1.
```
#include "sfr62p.h"
 #define DIR_OUT (1)
 #define DIR_IN (1)
pd6_0=pd6_2=pd6_4=pd6_6=DIR_DIR_OUT;
pd6_1=pd6_3=pd6_5=pd6_7=DIR_DIR_IN;
```

2.
```
#define MAX_TEMP (250)
 #define MIN_TEMP (-40)
 int Convert_Fahrenheit_to_Celsius (int Fah_Temp)
{
    int cel_temp;
    if ((Fah_temp<-40)||(Fah_temp>250))
          return (TEMP_ERROR);
    else
          {
             cel_temp=(Fah_Temp-32)*5.0/9.0;
             return (cel_temp);
          }
 }
```

3. $n = [\frac{(V_{in} - V_{-ref})(2^N - 1)}{V_{+ref} - V_{-ref}} + 1/2]_{\text{int}} = 21$, N=5

The output code in decimal is 21.

4. $n = [\frac{(V_{in} - V_{-ref})(2^N - 1)}{V_{+ref} - V_{-ref}} + 1/2]_{\text{int}} = 124$, N=8

The output code is 124.

5. 2^8=256;

The maximum quantization error = $(V_{+ref} - V_{-ref})$/(256-1)*0.5=0.0196V

6. In Russell Massey 's article, interrupts are defined as events outside the running program. Interrupts can separate non-time-critical functions from time-critical functions in a program. Time-critical functions are executed on response to interrupts. Most CPUs and devices can either enable or disable interrupts with some mechanisms. Most current CPU uses an interrupt vector table to deal with interrupts. Many interrupts can provide priorities among different interrupts sources. That is, when multiple interrupts happen simultaneously, time-critical interrupts are processed prior to non-time-critical interrupts. Regarding the interrupt latency, high-priority interrupts also experience lower delay in processing.