

Recursive Functions

Recursion

Recursion Concept

- Recursion is a programming technique that allows the programmer to express operations in terms of themselves.
- In C, this takes the form of a function that calls itself.
- A useful way to think of recursive functions is to imagine them as a process being performed where one of the instructions is to "repeat the process".

Recursion Cont.....

- This makes it sound very similar to a loop.
- This is because it repeats the same code, and in some ways it is similar to looping without the recursive call.
- Recursion makes it easier to express ideas in which the result of the recursive call is necessary to complete the task.

Example

```
void recursive()  
{  
    recursive();    /* Function calls itself */  
}
```

```
int main()  
{  
    recursive();    /* Sets off the recursion */  
    return 0;  
}
```

- This program will not continue forever.
- The computer keeps function calls on a stack and once too many are called without ending, the program will crash.
- The number of calls in a recursive function must be controlled, otherwise stack overflow causes the program to crash.

```
void recursive_Count ( int count )

/* Each call gets its own copy of count */
{
    println( "%d\n", count );
    /* It is not necessary to increment count since each
function's variables are separate (so each count will be
initialized one greater) */

    recursive_Count ( count + 1 );
}

int main()
{
    Recursive_Count ( 1 );

    /* First function call, so it starts at one */

    return 0;
}
```

```
void count_to_ten ( int count )
{
    /* we only keep counting if we have a value less than ten */
    if ( count < 10 )
    {
        count_to_ten( count + 1 );
    }
}

int main()
{
    count_to_ten ( 0 );
}
```

```
void printnum ( int begin )
{
    println( "%d", begin );
    if ( begin < 9 )
    /* The base case is when begin is no longer less than 9 */
    {
        printnum ( begin + 1 );
    }
    /* display begin again after we've already printed everything
    from 1 to 9 * and from 9 to begin + 1 */
    println( "%d", begin );
}
```