1. Fill in the blank with what is in the accumulator after each of the following sequences run.

A) `mov a, #0FFh
clr c
mov r0, #3
Lp: rlc a
djnz r0, Lp`

B) `mov a, #77h
mov r0, #0F3h
xrl a, r0
mov r1, a
orl a, #22h`

C) `setb c
mov a, #12h
mov r0, #3
Lp: rlc a
djnz r0, Lp`

A = ___________
A = ___________
A = ___________

2. For each instruction below fill in the addressing mode of the second operand.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Addressing Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>A) mov a, r0</td>
<td></td>
</tr>
<tr>
<td>B) mov a, @r0</td>
<td></td>
</tr>
<tr>
<td>C) mov a, r0</td>
<td></td>
</tr>
<tr>
<td>D) mov a, #0</td>
<td></td>
</tr>
<tr>
<td>E) mov a, P0</td>
<td></td>
</tr>
</tbody>
</table>

3. Write one or more instructions which uses logical operations to do the same thing that the following instruction does. `setb P1.7`

4. How many machine cycles does the following sequence require?

```
mov r3, #9
Lp: djnz r3, Lp
```

5. Explain how a program written using subprograms can take up less memory space than a program written without them.

6. Write a short sequence of assembly instructions which will copy bit 1 of Port 0 to bit 7 of Port 1.

7. Write a short sequence of assembly instructions to invert all of the bits at memory location 44h.
8. Write a *macro* which does an 8-bit left shift of a register argument. You should shift a zero in from the right. A sample line to use your macro is given by: `ShiftLeft R1`

9. Show what is in registers R0 to R3, the accumulator, and internal memory locations 8, 9, 10, 11, and 12 when the program below completes. Assume register bank 0.

```
Main SEGMENT CODE
CSEG at 0h
  ljmp Start
RSEG Main
Start:  mov sp, #8
        mov a, #0aah
        push acc
        mov r0, #0
        mov r1, #1
        mov r2, #2
        push 1
        mov r3, #3
        push 3
        push 2
        pop 0
        pop acc
        pop 1
        pop 3
Last:   sjmp Last
END
```

10. Assume that there are 8 LEDs connected to port 1. The LEDs are on if the bit is zero and off if it is one. Write a program which will turn on each LED in sequence beginning with bit 0. The LEDs should circulate 0 to 7 forever.