CD Lab 8 and 9

Name: Paawan Kohli
Reg No:180905416

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

#include "tokenGen.c" // inlcude file with getNextToken()

token cT;
FILE *f, *lex;

void invalid(char* exp, int r, int c) {
	printf("expected '%s' at row %d col %d\n", exp, r, c);
	exit(0);
}

void valid() {
	printf("----------------SUCCESS!---------------\n");
	exit(0);
}

void Program();
void declarations();
void data_type();
void identifier_list();
void ilprime();
void ilpprime();
void statement_list();
void statement();
void assign_stat();
void expn();
void eprime();
void simple_expn();
void seprime();
void term();
void tprime();
void factor();
void relop();
void addop();
void mulop();
void decision_stat();
void looping_stat();
void dprime();

token getNext() {
	return getNextToken(f, lex);
}

void Program() {
	if (strcmp(cT.token_name, "main") == 0) {
		cT = getNext();
		if (strcmp(cT.token_name, "(") == 0) {
			cT = getNext();
			if (strcmp(cT.token_name, ")") == 0) {
				cT = getNext();
				if (strcmp(cT.token_name, "{") == 0) {
					cT = getNext();
					declarations();
					statement_list();
					if (strcmp(cT.token_name, "}") == 0) {
						cT = getNext();
						return;
					}
					else invalid("}", cT.row, cT.col);
				}
				else invalid("{", cT.row, cT.col);
			}
			else invalid(")", cT.row, cT.col);
		}
		else invalid("(", cT.row, cT.col);
	}
	else invalid("main", cT.row, cT.col);
}
```

```
void declarations() {
	char firstofdata_type[20][20] = {"int", "char"};
	int nof = 2;

	int flag = 0;
	for (int i = 0; i < nof; i++)
		if ((strcmp(cT.token_name, firstofdata_type[i])) == 0)
			flag = 1;

	if (flag) {
		data_type();
		identifier_list();
		if (strcmp(cT.token_name, ";") == 0) {
			cT = getNext();
			declarations();
		}
		else invalid(";", cT.row, cT.col);
	}
}

void data_type() {
	if (strcmp(cT.token_name, "int") == 0) {
		cT = getNext();
		return;
	} else if (strcmp(cT.token_name, "char") == 0) {
		cT = getNext();
		return;
	} else invalid("int or char", cT.row, cT.col);
}

void identifier_list() {
	if (strcmp(cT.token_name, "id") == 0) {
		cT = getNext();
		ilprime();
	} else invalid("identifier", cT.row, cT.col);
}

void ilprime() {
	if (strcmp(cT.token_name, ",") == 0) {
		cT = getNext();
		identifier_list();
	}
	else if (strcmp(cT.token_name, "[") == 0) {
		cT = getNext();
		if (strcmp(cT.token_name, "num") == 0) {
			cT = getNext();
			if (strcmp(cT.token_name, "]") == 0) {
				cT = getNext();
				ilpprime();
			}
			else invalid("]", cT.row, cT.col);
		}
		else invalid("number", cT.row, cT.col);
	}
}

void ilpprime() {
	if (strcmp(cT.token_name, ",") == 0) {
		cT = getNext();
		identifier_list();
	}
}

void statement_list() {
	char firstofstatement[20][20] = {"id", "if", "for", "while"};
	int nof = 4;

	int flag = 0;

	for (int i = 0; i < nof; i++)
		if ((strcmp(cT.token_name, firstofstatement[i])) == 0)
			flag = 1;

	if (flag) {
		statement();
		statement_list();
	}
}
```

```
void statement() {
	int flag = 0;


	char firstofassignstat[20][20] = {"id"};
	int nofa = 1;

	for (int i = 0; i < nofa; i++)
		if ((strcmp(cT.token_name, firstofassignstat[i])) == 0)
			flag = 1;





	char firstofdecisionstat[20][20] = {"if"};
	int nofd = 1;

	for (int i = 0; i < nofd; i++)
		if ((strcmp(cT.token_name, firstofdecisionstat[i])) == 0)
			flag = 2;




	char firstofloopingstat[20][20] = {"for", "while"};
	int nofl = 2;

	for (int i = 0; i < nofl; i++)
		if ((strcmp(cT.token_name, firstofloopingstat[i])) == 0)
			flag = 3;



	if (flag == 1) {
		assign_stat();

