n);

}

}

5. Write a recursive program to compute n!

6. Write a recursive solution to get the reverse of a given integer. Function must return an int

Input:

123405

Output:

504321

int rev(int n, int c)  
{  
    if(n==0) return c;  
    c = c\*10 + n%10;  
    return rev(n/10, c);  
}