(5) Question 1. Precision in decimal digits	20/0.1 is 200 alternatives, which is **2½ decimal digits**
(2) Part 2a. Choose A-E	**B)** From memory to 6812, or from input device to 6812
(2) Part 2b. Choose A-E	**A)** Real time
(2) Part 2c. Choose A-E	**C)** FIFO queue
(2) Part 2d. Choose A-E	**C)** The two terms describe exactly the same synchronization method.
(2) Part 2e. Choose A-E	**B)** Drop out occurs after a right shift or a divide, and an intermediate result looses its ability to represent all of the values.
(2) Part 3a. Specify **`RegB`**	170+125=295, RegB = 295-256 = **39**
(2) Part 3b. Specify **`0 or 1`**	Consider as unsigned, 170+125=295 is too big, so **C=1**
(1) Part 3c. Specify **`0 or 1`**	Consider as signed, 86+125=39 is ok, so **V=0**
(2) Part 4a. Give the value for **`xxx`**	**2** (p is next to top)
(3) Part 4b. Give the value for **`yyy`**	**0** (q is on top)

(5) Question 5. Give the value for **`zzz`**	`SP  -> locals`
	`X   -> oldX`
	`X+2 -> return`
	`X+4 -> port` **`zzz = +4`**

(5) Question 6. Give the value	$C000 equals –16384 value = –16384/100 = **-163.84**

(5) Question 7. Give the hexadecimal value	After the pushes
	`SP  -> $43`
	`$21`
	`$65`
	`$87`
	`pula` makes RegA =$43
	After the `leas 2,sp`
	`SP  -> $87`
	`pulb` makes RegB =$87
	so **Reg D = $4387**

(5) Question 8. Give the op code	It is signed, so we need **`ble`**

(5) Question 9. Simplified memory cycles (you may or may not need all 5 entries)

R/W	Addr	Data
R	4000	EE
R	4001	71
R	3900	00
R	3901	01

(5) Question 10. Show the assembly code

```
ldd 13,x
```

(10) Question 11. Hand assemble this program

Address	Machine code	Source code
$3900		org $3900
$3900		aa rmb 2
$3902		bb rmb 1
$0044		TCNT equ $0044
$4300		org $4300
$4300	$DE $44	ldx TCNT
$4302	$CE $43 $0C	loop ldx #cc
$4305	$A6 $02	ldaa 2,x
$4307	$7A $39 $02	staa bb
$430A	$20 $F6	bra loop
$430C	$00 $01 $02 $03 $04	cc fcb 0,1,2,3,4

(15) Question 12. Show the assembly code

```
* Input one character from SCI terminal
* Inputs: none    Outputs: RegB is ASCII character
* Registers modified: CCR
RDRF        equ  $20
SCI_InChar  brclr SCISR1,#RDRF,SCI_InChar   Gadfly wait for RDRF set
            ldab SCIDRL   ASCII character code
            rts
```

(20) Question 13. Show the assembly code

```
Main  lds  #$4000
      bset DDRM,#$03   PM1,PM0 outputs
      bclr DDRT,#$03   PT1,PT0 inputs
      ldx  #Stop       Reg X => current state
*Linked list interpreter
LL    ldaa PTT
      anda #$03
      cmpa #1
      beq  is1
      cmpa #2                                LL2   ldaa PTT
      beq  is2                                     anda #$03
      cmpa #3                                      movb A,x,PTM  ; output
      beq  is3                                     lsla          ; 0,2,4,6
is0   movb 0,x,PTM  ; output In=0                  adda #4       ; 4,6,8,10
      ldx  4,x      ; next In=0                    ldx  A,x      ; next
      bra  LL                                      bra  LL
is1   movb 1,x,PTM  ; output In=1
      ldx  6,x      ; next In=1
      bra  LL
is2   movb 2,x,PTM  ; output In=2
      ldx  8,x      ; next In=2
      bra  LL
is3   movb 3,x,PTM  ; output In=3
      ldx  10,x     ; next In=3
      bra  LL
      org  $FFFE
      fdb  Main        reset vector
```