Notes for LAB 3

Text Files. Reading and Writing

Reading from a Text File

To read from a text file:

1) First you need to open the file for reading:

FILE ***input_file**=fopen("data.txt", "**r**"); /* opens file data.txt for reading */

2) Read data from the file:

- **fscanf()** can be used. **fscanf()** reads from the input file in the same way **scanf()** reads from the standard input stream (keyboard).
- **fscanf()** needs one more argument which specifies the input file.

fscanf(**input_file**,"%d", &n); /* reads **the next** integer from the file specified by input_file and stores it in the integer variable n */

To read more integers you may use the above statement in a loop. The integers have to be separated by white spaces.

If you have read some values from the file and you want to start reading again from the beginning use first:

rewind(input_file); //moves the file position pointer at the beginning of the file

then read the values with scanf()

. **3) When you are done you have to close the file:**

fclose(input_file);

- **#include <stdio.h> for file I/O**

Writing in a Text File

To write in a text file:

1) First you have to open the file for writing.

FILE ***output_file**=fopen("output.txt", "**w**"); /* opens file output.txt for writing; if the file does not exist it is created; if it does exist its contents is wiped out. If the name of the file is stored in some string s, use s as the first argument of fopen()*/

2) Write the data into the file:

- **fprintf()** can be used. **fprintf()** writes into the output file in the same way **printf()** writes into the standard output stream (on the screen).
- **fprintf()** needs one more argument which specifies the output file.

fprintf(**output_file**,"%d\n", n); **/* prints integer n followed by newline */**

3) When you are done you have to close the file.

fclose(output_file);

- **#include <stdio.h> for file I/O**