		if (strcmp(cT.token_name, ";") == 0)
			cT = getNext();
		else invalid(";", cT.row, cT.col);
	}
	else if (flag == 2)
		decision_stat();
	else if (flag == 3)
		looping_stat();
}




void looping_stat() {
	if (strcmp(cT.token_name, "while") == 0) {
		cT = getNext();
		if (strcmp(cT.token_name, "(") == 0) {
			cT = getNext();

			expn();
			if (strcmp(cT.token_name, ")") == 0) {
				cT = getNext();
				if (strcmp(cT.token_name, "{") == 0) {
					cT = getNext();
					statement_list();
					if (strcmp(cT.token_name, "}") == 0) {
						cT = getNext();
						return;
					}
					else invalid("}", cT.row, cT.col);
				}
				else invalid("{", cT.row, cT.col);
			}
			else invalid(")", cT.row, cT.col);
		}
		else invalid("(", cT.row, cT.col);
	}
```

```
	else if (strcmp(cT.token_name, "for") == 0) {
		cT = getNext();
		if (strcmp(cT.token_name, "(") == 0) {
			cT = getNext();
			assign_stat();
			if (strcmp(cT.token_name, ";") == 0) {
				cT = getNext();
				expn();
				if (strcmp(cT.token_name, ";") == 0) {
					cT = getNext();
					assign_stat();
					if (strcmp(cT.token_name, ")") == 0) {
						cT = getNext();
						if (strcmp(cT.token_name, "{") == 0) {
							cT = getNext();
							statement_list();
							if (strcmp(cT.token_name, "}") == 0) {
								cT = getNext();
								return;
							}
							else invalid("}", cT.row, cT.col);
						}
						else invalid("{", cT.row, cT.col);
					}
					else invalid(")", cT.row, cT.col);
				}
				else invalid(";", cT.row, cT.col);
			}
			else invalid(";", cT.row, cT.col);
		}
		else invalid("(", cT.row, cT.col);
	}
	else invalid("while or for", cT.row, cT.col);
}

void dprime() {
	if (strcmp(cT.token_name, "else") == 0) {
		cT = getNext();
		if (strcmp(cT.token_name, "{") == 0) {
			cT = getNext();
			statement_list();
			if (strcmp(cT.token_name, "}") == 0) {
				cT = getNext();
				return;
			}
			else invalid("}", cT.row, cT.col);
		}
		else invalid("{", cT.row, cT.col);
	}
}


void decision_stat()
{
	if (strcmp(cT.token_name, "if") == 0) {
		cT = getNext();
		if (strcmp(cT.token_name, "(") == 0) {
			cT = getNext();
			expn();
			if (strcmp(cT.token_name, ")") == 0) {
				cT = getNext();
				if (strcmp(cT.token_name, "{") == 0) {
					cT = getNext();
					statement_list();
					if (strcmp(cT.token_name, "}") == 0) {
						cT = getNext();
						dprime();
					}
					else invalid("}", cT.row, cT.col);
				}
				else invalid("{", cT.row, cT.col);
			}
			else invalid(")", cT.row, cT.col);
		}
		else invalid("(", cT.row, cT.col);
	}
	else invalid("if", cT.row, cT.col);
}
```

```
void assign_stat() {
	if (strcmp(cT.token_name, "id") == 0) {
		cT = getNext();
		if (strcmp(cT.token_name, "=") == 0) {
			cT = getNext();
			expn();
		}
		else invalid("=", cT.row, cT.col);
	}
	else invalid("identifier", cT.row, cT.col);
}



void expn() {
	simple_expn();
	eprime();
}



void eprime() {
	char firstofrelop[20][20] = {"=", "!", "<", ">"};
	int nof = 4;
	int flag = 0;

	for (int i = 0; i < nof; i++)
		if ((strcmp(cT.token_name, firstofrelop[i])) == 0)
			flag = 1;

	if (flag) {
		relop();
		simple_expn();
	}
}



void simple_expn() {
	term();
	seprime();
}



void seprime() {
	char firstofaddop[20][20] = {"+", "-"};
	int nof = 2;

	int flag = 0;
	for (int i = 0; i < nof; i++)
		if ((strcmp(cT.token_name, firstofaddop[i])) == 0)
			flag = 1;

	if (flag) {
		addop();
		term();
		seprime();
	}
}



void term() {
	factor();
	tprime();
}
```

```
void tprime() {
	char firstofmulop[20][20] = {"*", "/", "%"};
	int nof = 3;
	int flag = 0;

