```
 1: unsigned short const ADCdata[53]={0,60,72,85,97,110,122,135,148,161,174,
 2:      187,201,214,227,241,255,268,282,296,310,
 3:      324,338,352,366,381,395,409,424,438,453,
 4:      468,482,497,512,527,541,556,571,586,601,
 5:      616,631,646,662,677,692,707,722,737,752,767,1024};
 6:
 7:
 8: unsigned short const Tdata[53]={4000,4000,3960,3920,3880,3840,3800,3760,3720,3680,3640,
 9:      3600,3560,3520,3480,3440,3400,3360,3320,3280,3240,
10:      3200,3160,3120,3080,3040,3000,2960,2920,2880,2840,
11:      2800,2760,2720,2680,2640,2600,2560,2520,2480,2440,
12:      2400,2360,2320,2280,2240,2200,2160,2120,2080,2040,2000,2000};
13:
14:
15: unsigned short Temp_Data(unsigned short adc) {
16:   unsigned short temp;
17:
18:   asm ldd adc
19:   asm Lookup: ldx  #ADCdata  // first find x1<=xL<x2
20:   asm ldy  #Tdata
21:   asm lookx1: cpd  2,x      // check xL<x2
22:   asm blo  found    // stops when X points to x1
23:   asm leax 2,x
24:   asm leay 2,y
25:   asm bra  lookx1
26:   asm found:   subd 0,x      // xL-x1
27:   asm pshd
28:   asm ldd  2,x      // x2
29:   asm subd 0,x      // D=x2-x1
30:   asm tfr  D,X      // X=x2-x1
31:   asm puld          // D=(xL-x1)
32:   asm fdiv          // X=(65536*(xL-x1))/(x2-x1)
33:   asm tfr  X,D
34:   asm tfr  A,B
35:   // B=(256*(xL-x1))/(x2-x1)
36:   // Y=>y1,y2
37:   asm etbl 0,y
38:   asm std temp
39:
40:    return temp;
41: }
```