

Q1-

1918
1917
1916
1916
1917
1912
1919
1914
1912
1918
1912
1916
1914
1914
1912
1918
1919
1918
1912
1912
1919
1912
1916
1917
1918
1919
1917
1914
19

Q3-

A screenshot of a terminal window. The title bar at the top reads "COM6 9600 bps, 8N1, no handshake". On the right side of the title bar are three buttons: "Settings", "Clear", and "About". The main area of the terminal is filled with 25 lines of the text "0,42" in green. The final line of the visible text is "0,".

Q4-

[illegible]

Q5-

```
1: #include "Init.h"
2: int data2,output_data,temp;
3: int main()
4: {
5:     init_gpio_adc();
6:     systick_init();
7:     init_LED();
8:     while(1)
9:     {
10:
11:
12:     }
13: }
14: void SysTick_Handler ( void )
15: {
16:     int data=take_value();
17:     temp=165*data/2047;
18:     print_number_with_decimal(temp);
19:     if(temp>164)
20:         GPIOF->DATA |=0x8;
21:     else
22:         GPIOF->DATA &=0;
23: }
```

```
1: #include "TM4C123GH6PM.h"
2: extern void OutStr(char*);
3: int value1,value2;
4: char msg1[100],msg2[100];
5: int new_number,int_number,detect;
6: int ;
7: void print_number(int number);
8: void init_gpio_adc();
9: int take_value();
10: void print_number_with_decimal(int number);
11: void systick_init();
12: void print_number_with_decimal(int number)
13: {
14:     int i=0,j=0;
15:     if(number>99)
16:         detect=2;
17:     else if(number<100 && number>9)
18:         detect=1;
19:     else if(number>0 && number<10)
20:     {
21:         detect=0;
22:     }
23:     else if(number<-9 && number>-100)
24:     {
25:         detect=3;
26:         number=number*(-1);
27:     }
28:     else if(number<0 && number>-10)
29:     {
30:         detect=5;
31:         number=number*(-1);
32:     }
33:     else if(number<-99)
34:     {
35:         detect=4;
36:         number=number*(-1);
37:     }
38:     while(number){
39:         new_number=number/10;
40:         msg1[i]=number-(new_number*10)+48;
41:         number=new_number;
42:         i++;
43:     }
44:     if(detect==1)
45:     {
46:         msg2[j]=48;
47:         j++;
48:         msg2[j]=44;
49:         j++;
50:         for(i=i-1;i>=0;i--)
51:         {
52:             msg2[j]=msg1[i];
53:             j++;
54:         }
```

```
55:         }
56:     else if(detect==2)
57:     {
58:         for(i=i-1;i>=0;i--)
59:         {
60:             if(i==1)
61:             {
62:                 msg2[j]=44;
63:                 j++;
64:             }
65:             msg2[j]=msg1[i];
66:             j++;
67:         }
68:     }
69:     else if(detect==3)
70:     {
71:         msg2[j]=45;
72:         j++;
73:         msg2[j]=48;
74:         j++;
75:         msg2[j]=44;
76:         j++;
77:         for(i=i-1;i>=0;i--)
78:         {
79:             msg2[j]=msg1[i];
80:             j++;
81:         }
82:     }
83:     else if(detect==4)
84:     {
85:         msg2[j]=45;
86:         j++;
87:         for(i=i-1;i>=0;i--)
88:         {
89:             if(i==1)
90:             {
91:                 msg2[j]=44;
92:                 j++;
93:             }
94:             msg2[j]=msg1[i];
95:             j++;
96:         }
97:     }
98:     else if(detect==0)
99:     {
100:         msg2[j]=48;
101:         j++;
102:         msg2[j]=44;
103:         j++;
104:         msg2[j]=48;
105:         j++;
106:         for(i=i-1;i>=0;i--)
107:         {
108:             msg2[j]=msg1[i];
109:             j++;
110:         }
111:     }
112:     else if(detect==5)
113:     {
114:         msg2[j]=45;
115:         j++;
116:         msg2[j]=48;
117:         j++;
118:         msg2[j]=44;
119:         j++;
120:         msg2[j]=48;
121:         j++;
122:         for(i=i-1;i>=0;i--)
123:         {
124:             msg2[j]=msg1[i];
125:             j++;
126:         }
127:     }
128:     msg2[j]='\r';
129:     msg2[j+1]='\4';
130:     OutStr(msg2);
131: }
132: void print_number(int number)
133: {
134:     int i=0,j=0;
135:     while(number){
136:         new_number=number/10;
137:         msg1[i]=number-(new_number*10)+48;
138:         number=new_number;
139:         i++;
140:     }
141:     for(i=i-1;i>=0;i--){
142:         msg2[j]=msg1[i];
143:         j++;
144:     }
145:     msg2[j]='\r';
146:     msg2[j+1]='\4';
147:     OutStr(msg2);
148: }
149: }
150: void init_gpio_adc()
151: {
152:     SYSCTL->RCGCADC |= 0x1;
153:     value1= SYSCTL->PRADC;
154:     __NOP();
155:     __NOP();
156:     __NOP();
157:     SYSCTL->RCGCGPIO |= 0x10;
158:     __NOP();
159:     __NOP();
160:     __NOP();
161:     GPIOE->AFSEL |= 0x08;
162:     GPIOE->DEN |= 0x08;
```

```
163:     GPIOF->AMSEL |= 0x08;
164:     //GPIOF->DIR |= ~0x08;
165:     ADC0->ACTSS |= 0x08;
166:     ADC0->EMUX |= ~0xF000;
167:     ADC0->SSMUX3 |= 0x0000;
168:     ADC0->SSCTL3 |= 0x06;
169:     ADC0->PC |= 0x01;
170:     ADC0->ACTSS |= 0x08;
171:
172: }
173: int take_value()
174: {
175:     while(1)
176:     {
177:         ADC0->PSSI |= 0x08;
178:         while(1)
179:         {
180:             if(ADC0->RIS & 0x08)
181:             {
182:                 break;
183:             }
184:         }
185:         value2=ADC0->SSFIF03;
186:         ADC0->ISC |= 0x08;
187:         return(value2);
188:     }
189: }
190: void systick_init()
191: {
192:     SysTick->LOAD = 1599999; // C o n f
193:     SysTick->VAL = 0; // C l e a r t h e t
194:     SysTick->CTRL = 0x07; // s o u r c
195: }
196: }
197: void init_LED(){
198:     SYSCCTL->RCGCGPIO |= 0x20;
199:     GPIOF->DIR |= 0x8; // s e t G R E E N
200:     GPIOF->DEN |= 0x8; // E n a b l e P F 2 p
201:
202:
203: }
```