**Recursion Basics**

Recursion is one of the basic topics in competitive programming now-a-days .At some point of time you might have come across a problem that you know can be solved using recursion but you are really confused about where to start. I will try to explain a few points to keep in mind while designing a recursive solution.

1) Try to come up with solutions of smaller or base cases .  
2) Then think about the recursive solution or bigger solution.

Let’s take an example of an easy question :  
**Finding the base b equivalent of any decimal. 2<=b<=9.**

A little overview of conversion algorithm for those who don’t know :  
You have to repeatedly divide the number by b and store the remainders until you get to the point where you can’t divide any more i.e the number becomes less than b. Because in that case ,the remainder will be 0.Finally you have to print the remainders backwards.

For example 31(base 10) == 37(base 8) == 34(base 9)

So, you can see the base case occurs when the number is less than b. So we can clearly form the base case in c++ in the following way :

Void ConvertToBase(int n,int b)

{

If (n<b)

{

cout<<n; //Base Case

return;

}

//Recursive case goes here

}

Now, let's try to figure out how to write the recursive case for this problem.Now you can observe that we are printing the remainders backwards.So,this means that first you have to print the remainder of the recursive function and then you have to print the remainder of the current number divided by b.

This means first you have to call ConvertToBase(n/b, b) so that it prints all the remainders for the smaller cases and then you can print current remainder.

So, Recursive case becomes :

ConvertToBase(n/b, b);

cout<<n%b;

So finally our recursive function will look something like this :

Void ConvertToBase(int n,int b)

{

If (n<b)

{

cout<<n;

return;

}

ConvertToBase(n/b, b);

cout<<n%b; //print the current remainder.

}

So, you how we easily come up with neat and quick solutions using recursion.Similarly we can try to write a recursive solution for base b>10. Let's take an example where b<=16.The only change you have to make in the above function is to add a few more base cases by storing the base b equivalents of numbers up to b-1(or all possible remainders) in a hash.In this case you can use character array like this :

hash[0] = '0'

hash[1] = '1'

.

.

.

.

hash[10]='A'

hash[11]='B'

.

.

hash[15]='F'

Now ,you just have to change the function to print the values of remainders from the hash.

Void ConvertToBase(int n,int b)

{

If (n<b)

{

cout<<hash[n];

return;

}

ConvertToBase(n/b, b);

cout<<hash[n%b]; //print the current remainder.

}

There are many other examples of recursion like Fibonacci sequence (most common), binary search,finding number of combinations from n items taking r at a time etc. You will find the use of recursion mainly in backtracking algorithms like N-Queens,Sudoku solving,depth first search ,Subset sum,knight's tour , maze solving problem and many more.