Sets – Explicit & Implicit Representations

The base type for two sets is even numbers from 1 to 50
    If setA has the following elements:
    (4, 48, 32, 12, 40, 22, 18, 36, 6, 34, 14, 26, 16)
    If setB has the following elements:
    (20, 4, 38, 48, 30, 32, 10, 12, 24, 42, 40, 44, 8)

a. Show a bit vector representation for each of the sets
setA:
    0000000000000001000000010001010100000100010001010 10
    100000101000
setB:
    0000000000000001000101010100000101000001000100000 00
    101010001000
b. Show a bit vector for the set setA U setB (union)
0000000000000001000101010101010100010101010101010101 01
000
c. Show a bit vector for the set setA ∩ setB (intersection)
0000000000000001000000010000000100000000000000000100 000001
000
d. Show a bit vector for the set setA - setB (difference)
0000000000000000000000000101000000010001000101010000 000100
000

If setA has the following elements:
   (4, 48, 32, 12, 40, 22, 18, 36, 6, 34, 14, 26, 16)

If setB has the following elements:
   (20, 4, 38, 48, 30, 32, 10, 12, 24, 42, 40, 44, 8)

a. Show an implicit representation of setA and setB:
setA: P(x)= {x | x is a even natural number less than 50}

setB: P(x)= {x | x is a even natural number less than 50}

b. Show implicit representation of setA U setB:
setA ∪ setB =
{4,48,32,12,40,22,18,36,6,34,14,26,16,20,38,30,10,24,42,44,8}

c. Show implicit representation of setA ∩ setB:
setA ∩ setB = {4,48,32,12,40}

d. Show implicit representation of setA - setB:
setA - setB = {22,18,36,6,34,14,26,18}

Manipulate a Binary File

- In a small program –
  - Implement a method to randomly generate 25,000 (integer) numbers in the range between 1000 and 9999
    - Save the generated numbers as a **binary data** file
    - Save the same generated numbers as an ASCII text file
    - Compare and contrast the two files. What did you notice?
  - Implement another method to read (process) a binary data file of similar information
    - Ensure that you can read the content of the binary data file correctly
    - My copy of the data file, with a different set of numbers
- In several paragraphs, discuss your learning experience
  - Compare & contrast -
    - File sizes between the binary vs. ASCII files
    - How easy (or hard) is it to read these files
    - Pros vs. cons between the two formats

Learning Experience:

I found the first part of the assignment to be pretty straight forward. I didn't really struggle too much and your lecture ppt along with some YouTube videos cleared everything up for me. I found the coding section to be straight forward as well, except for the reading of the Binary File. I was able to read all of the integers but not in the correct order. Not sure if it's due to the binary text changing the original order of the integers. For comparing the binary file vs the ascii text file, the binary file came in at 100 KB while the ascii text file was 125 KB. The binary file took about 5 or 6 seconds to scroll through the whole 25,000 data values. The ascii text file took considerably longer, so long that I stopped scrolling midway through. I was seriously thrown off at how much shorter it took to go through the dat file instead of the text file. Now reading the file is a whole another ball game. It is considerably harder to read the binary file than it is the ascii text file. The text file is simple integers while the dat file is binary data which is foreign language to anyone that doesn't know how to read binary data. The pros that I see with binary data include been able to traverse huge amounts of data quickly and efficiently while also keeping the file size smaller than it would be if it were a text file. The pros that I see with the text file include being able to understand and read your text data clearly and effectively. It allows for cleared information to be understood rather than needing a converter of some sort. The cons with binary would be being difficult to understand and read for some and always needing to convert the data to be read. The cons with text file would be being difficult to traverse when there is huge amounts of data and having a bigger file size compared to a dat file.