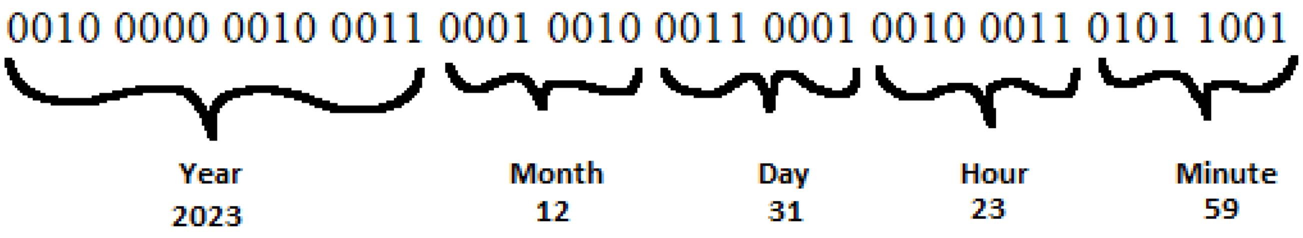
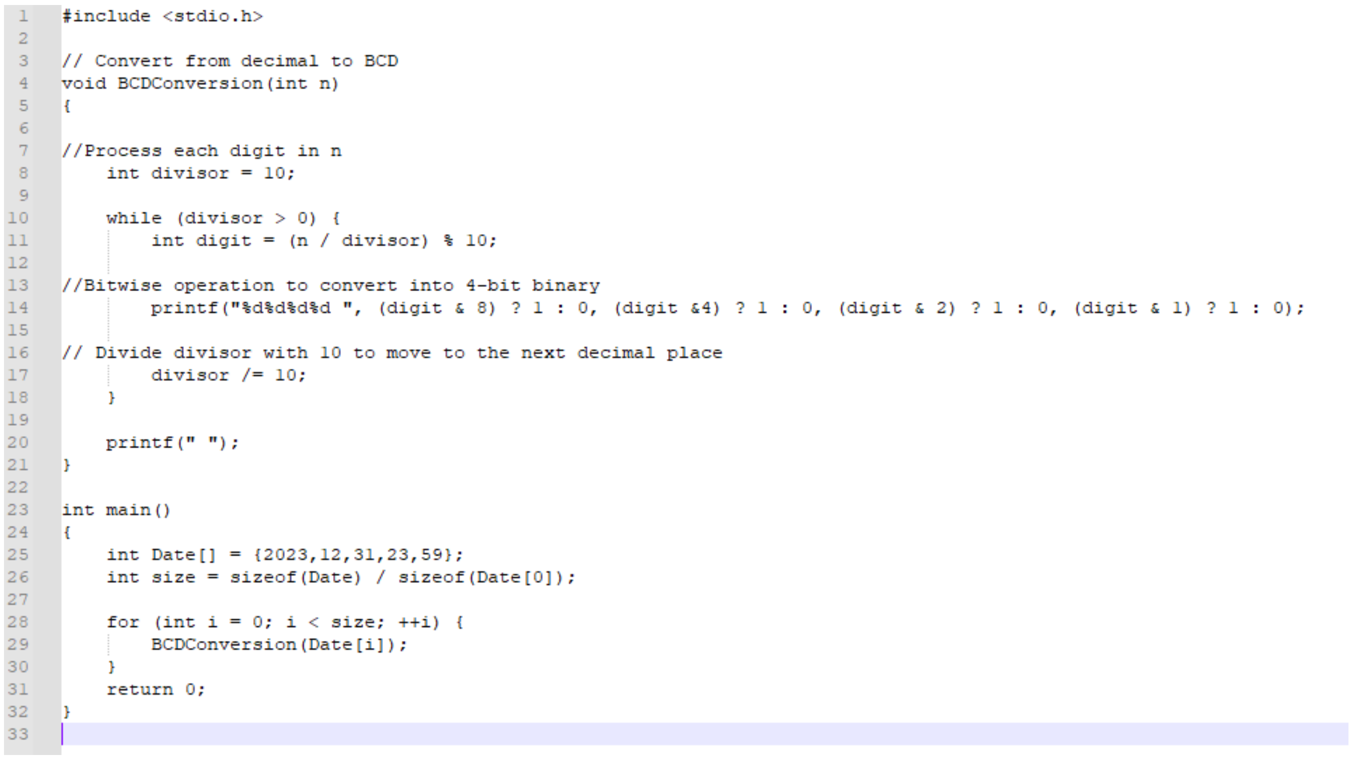
Part 1

1)

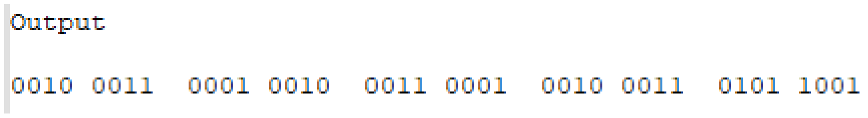
It can be represented in BCD as demonstrated below, where first 16 bits are the year, the next 8-bits are the months, follow by days, hours and minutes in 8 bit sections. The year in the example below is YY= 2023, MM=12, DD=31, HO=23, MI=59.



