1. (a) Assume the register A has packed BCD (59H), write a program to convert packed BCD to two ASCII numbers and place them in R2 and R6. (hint: using ANL, ORL, SWAP) and (b) make a “list file” consisting of line number, ROM locations, Opcodes, and Instructions. (The program starts at 0000H) (20 points)

<table>
<thead>
<tr>
<th>Line</th>
<th>ROM locations</th>
<th>Opcodes</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000</td>
<td>ORG 0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0000</td>
<td>7459</td>
<td>MOV A, #59H</td>
</tr>
<tr>
<td>3</td>
<td>0002</td>
<td>FA</td>
<td>MOV R2, A ; A=59H</td>
</tr>
<tr>
<td>4</td>
<td>0003</td>
<td>540F</td>
<td>ANL A, #0FH ; A=09H</td>
</tr>
<tr>
<td>5</td>
<td>0005</td>
<td>4430</td>
<td>ORL A, #30H ; A=39H</td>
</tr>
<tr>
<td>6</td>
<td>0007</td>
<td>FE</td>
<td>MOV R6, A ; R6=39H</td>
</tr>
<tr>
<td>7</td>
<td>0008</td>
<td>EA</td>
<td>MOV A, R2 ; A=59H</td>
</tr>
<tr>
<td>8</td>
<td>0009</td>
<td>54F0</td>
<td>ANL A, #0F0H; A=50H</td>
</tr>
<tr>
<td>9</td>
<td>000B</td>
<td>C4</td>
<td>SWAP A ; A=05H</td>
</tr>
<tr>
<td>10</td>
<td>000C</td>
<td>4430</td>
<td>ORL A, #30H ; A=35H</td>
</tr>
<tr>
<td>11</td>
<td>000E</td>
<td>FA</td>
<td>MOV R2, A ; R2=35H</td>
</tr>
<tr>
<td>12</td>
<td>000F</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>

2. Read and test P1 to see whether it has the value A3H. If it does, clear 5 RAM locations starting at RAM address 78H; otherwise, convert 23H to decimal. (20 points)

```
MOV P1, #0FFH ; P1 is an INPUT
MOV A, P1     ; Read Data from P1
CJNE A, #A3H, Next  ; if A!=A3H, then jump
CLR A         ; clear 5 RAM locations
MOV R1, #78H
MOV R2, #5
AGAIN: MOV @R1, A
INC R1
DJNZ R2, AGAIN
Next: MOV A, #23H
MOV B, #10
DIV AB
MOV R3, B
MOV R4, A
```
2-1. Read and test P1 to see whether it has the value A3H. If it does NOT, clear 5 RAM locations starting at RAM address 78H; otherwise, convert A3H to decimal. (20 points)

```
MOV P1, #0FFH ; P1 is an INPUT
MOV A, P1  ; Read Data from P1
CJNE A, #A3H, Next ; if A!=A3H, then jump
MOV B, #10
DIV AB
MOV R3, B
MOV B, #10
DIV AB
MOV R4, B
MOV R5, A
Next:  CLR A   ; clear 5 RAM locations
       MOV R1, #78H
       MOV R2, #5
AGAIN: MOV @R1, A
       INC R1
       DJNZ R2, AGAIN
```

3. (a) Write a program to get the x value (ranging from 0 to 9) from P1 and send $x^2+x+1$ to ACC continuously using indexed addressing mode, and (b) what is the state of the P bit in the PSW after execution of each iteration. (20 points)

```
ORG 0    ; ROM locations starts from 0000H
MOV DPTR, #300H  ; DPTR is a pointer starting from
MOV P1, #0FFH  ; Make P1 as an INPUT
Back:  MOV A, P1   ; get x from P1
       MOVC A, @A+DPTR ; get $x^2+x+1$ from table
       SJMP Back
ORG 300H
Table: DB 1, 3, 7, 13, 21, 31, 43, 57, 73, 91
END
```

<table>
<thead>
<tr>
<th>ACC Binary</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0000 0001</td>
<td>1</td>
</tr>
<tr>
<td>3 0000 0011</td>
<td>0</td>
</tr>
<tr>
<td>7 0000 0111</td>
<td>1</td>
</tr>
<tr>
<td>13 0000 1011</td>
<td>1</td>
</tr>
<tr>
<td>21 0001 0101</td>
<td>1</td>
</tr>
<tr>
<td>31 0001 1111</td>
<td>1</td>
</tr>
<tr>
<td>43 0010 1011</td>
<td>0</td>
</tr>
<tr>
<td>57 0011 1001</td>
<td>0</td>
</tr>
<tr>
<td>73 0100 1001</td>
<td>1</td>
</tr>
<tr>
<td>91 0101 1011</td>
<td>1</td>
</tr>
</tbody>
</table>
4. An 8051 subroutine is shown below (20 points)

   \[\text{SUB: MOV R0, #20H}\]
   \[\text{Loop: MOV @R0, #0}\]
   \[\text{INC R0}\]
   \[\text{CJNE R0, #80H, Loop}\]
   \[\text{RET}\]

(a) What does this subroutine do?

