1. Print 1 to n without loop

void printNos(int N)

{

if(N==0) return;

printNos(N-1);

cout<<N<<" ";

}

1. Sum of digits of a number

int sumOfDigits(int n)

{

if(n==0)

return n;

return n%10+sumOfDigits(n/10);

}

1. Count total digits in anumber

int countDigits(int n)

{

if(n<10) return 1;

return 1+countDigits(n/10);

}

1. Digital root

int digitalRoot(int n)

{

if(n<10) return n;

int cpy = n, sum = 0;

while(cpy)

{

sum+=cpy%10;

cpy/=10;

}

return digitalRoot(sum);

}

1. Fibonacci using recursion

int fibonacci(int n)

{

if(n<3) return 1;

return fibonacci(n-1)+fibonacci(n-2);

}

1. Tower of Hanoi

long long toh(int N, int from, int to, int aux) {

if(N==1)

{

cout<<"move disk "<<N<<" from rod "<<from<<" to rod "<<to<<endl;

return 1;

}

int count = 0;

count+=toh(N-1, from, 6-to-from, 0);

count++;

cout<<"move disk "<<N<<" from rod "<<from<<" to rod "<<to<<endl;

count+=toh(N-1, 6-to-from, to, 0);

return count;

}

1. Josephus problem

int josephus(int n, int k)

{

if(n==1) return n;

return (josephus(n-1, k)+k-1)%n+1;

}

1. Lucky numbers

bool isLucky(int n) {

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

{

if (n%i==0) return false;

if(n<i) return true;

n = n-n/i;

}

}

1. Power using recursion

int RecursivePower(int n,int p)

{

if(p==0) return 1;

return n\*RecursivePower(n,p-1);

}

1. Power of numbers

long long power(long long N,long long R)

{

long long lim = 1e9+7;

if(R==0) return 1;

if(R&1) return N\*power(N\*N%lim,R/2)%lim;

else return power(N\*N%lim, R/2);

}

1. Power set using recursion

def powerSet(s):

cpy = s

res = []

if(len(cpy)==1):

res = ["",cpy]

return res

ans = powerSet(cpy[1:])

for i in range(len(ans)):

res.append(ans[i])

res.append(cpy[0]+ans[i])

return res;

1. Possible words from phone digits

def possibleWords(self,a,N):

arr = ["abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "wxyz"]

if N==1:

ans = []

for i in range(len(arr[a[0]-2])):

ans.append(arr[a[0]-2][i])

return ans

res = self.possibleWords(a[1:],N-1)

ans = []

for i in range(len(arr[a[0]-2])):

for j in range(len(res)):

ans.append(arr[a[0]-2][i]+res[j])

return ans