

CS 442/542 Homework 2

I/O Manager

- Implement the I/O manager described in the following header file. This module will provide the **sole** access point for reading characters from the source program.
- “Due” Monday February 20

I/O Manager General Functionality

- Open source File
- Open a listing file if a listing file name is provided
- Close file(s)
- Return the next source character
- Write a indicator marker at a specified column
 - Used to mark the position of an error
- Write a message
 - Used to write an error message
- Return the current line number
- Return the current column number

IOMngr.h

```
#include <stdlib.h>  
#include <stdio.h>
```

```
#define MAXLINE 1024
```

```
int openFiles(char * sourceName, char * listingName);
```

```
void closeFiles();
```

```
char getNextSourceChar();
```

```
void writeIndicator(int column);
```

```
void writeMessage(char * message);
```

```
int getCurrentLineNum();
```

```
int getCurrentColumnNum();
```

I/O Manager

- `openFiles(char * sourceName, char * listingName);`
- Open the source file whose name is given in `sourceName`
 - You can assume `sourceName` is assigned a legal `char*` (i.e. a string)
- If `listingName` is not `NULL` open the listing file whose name is given in `listingName`
 - If `listingName` is `NULL`, the output goes to `stdout`
- Return 1 if the file open(s) were successful, otherwise return 0

I/O Manager

- `void closeFiles()`
- Close the source file and the listing file if one was created

I/O Manager

- `char getNextSourceChar()`
- Return the next source char
- This function is also responsible for echoing the lines in the source file to the listing file (if one exists)
- The lines in the listing file should be numbered
- Return EOF when the end of the source file is reached

I/O Manager

- `void writeIndicator(int column)`
- Write a line containing a single '^' character in the indicated column
- If there is no listing file then the current line should be echoed to stdout the first time (for that line) that `writeIndicator` or `writeMessage` is called.

I/O Manager

- `void writeMessage(char * message)`
- Write the message on a separate line

Example with Listing File

1. float y;
2. floam x;
 ^

Unexpected character

3. y = 10;
4. x = 2*y@3;
 ^

Variable not declared
 ^

Unexpected character

Write Example without a Listing File

2. floam x;
^

Unexpected character

4. x = 2*y@3;
^

Variable not declared
^

Unexpected character

I/O Manager

- `int getCurrentLineNum()`
 - Return the current line number
- `int getCurrentColumnNum()`
 - Return the current column number in the current line.

I/O Manager

- Because the line must be echoed before error indicators and messages the implementation must buffer the current line.
- That is IOMngr should have an array of characters that hold the current line. When the last character of the current line is returned getNextSourceChar will need to read another line
- Every character in the line should be returned including the carriage return and newline characters.