   \textit{This subroutine is to clear the RAM locations from 20H to 7FH.}

(b) In how many machine cycles does each instruction execute?

\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
MC & Bytes & Opcodes \\
\hline
SUB: MOV R0, #20H & 1 & 2 & 78H, 20H \\
Loop: MOV @R0, #0 & 1 & 2 & 76H, 00H \\
INC R0 & 1 & 1 & 08H \\
CJNE R0, #80H, Loop & 2 & 3 & B8H, 80H, FAH \\
RET & 2 & 1 & 22H \\
\hline
\end{tabular}
\end{center}

\textit{Hint: related address= - 6 = FAH (Jump back)}

(c) How many bytes long is each instruction

   \textit{Shown as above.}

(d) Convert the instructions to Opcodes

   \textit{Shown as above.}

(e) How long does this subroutine take to execute? (assume 12MHz operation)

\[
\frac{12\text{MHz}}{12}=1\text{MHz} \Rightarrow 1/1\text{MHz}=1\mu s \text{ for each MC}
\]

\[
80\text{H}-20\text{H}=60\text{H} = 96_{(10)}
\]

\[
(4\times96)+1+2=387 \text{ MCs} \Rightarrow 387\mu s=0.387\text{ms}
\]

5. Find the results at points (1), (2), and (3) in the following code (5 points)

   CJNE A, #50, not_equal
   \[\text{...... ; point (1)}\]
   not_equal: JC Next
   \[\text{...... ; point (2)}\]
   Next: \[\text{...... ; point (3)}\]

   \textit{Point (1): A=50; Point (2): A>50; Point (3): A<50}
6. Observe the following, noting the role of the OV flag for R1, R2, and R3 (5 points)

\[
\begin{align*}
&\text{MOV } A, \#-128 \\
&\text{ADD } A, \#-2 \\
&\text{MOV } R1, A \\
&\text{MOV } A, \#-2 \\
&\text{ADD } A, \#-5 \\
&\text{MOV } R2, A \\
&\text{MOV } A, \#7 \\
&\text{ADD } A, \#18 \\
&\text{MOV } R3, A \\
&\text{MOV } A, \#-128 \quad ; A=-128 \\
&\text{ADD } A, \#-2 \quad ; A=-130 \\
&\text{MOV } R1, A \quad ; R1=-130 \rightarrow \text{OV}=1 \\
&\text{MOV } A, \#-2 \quad ; A=-2 \\
&\text{ADD } A, \#-5 \quad ; A=-7 \\
&\text{MOV } R2, A \quad ; R2=-7 \rightarrow \text{OV}=0 \\
&\text{MOV } A, \#7 \quad ; A=7 \\
&\text{ADD } A, \#18 \quad ; A=25 \\
&\text{MOV } R3, A \quad ; R3=25 \rightarrow \text{OV}=0
\end{align*}
\]

7. Find the time delay for the delay subroutine shown below, if the system frequency is 12MHz. (10 points)

\[
\begin{align*}
\text{Delay: } &\text{MOV } R1, \#100 \\
\text{Back: } &\text{MOV } R2, \#200 \\
\text{Again: } &\text{MOV } R3, \#300 \\
\text{Here: } &\text{NOP} \\
&\text{NOP} \\
&\text{DJNZ } R3, \text{Here} \\
&\text{DJNZ } R2, \text{Again} \\
&\text{DJNZ } R1, \text{Back} \\
&\text{RET}
\end{align*}
\]

\[\text{[For Loop R3]}\]
\[
\begin{align*}
&\text{Delay: } \text{MOV } R1, \#100 \\
&\text{Back: } \text{MOV } R2, \#200 \\
&\text{Again: } \text{MOV } R3, \#300 \\
&\text{Here: } \text{NOP} \\
&\text{NOP} \\
&\text{DJNZ } R3, \text{Here} \\
&\text{DJNZ } R2, \text{Again} \\
&\text{DJNZ } R1, \text{Back}
\end{align*}
\]

\[\text{(1+1+2)*300*200*100=24,000,000}\]
RET

[For Loop R2]
Delay: MOV R1, #100
Back: MOV R2, #200
Again: MOV R3, #300
Here: NOP
      NOP
      DJNZ R3, Here
      DJNZ R2, Again
      DJNZ R1, Back
      RET

[For Loop R1]
Delay: MOV R1, #100
Back: MOV R2, #200
Again: MOV R3, #300
Here: NOP
      NOP
      DJNZ R3, Here
      DJNZ R2, Again
      DJNZ R1, Back
      RET

[Others]
Delay: MOV R1, #100
Back: MOV R2, #200
Again: MOV R3, #300
Here: NOP
      NOP
      DJNZ R3, Here
      DJNZ R2, Again
      DJNZ R1, Back
      RET

The total MCs are 24,060,303 $\Rightarrow$ 24,060,303 $\mu$s = 24.060303 seconds