**THE C PREPROCESSOR**

The two most frequently used features are **#include**, to include the contents of a file during compilation, and **#define**, to replace a token by an arbitrary sequence of characters.

**1) FILE INCLUSION**

Any source line of the form

#include “filename”

Or

#include <filename>

is replaced by the contents of the file **filename**. If the **filename** is quoted, searching for the file typically begins where the source program was found; if it is not found there, or if the name is enclosed in **<** and **>,** searching follows an implementation-defined rule to find the file. An included file may itself contain **#include** lines.

**2) MACRO SUBSTITUTION**

#define name replacement-text

Subsequent occurrences of the token **name** will be replaced by the **replacement*-*text**. Never terminate a Macro definition by a **;**

If we need to upgrade the value at all the places in the program, this is lot easier if we have used **#define** directive as we require to change only at macro definition and all changes will be reflected at all places.

Substitution are made only for tokens, and do not take place within quoted strings. For example, if YES is a defined name, there would be no substitution in printf( “YES” ) or in YESMAN.

It is also possible to define macro with arguments, so the **replacement-text** can be different for different calls of the macro.

#define max( A, B ) ( ( A ) > ( B ) ? ( A ) : ( B ) )

Macros can be split into multiple lines with a ‘\’ (backslash) present at the end of each line.

#define display for( i = 0; i <= 2; i++ )\

Printf(“A”);

**3) CONDITIONAL INCLUSION**

It is possible to control preprocessing itself with conditional statements that are evaluated during preprocessing. This provides a way to include code selectively, depending on the value of conditions evaluated during compilation.

#ifdef, #ifndef, #if, #endif, #elif, #else