This is a closed book exam. You must put your answers on this piece of paper only. You have 50 minutes, so allocate your time accordingly. Please read the entire quiz before starting.

<table>
<thead>
<tr>
<th>(5) Question 1.</th>
<th>$56</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5) Question 2.</td>
<td>$3900 wrap</td>
</tr>
</tbody>
</table>
| (5) Question 3. | 0  
| | X=>  0 data  
| | 1 Old X  
| | 3 Return address  
| | 5 in |
| (5) Question 4. | 5 |
| (5) Question 5. | R1 = (5-2.5-0.5V)/0.001A = 2000 Ω =  2 kΩ |
| (5) Question 6. | C) The l bit in the CCR is clear  
| | D) The software arms the interrupt (e.g., RTIE=1)  
| | G) The hardware sets the flag bit (e.g., RTIF=1) |
| (5) Question 7. | Resolution = range/precision = 20V/1023 ≈ 20mV |
| (5) Question 8. | The first guess is half way in between 0 and 5V  
| | E) 2.5 V |
| (5) Question 9. | C) This ISR did not acknowledge the interrupt (clear RTIF), so it will interrupt over and over continuously. |
| (15) Question 10. | org $3800 ;globals in RAM  
| | count rmb 2 ;16-bit debugging counter  
| | org $4000 ;programs in ROM  
| | Init movw #0,count ; initialize counter  
| | rts  
| | Pulse bset PTT,#1 ; send pulse on PT0  
| | bclr PTT,#1  
| | pshx ;don’t want to mess up system  
| | ldx count  
| | inx ;debugging instrument  
| | stx count  
| | pulx  
| | rts |
(15) Question 11.

Convert

```assembly
ldd 0,Y ; RegD = 0 to 1023, the input data
ldx #26214 ; RegX = 65536/2.5
fdiv ; RegX = (65536*Input)/(65536/2.5)
tfr X,D ; RegD = 2.5*Input
addd #100 ; Output = 2.5*Input+100
rts
```

Convert2

```assembly
ldd 0,Y ; RegD = 0 to 1023, the input data
lsld ; 2*Input
pshd
ldd 0,Y ; second copy of input data
lsrd ; Input/2 (ignore LSB error)
addd 2,sp+ ; 2*Input+Input/2
addd #100 ; Output = 2.5*Input+100
rts
```

(5) Question 12.

baud rate = 2000 bytes/sec *1 frame/byte* 11 bits/frame = 22000 bps

(20) Question 13.

```c
; short calc(void){
sum set 2
n set 0 ;short sum,n;
leas -4,sp ;allocate
movw #100,n,sp ;n = 100;
movw #0,sum,sp ;sum = 0;
loop
  ldd sum,sp ;do( sum = sum+n;
  addd n,sp ;sum+n
  std sum,sp
  ldx n,sp
  dex
  stx n,sp
  bne loop
  ldd sum,sp ;return(sum);
leas 4,sp
rts
```