Data Structures

Objective

- At the end of the meeting, students should be able to:
  - Enumerate basic operations on files

Text files

- Arrays and structures are normally stored in the main memory – these data are released when the program terminates

- Data are often stored more permanently in files, e.g., your C programs are stored as plain text files in secondary storage (hard disks, floppies, CDs, flash drives, etc)

File redirection and pipes

- Programs normally read input from the keyboard, and output to the screen
- Output can be stored to a text file using >

  $ prog > outfile
- Output can be appended to a text file using >>

  $ prog >> outfile
- Input can be made to come from a text file using <

  $ prog < infile
- The output of one program can be made as the input of another program using the "pipe" |

  $ prog1 | prog2

Useful predefined functions for text file processing

- **fopen**(filename, mode) -- try to open the file, where mode = “r” for read, “w” for write, “a” for append
- **feof**(filepointer) – test if we have reached the end of the file
- **fgets**(string, maxlength, filepointer)
- **fprintf**(filepointer, format, ….)
- **fclose**(filepointer) – close the file after use

Sample code to read a text file

```
FILE *fp;
char oneline[80];
int linecounter = 0;
if ( ( fp = fopen("filedemo.c", "r") ) == NULL ) {
  printf("error: unable to open the file\n");
}
else { // read and print each line until the end of the file
  while ( ! feof(fp) ) {
     fgets(oneline, 80, fp);
     printf("%4d %s", ++linecounter, oneline);
  }
  fclose(fp);
}
```

Text file exercises

- Encrypt/decrypt a text file for security or privacy purposes using "simple" algorithms like replacing letters
  - **Caesar cypher** (k=1): a-b, b-c, …, y-z, z-a
  - **Atbash cypher**: a-z, b-y, c-x, …, m-n

- Merge two sorted text files to create a single sorted file (Note: several files may be opened simultaneously)