

```

PUBLIC      main_
PUBLIC      _counter
DGROUP     GROUP _CONST,_DATA
_TEXT      SEGMENT      BYTE PUBLIC USE16 'CODE'
ASSUME CS:_TEXT, DS:DGROUP, SS:DGROUP
;
; static uint16_t fatt( uint16_t n )
0000  _fatt:                                ; NOTARE l'offset 0 di inizio segmento
0000      PUSH      BP
0003      MOV       BP,SP
0006      SUB       SP,2
; {      uint16_t ris;
;      if( n==0 )
0010 L$1:
0010      CMP       word ptr 8[BP],0
0014      JNE       L$2
;      ris = 1;
0016      MOV      word ptr -2[BP],1          ; notare accesso a variabile sullo stack
;      else
0021      JMP       L$3
;      ris=fatt( n-1 );
0023 L$2:
0023      MOV       AX,word ptr 8[BP]
0026      DEC       AX
0027      PUSH      AX
0028      CALL     near ptr _fatt
0031      ADD       SP,2
0034      MOV       word ptr -2[BP],AX
;      return ris;
0037 L$3:
0037      MOV       AX,word ptr -2[BP]
; }
0046 L$4:
0046      MOV       SP,BP
0048      POP       BP
0049      RET
;
; int main(void)
0050 main_:
0050      PUSH      BP
0051      MOV       BP,SP
0053      SUB       SP,2
; {      uint16_t ris;
;      counter=3;
0057 L$5:
0057      MOV      word ptr _counter,3      ; notare accesso a variabile globale (Data)
;      ris = fatt(3);
0063      MOV       AX,3
0066      PUSH      AX
0067      CALL     near ptr _fatt
0070      ADD       SP,2
0073      MOV       word ptr -2[BP],AX
;
;      return 0;
; }
0081      MOV       AX,0
0084 L$6:
0084      MOV       SP,BP
0086      POP       BP
0087      RET
0088 L$7:
_TEXT      ENDS
_CONST     SEGMENT      WORD PUBLIC USE16 'DATA'
_CONST     ENDS
_DATA     SEGMENT     WORD PUBLIC USE16 'DATA'
;
; uint16_t counter=99;
0000 _counter:                                ; NOTARE l'offset 0 di inizio segmento
0000      DB      63H, 0
0002 _DATA     ENDS

```