

Emily Falkner (emfalkne), Matthew Scott Dexheimer (SDexh), Savannah Puckett (snpucket)

SCANNER.C

```
void init_scanner(char source_name[]);  
  
//This function opens the file and copies the source name to the src_name variable and clears  
the current tokens from memory.  
  
void close_scanner();  
  
//The function closes the file and clears the tokens from memory.  
  
void clear_token_line();  
  
//This function clears all the tokens that are in memory and sets to null  
  
char read_next_line();  
  
//This function reads the next line from the input stream. It also clears the current tokens that  
are in memory. Then reads the line, tokenizes it and returns the char used to determine when  
the end of file is reached.  
  
char *get_file_line();  
  
//This gets the line that was most recently read.  
  
struct Token *get_token_line();  
  
//This function returns the current token linked list  
  
static void read_line(char *last);  
  
//This function is used to convert the file into a char array.  
  
static void rem_token(struct Token *t);  
  
//This function is used recursively removes tokens from the memory  
  
static void tokenize_line();  
  
//This function tokenizes the current line  
  
static struct Token *tokenize_word(int *i, char *ch);  
  
//This function returns a token of the next word in the file
```

```
static int adv_to_valid(char *ch);
//This function advances to the next valid character

static int adv_to_nl(char *ch);
//this function skips the line and goes to the next one

static int interpret_ch(char *dest, char *src);
//This function interprets the character and parses the meaning of the word

static int identify(char ch);
//This function identifies the char as the start of a string, number, or symbol

static int is_letter(char ch);
//If the start of the char is a letter

static int is_number(char ch);
//if the start of the char is a number

static int is_symbol(char ch);
//if the start of the char is a symbol

static int parse_word(char *dest, char *src);
//this function parses the word

static int parse_number(char *dest, char *src);
//this function parses the number

static int parse_symbol(char *dest, char *src);
//This function parses a symbol

static void add_token(struct Token *t);
//This function adds the current token to the list

static void make_sense_token(struct Token *t);
//This function takes the token an puts it with the appropriate t code and t type.
```

PRINT.C

```
void print_line(char line[], char source_name_to_print[], char date_to_print[], int token)  
//this function prints the current line and checks to see if there are more than enough lines on  
the page if so call the print headed and resets the line count.  
  
static void print_page_header(char source_name[], char date[])  
  
//This function prints the page header which includes the page number, source name and the  
date  
  
void print_tokens(struct Token *t, char source_name[], char date[])  
  
//This function prints the tokens that are given at the beginning
```

MAIN.C

```
int main ()  
{  
    src_name  
    date  
    initialize the scanner  
    while ( the end of file is not reached)  
    {  
        print the line  
        print the token  
    }  
    close scanner  
    return  
}  
  
void init_lister(const char *name, char source_file_name[], char dte[])  
{ //used for debugging.  
    timer  
    copy the name into the source file name  
    time (timer)  
    copthe the date and the time  
}
```

WORK DISTRUBTION TABLE

Team Name	Number
Savannah Puckett (snpucket)	2
Matthew Scott Dexhimer (SDexh)	2
Emily Falkner (emfalkne)	2