Here are some important points to remember while writing macros with arguments:

(a) Be careful not to leave a blank between the macro template and its argument while defining the macro. For example, there should be no blank between **AREA** and **(x)** in the definition, #define AREA(x) (3.14 * x * x)

If we were to write **AREA (x)** instead of **AREA(x)**, the **(x)** would become a part of macro expansion, which we certainly don't want. What would happen is, the template would be expanded to

(r1) (3.14 * r1 * r1)

which won't run. Not at all what we wanted.

(b) The entire macro expansion should be enclosed within parentheses. Here is an example of what would happen if we fail to enclose the macro expansion within parentheses.

```
#define SQUARE(n) n * n
main( )
{
    int j ;

    j = 64 / SQUARE ( 4 ) ;
    printf ( "j = %d", j ) ;
}
```

The output of the above program would be:

j = 64

whereas, what we expected was j = 4.

What went wrong? The macro was expanded into

```
j = 64 / 4 * 4 ;
```

which yielded 64.

(c) Macros can be split into multiple lines, with a '\' (back slash) present at the end of each line. Following program shows how we can define and use multiple line macros.

```
#define HLINE   for ( i = 0 ; i < 79 ; i++ ) \
                        printf ( "%c", 196 ) ;

#define VLINE( X, Y ) {\
                          gotoxy ( X, Y ) ; \
                          printf ( "%c", 179 ) ; \
                      }
main( )
{
   int i, y ;
   clrscr( ) ;

   gotoxy ( 1, 12 ) ;
   HLINE

   for ( y = 1 ; y < 25 ; y++ )
       VLINE ( 39, y ) ;
}
```

This program draws a vertical and a horizontal line in the center of the screen.

(d) If for any reason you are unable to debug a macro then you should view the expanded code of the program to see how the macros are getting expanded. If your source code is present in the file PR1.C then the expanded source code would be stored

in PR1.I. You need to generate this file at the command prompt by saying:

```
cpp pr1.c
```

Here CPP stands for C PreProcessor. It generates the expanded source code and stores it in a file called PR1.I. You can now open this file and see the expanded source code. Note that the file PR1.I gets generated in C:\TC\BIN directory. The procedure for generating expanded source code for compilers other than Turbo C/C++ might be a little different.