1

1. Decode the following AVR binary code located in the program memory.
 What are the registers involved?

 r18,r19,r20
 What are their values before and after the program finishes?

 LDI r18   1 > 2 > 4 > 8 > 16 > 32 > 64 > 128 > 256
 LDI r19   1 > 2 > 4 > 8 > 16 > 32 > 64 > 128 > 256
 LDI r20   8 > 7 > 6 > 5 > 4 > 3 > 2 > 1   > 0

 You will need to pay attention to the endianess of the coding.
 (The dashes are not part of the code but are added to increase readability)

Starting program memory address is 0x0100

	LOW BYTE	HIGH BYTE
0x0100:	0010-0001	1110-0000
0x0101:	0011-0001	1110-0000
0x0102:	0100-1000	1110-0000
0x0103:	0011-0010	0000-1111
0x0104:	0010-0011	0010-1111
0x0105:	0100-1010	1001-0101
0x0106:	1110-0001	1111-0111

```
1110 -0000 - 0010-0001   LDI r18 0x01   1 > 2 > 4 > 8 > 16 >
32 > 64 > 128 > 256
1110 -0000 - 0011-0001   LDI r19 0x01   1 > 2 > 4 > 8 > 16 >
32 > 64 > 128 > 256
1110 -0000 - 0100-1000   LDI r20 0x08   8 > 7 > 6 > 5 > 4  >
3 > 2  > 1   > 0
0000-11 11 - 0011-0010   L1: add r19, r18
0010-11 11 - 0010-0011   move r18,r19
1001-0101 - 0100-1010    dec r20
1111-01 11 - 1110-0 001  BRNE L1
```

2

2. Indicate the value loaded into R30, R31, and R20 in the following program:

```
            .ORG 0x0
                        LDI R30, LO8(OUR_DATA )
                        LDI R31, HI8(OUR_DATA)
                        LPM  R20, Z
            .ORG 0x0524
            OUR_DATA: .byte 20, ‘$’, ‘5’
```

r30 0x24 r31 0x05 r20 $ or 0x24

3

3. Write a program to read the following message from program ROM and place it in data RAM starting at 0x200:

```
            .ORG 0x0500
MYDATA: .asciz “Will artificial intelligence rule human?”
```

```
LDI R30, lo8(MYDATA)
LDI R31, hi8(MYDATA)
LDI R28, lo8(0x200)
LDI R29, hi8(0x200)
 // repeal while input not 0, read , write and then check
L:  LPM R16, Z+
     ST Y+, R16
     CPI R16, 0
```

```
        BRNE L
```

4

4. Write a program that calculates the checksum of the values at location 0x00D5 to 0x0300 of EEPROM.

```
CLR R20  /// my checksum
LDI R17, 0x00
LDI R16, 0xD5
LDI R18, 0 // used for ADDC  for R17
LOOP:
       \\ put R17 and R16 into EEPROM registers
       \\ tell EEPROMG to  REAAD
       \\ pull data from EEPROM into R21

        ADD R20, R21

        \\ move my R16 and R17 forward.

        INC R16

        ADC R17, R18

        CPI R17 , 0x03

        BREQ GETOUT

        JMP LOOP

GETOUT:// R16  is the

    /// now calculate the checksum

         NEG R16
```

5

5. In each of the following cases perform checksum calculation to see if data is corrupted or not.

(a) Data=62,F3, and 15; $checksum$ =72

this is corrupted

(b) Data=50,88, $3C, and$8E; checksum=$6D

this is corrupted

(c) Data=0, 0, 0, 0, 0, 0; checksum=0

not corrupted

(d) Data=1,-1,1,-1,1,-1; checksum=1

this is corrupted