```
////////////////////////////////////////////////////////////////////////
//      fact.c program to test a recursive call in C
//         M. Mizuno (c) 1995, 2004, 2005
//          modified for Learning Tree course 223P
//      NOTE: In gcc for H-8, default for char is "unsigned"
////////////////////////////////////////////////////////////////////////
/* Draw memory image of execution of the following program */
   int ans;

   void fact(int i)
   {
        /* what will happen if x is declared as internal static */
        int x;

        x = i - 1;
        if  ((i == 0) || (x == 0)) ans = 1;
        else {
          fact(x);
          ans = (x + 1) * ans;
                }
   }

   int main(void)
   {
        fact(3);
        printf("asnswer = %d\n", ans);
        return 0;
   }

```

```
; Listing generated by Microsoft (R) Optimizing Compiler Version 15.00.21022.08

        TITLE   c:\LearningTree\223P\Programs\memory_test\Pentium\fact.c
        .686P
        .XMM
        include listing.inc
        .model  flat

INCLUDELIB LIBCMT
INCLUDELIB OLDNAMES

_DATA   SEGMENT
COMM    _ans:DWORD
$SG742  DB      'asnswer = %d', 0aH, 00H
_DATA   ENDS
PUBLIC  _fact
; Function compile flags: /Odtp
; File c:\learningtree\223p\programs\memory_test\pentium\fact.c
_TEXT   SEGMENT
_x$ = -4                                                ; size = 4
_i$ = 8                                                 ; size = 4
_fact   PROC

; 11   :    {

  00000 55              push    ebp
  00001 8b ec           mov     ebp, esp
  00003 51              push    ecx

; 12   :         /* what will happen if x is declared as internal static */
; 13   :         int x;
; 14   :
; 15   :         x = i - 1;

  00004 8b 45 08        mov     eax, DWORD PTR _i$[ebp]
  00007 83 e8 01        sub     eax, 1
  0000a 89 45 fc        mov     DWORD PTR _x$[ebp], eax

; 16   :         if  ((i == 0) || (x == 0)) ans = 1;

  0000d 83 7d 08 00     cmp     DWORD PTR _i$[ebp], 0
  00011 74 06           je      SHORT $LN2@fact
  00013 83 7d fc 00     cmp     DWORD PTR _x$[ebp], 0
  00017 75 0c           jne     SHORT $LN3@fact
$LN2@fact:
  00019 c7 05 00 00 00
        00 01 00 00 00  mov     DWORD PTR _ans, 1

; 17   :         else {

  00023 eb 1f           jmp     SHORT $LN4@fact
$LN3@fact:

; 18   :           fact(x);

  00025 8b 4d fc        mov     ecx, DWORD PTR _x$[ebp]
  00028 51              push    ecx
  00029 e8 00 00 00 00  call    _fact
  0002e 83 c4 04        add     esp, 4

; 19   :           ans = (x + 1) * ans;

  00031 8b 55 fc        mov     edx, DWORD PTR _x$[ebp]
  00034 83 c2 01        add     edx, 1
  00037 0f af 15 00 00
        00 00           imul    edx, DWORD PTR _ans
  0003e 89 15 00 00 00
        00              mov     DWORD PTR _ans, edx
$LN4@fact:

; 20   :               }
; 21   :    }
```



```
  00044 8b e5           mov     esp, ebp
  00046 5d              pop     ebp
  00047 c3              ret     0
_fact   ENDP
_TEXT   ENDS
PUBLIC  _main
EXTRN   _printf:PROC
; Function compile flags: /Odtp
_TEXT   SEGMENT
_main   PROC

; 24   :    {

  00050 55              push    ebp
  00051 8b ec           mov     ebp, esp

; 25   :         fact(3);

  00053 6a 03           push    3
  00055 e8 00 00 00 00  call    _fact
  0005a 83 c4 04        add     esp, 4

; 26   :        printf("asnswer = %d\n", ans);

  0005d a1 00 00 00 00  mov     eax, DWORD PTR _ans
  00062 50              push    eax
  00063 68 00 00 00 00  push    OFFSET $SG742
  00068 e8 00 00 00 00  call    _printf
  0006d 83 c4 08        add     esp, 8

; 27   :        return 0;

  00070 33 c0           xor     eax, eax

; 28   :    }

  00072 5d              pop     ebp
  00073 c3              ret     0
_main   ENDP
_TEXT   ENDS
END
```