This code written in C and converts from decimal to BCD. Because we want to minimize the memory usage, i am only using the last two digits in the year 2023, i.e 23. This is done by setting the divisor to 10. In this example where we look a the code going through array element 0 (2023), the code runs through the while loop the first time and calculates the int digit=((2023/10)%10)=2.3 which becomes 2 because it is an int. And afterwards it goes through the bitwise operator which in the first iteration outputs 0010. In second iteration the divisor it divided with 10, so it becomes 1. The "int digit now becomes ((2023/1)%10)=3, which results as 0011 in the bitwise operator. Hence year 2023(23) is stored as 8-bit BCD code, 0010 0011.



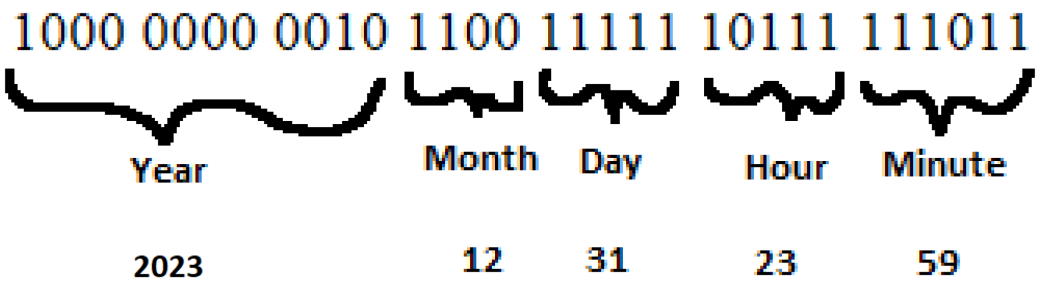
```
#include <stdio.h>

// Convert from decimal to BCD
void BCDConversion(int n)
{

//Process each digit in n
    int divisor = 10;

    while (divisor > 0) {
        int digit = (n / divisor) % 10;

//Bitwise operation to convert into 4-bit binary
        printf("%d%d%d%d ", (digit & 8) ? 1 : 0, (digit &4) ? 1 : 0, (digit & 2) ? 1 : 0, (digit & 1) ? 1 : 0);

// Divide divisor with 10 to move to the next decimal place
        divisor /= 10;
    }

    printf(" ");
}

int main()
{
    int Date[] = {2023,12,31,23,59};
    int size = sizeof(Date) / sizeof(Date[0]);

    for (int i = 0; i < size; ++i) {
        BCDConversion(Date[i]);
    }
    return 0;
}
```



```
Output

0010 0011  0001 0010  0011 0001  0010 0011  0101 1001
```

2)

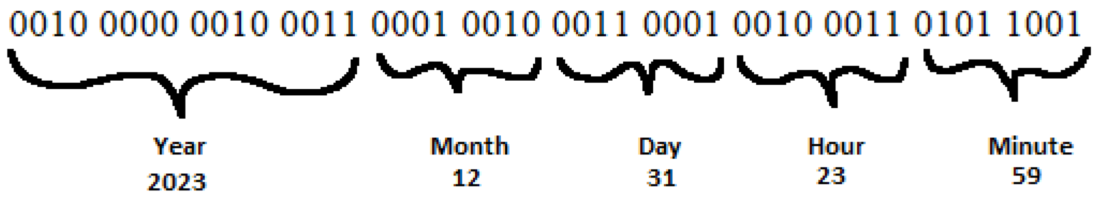
5 integer data structure



The year uses 2 bytes and the rest all use one byte each, which is 6 bytes.

$6\,\text{byte}\cdot 60\cdot 24\cdot 365 = 3153600\ \text{byte}$

BCD representation



$6\,\text{byte} \cdot 60 \cdot 24 \cdot 365 = 3153600\ \text{byte}$

The BCD representation and binary use the same number of bytes in a year.

Part 2

1) What measurements should be available for the motor?

A rotary encoder could be used to determine the speed and which way the motor is spinning.
A temperature sensor to be sure that the motor isint running to warm.
A voltmeter to measure voltage across a shunt resistor to determine the current.

2) What I/O modules should the PLC have?

In:
1. digital encoder 1
2. digital encoder 2
3. Analog temperature measurement
4. Analog voltmeter (to measure current with a shunt resistor)

Out:
1. Vcc (power to motor)
2. Encoder powersupply

If the digital encoder uses optical sensors it is nessasary to use two of them in order to determine which way the motor is spinning.

Part 3

1)

Speed

$$\frac{0.8}{0.05} + 1 = 17.00000000$$

$$solve\left(2^{x}=17\right) = \frac{\ln(17)}{\ln(2)} \xrightarrow{\text{at 5 digits}} 4.0874$$

In regular binary notation we only need 5-bits to represent the 17 values of speed.

If we were to use BCD notation, speed will use 8-bits.

Position

$$\frac{3}{0.001} + 1 = 3001.000000$$

$$solve\left(2^{x}=3001\right) = \frac{\ln(3001)}{\ln(2)} \xrightarrow{\text{at 5 digits}} 11.551$$

In regular binary notation we only need 12-bits to represent the 3001 values of speed.

If we use BCD notation, speed will use 16-bits.

I will choose regular binary notation, because it utilizes less memory usage in this case.
I would use a 12-bit ADC because that is the resolution of "position", which is the highest resolution needed.

2)

The total memory usage needed is 5+12 bits=17bits.