

Software design

1. Open the same file twice since we are going to read the same data in reverse order.
2. Use `lseek()` to find the file's data size and create a new file to be the output file.
3. Setting variables. We need
 - Tracker for file1 and file2, which will use for comparison.
 - 2 char array called buffer to store file data
 - 1 integer array to store the number of different characters; 2 char array to store the different characters from file 1 and file 2.
 - 1 tracker to fill the data into those difference arrays.
4. Since we can only declare the array size up to 32, it's impossible to store all the results into an array when you have file data that is over 32 characters. Therefore, we need to compare the data from file1 and file2 character by character and then only store the output when there is a difference occurred.
- 5 & 6. Display the output to file and console. Use for loop to display the arrays that store which number occurs difference and what characters are different.

Issues that I have encountered:

1. I can't use arrays to compare all data because of the limited size; how to solve it?

Solution:

Since there is a limitation to the array size, so we need to compare the data first and only store the different results to the limited size array.

2. When setting `read(fd1, &buffer1, 1);` && `read(fd2, &buffer2, 1);` and then write `if (buffer1 == buffer2)`, why the if-statement doesn't work?

Solution:

In this read function, we set `&buffer`, which means we are reading data from `fd1` and store the "address" to "buffer." Therefore, when we use `if (buffer1 == buffer2)`, this is comparing the data's Address, not the value. If we want to compare the data value, we need to add a pointer to become `if (*buffer1 == *buffer2)` to point to address to retrieve the value.

3. How to use `write()` to print out values in array?

Solution:

```
ssize_t write(int fd, const void *buf, size_t nbyte);
```

the above definition shows that the middle parameter of `write()` takes pointer. So, if we are going to print out the array value, we need to reference the variable so that the write function will find the value based on the pointing address.

```
storeN = store_number[i] + '0';  
write(fd3, &storeN, sizeof(storeN));  
write(fd3, " ", 1);
```

4. Why `write()` can't print two digits numbers?

Solution:

`Write()` can only print a single-digit CHAR value, so if we need to print a single-digit integer, we need to add '0' to convert an integer to char type and then print it out. As to two digits or more digits number, we need to use % and / operations to get ones, ten, and hundred's digit, and then do the same operation when dealing with a single digit.

