(5) **Question 1.** C) drop out

(5) **Question 2.** \( \%11010011 = -128 + 64 + 16 + 2 + 1 = -45 \)

(4) **Part 3a.** \( 101 + 210 = 311 \), which wraps around to \( 311 - 256 = 55 \)

(2) **Part 3b.** To find the C bit, do the math in unsigned math, which is \( 101 + 210 = 311 \). The carry bit is 1 because 55 is the wrong answer.

(2) **Part 3c.** To find the V bit, do the math in signed math, which is \( 101 + -46 = 55 \). The overflow bit is 0 because 55 is the correct answer.

(2) **Part 3d.** The Z bit is 0 because the result is not zero.

(2) **Part 3e.** The N bit is 0 because the result is between 0 and 127 (bit 7 is zero).

(10) **Question 4.** The minimum number is \(-128/8\) which is \(-16\). The maximum number is \(+127/8\) which is \(15.875\).

(10) **Question 5.** You get the opcode, E4, from the ANDB page. You get the postbyte \( \textbf{x}b \), 5E, from Table A-3. Together the machine code is \$E45E\.

(5) **Question 6.** Since it is a post decrement the effective address is simply the initial value of RegY before the decrement, which is \$3900\.

(15) **Question 7.** There are two read cycles to fetch the opcode and operand. There are two write cycles to push the return address. The stack pointer is decremented as data is pushed, The PC is changed to point to the subroutine.

<table>
<thead>
<tr>
<th>R/W</th>
<th>Addr</th>
<th>Data</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>$F124</td>
<td>$07</td>
<td>opcode fetch IR=$07, PC=$F125</td>
</tr>
<tr>
<td>R</td>
<td>$F125</td>
<td>$66</td>
<td>operand fetch, PC=$F126</td>
</tr>
<tr>
<td>W</td>
<td>$3FFF</td>
<td>$26</td>
<td>push return address, SP=$3FFF</td>
</tr>
<tr>
<td>W</td>
<td>$3FFE</td>
<td>$F1</td>
<td>push return address, SP=$3FFE, PC=$F18C</td>
</tr>
</tbody>
</table>

(40) **Question 8.** All the object code goes in ROM

```
org $4000  ; ROM
**********Init**********
* Set direction registers so PTT are output and PTM are input
* Inputs: none
* Outputs: none
* Errors: none
Init movb #$FF,DDRT  ; PTT are outputs
   clr DDRM       ; PTM are inputs
   rts
**********Look**********
* Look at PTM inputs, and set PTT to $55 if PTM<=$20
* Inputs: none
* Outputs: none
* Errors: none
Look ldaa PTM  ; read inputs
   cmpa #$20
   bhs skip    ; leave unchanged if PTM>=$20
   movb #$55,PTT
skip rts
```