	for (int i = 0; i < nof; i++)
		if ((strcmp(cT.token_name, firstofmulop[i])) == 0)
			flag = 1;

	if (flag) {
		mulop();
		factor();
		tprime();
	}
}



void factor() {
	if (strcmp(cT.token_name, "id") == 0)
	{
		cT = getNext();
		return;
	}
	else if (strcmp(cT.token_name, "num") == 0)
	{
		cT = getNext();
		return;
	}
	else
		invalid("identifier or number", cT.row, cT.col);
}



void relop() {
	char arr[6][10] = {"==", "!=", "<=", ">=", ">", "<"};
	int no = 6;

	int flag = 0;

	for (int i = 0; i < no; i++)
		if ((strcmp(cT.token_name, arr[i])) == 0)
			flag = 1;

	if (flag) {
		cT = getNext();
		return;
	}
	else
		invalid("relational operator", cT.row, cT.col);
}



void addop() {
	char arr[2][10] = {"+", "-"};
	int no = 2;

	int flag = 0;

	for (int i = 0; i < no; i++)
		if ((strcmp(cT.token_name, arr[i])) == 0)
			flag = 1;

	if (flag) {
		cT = getNext();
		return;
	}
	else
		invalid("addition or subtraction operator", cT.row, cT.col);
}
```

```
void mulop() {
	char arr[3][3] = {"*", "/", "%"};
	int no = 3;

	int flag = 0;

	for (int i = 0; i < no; i++)
		if ((strcmp(cT.token_name, arr[i])) == 0)
			flag = 1;

	if (flag) {
		cT = getNext();
		return;
	}
	else invalid("multiplication, division or modulus operator", cT.row, cT.col);
}

int main() {
	printf("Enter filename: ");
	char name[20];
	scanf("%s", name);

	f = fopen(name, "r");
	lex = fopen("lex.txt", "w");

	cT = getNext();
	Program();

	if (strcmp(cT.token_name, "$") == 0)
		valid();
	else
		invalid("EOF", cT.row, cT.col);

	fclose(f);
	fclose(lex);

	return 0;
}
```

~/Desktop/CD-Lab/paawan/lab789/sample1.c - Sublime Text (...

File Edit Selection Find View Goto Tools Project Preferences Help

sample1.c | sample2.c | sample3.c

```c
main()
{
    int a[40], b, c, i;
    c = a + b;
    for (i = 0; i < 5; i = i + 1)
    {
        c = c + 1;
    }
}
```

Line 9, Column 2 | master 9 | Tab Size: 4 | C

student@c35: ~/Desktop/CD-Lab/paawan/lab789

File Edit View Search Terminal Help

```
student@c35:~/Desktop/CD-Lab/paawan/lab789$ ./parser
Enter filename: sample1.c
----------------SUCCESS!---------------
student@c35:~/Desktop/CD-Lab/paawan/lab789$
```

~/Desktop/CD-Lab/paawan/lab789/sample2.c - Sublime Text (...

File Edit Selection Find View Goto Tools Project Preferences Help

sample1.c | sample2.c | sample3.c

```c
main()
{
    int a[40]
    int b, c, i;
    c = a + b;
    for (i = 0; i < 5; i = i + 1)
    {
        c = c + 1;
    }
}
```

Line 4, Column 9 | master 9 | Tab Size: 4 | C

student@c35: ~/Desktop/CD-Lab/paawan/lab789

File Edit View Search Terminal Help

```
student@c35:~/Desktop/CD-Lab/paawan/lab789$ ./parser
Enter filename: sample2.c
expected ';' at row 4 col 2
student@c35:~/Desktop/CD-Lab/paawan/lab789$
```

~/Desktop/CD-Lab/paawan/lab789/sample3.c - Sublime Text (...

File Edit Selection Find View Goto Tools Project Preferences Help

sample1.c | sample2.c | sample3.c

```c
main()
{
    int a[40], b, c, i;
    c = a + b;
    for (i = 0; i < 5; i = i + 1)
    {
        c = c + 1;
        b = ;
    }
}
```

Line 5, Column 14; Saved ~/Desktop/CD-Lab/paawan/lab789/sample3.c (UTF-8)

student@c35: ~/Desktop/CD-Lab/paawan/lab789

File Edit View Search Terminal Help

```
student@c35:~/Desktop/CD-Lab/paawan/lab789$ ./parser
Enter filename: sample3.c
expected 'identifier or number' at row 8 col 7
student@c35:~/Desktop/CD-Lab/paawan/lab789$
```