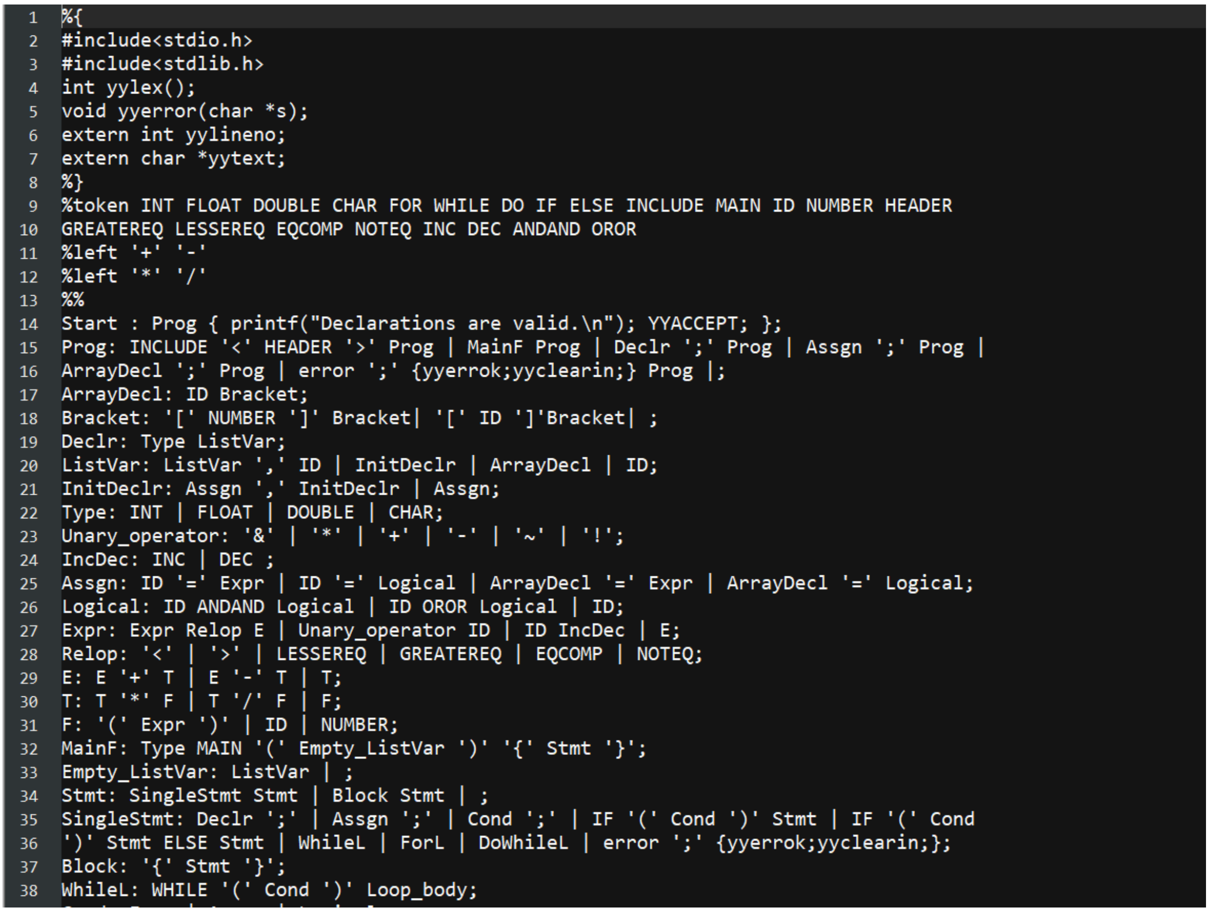
Compiler Design Assignment – 1

Name: Ananya Narayan Bhat

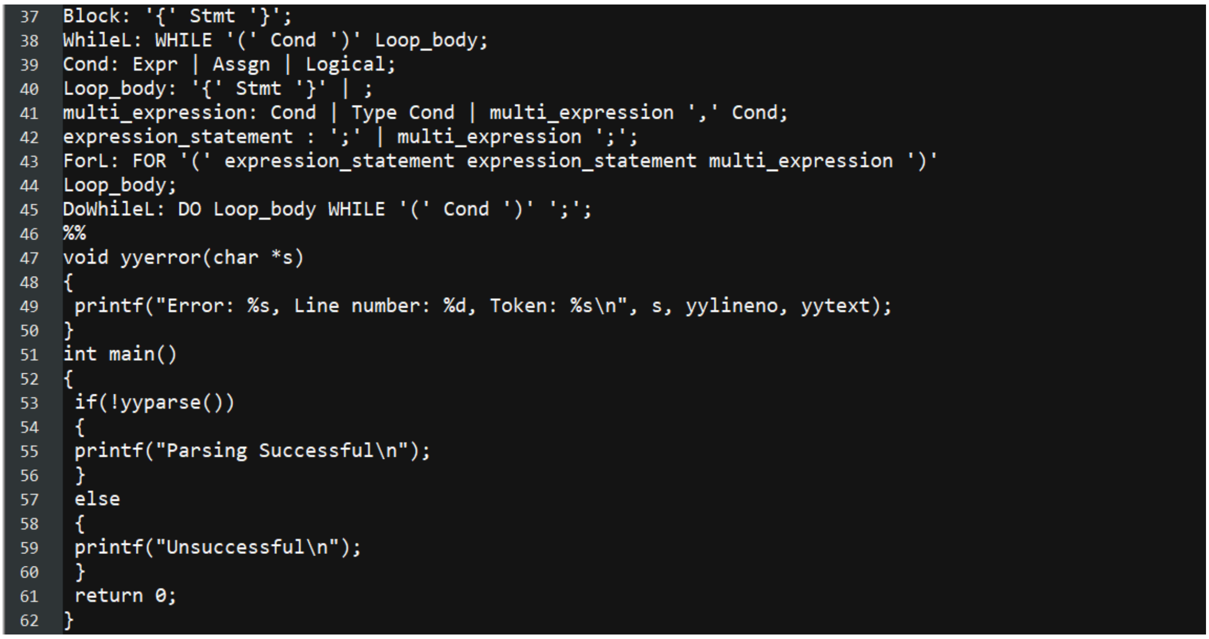
SRN: PES2UG20CS045

SEC: A

parser.y

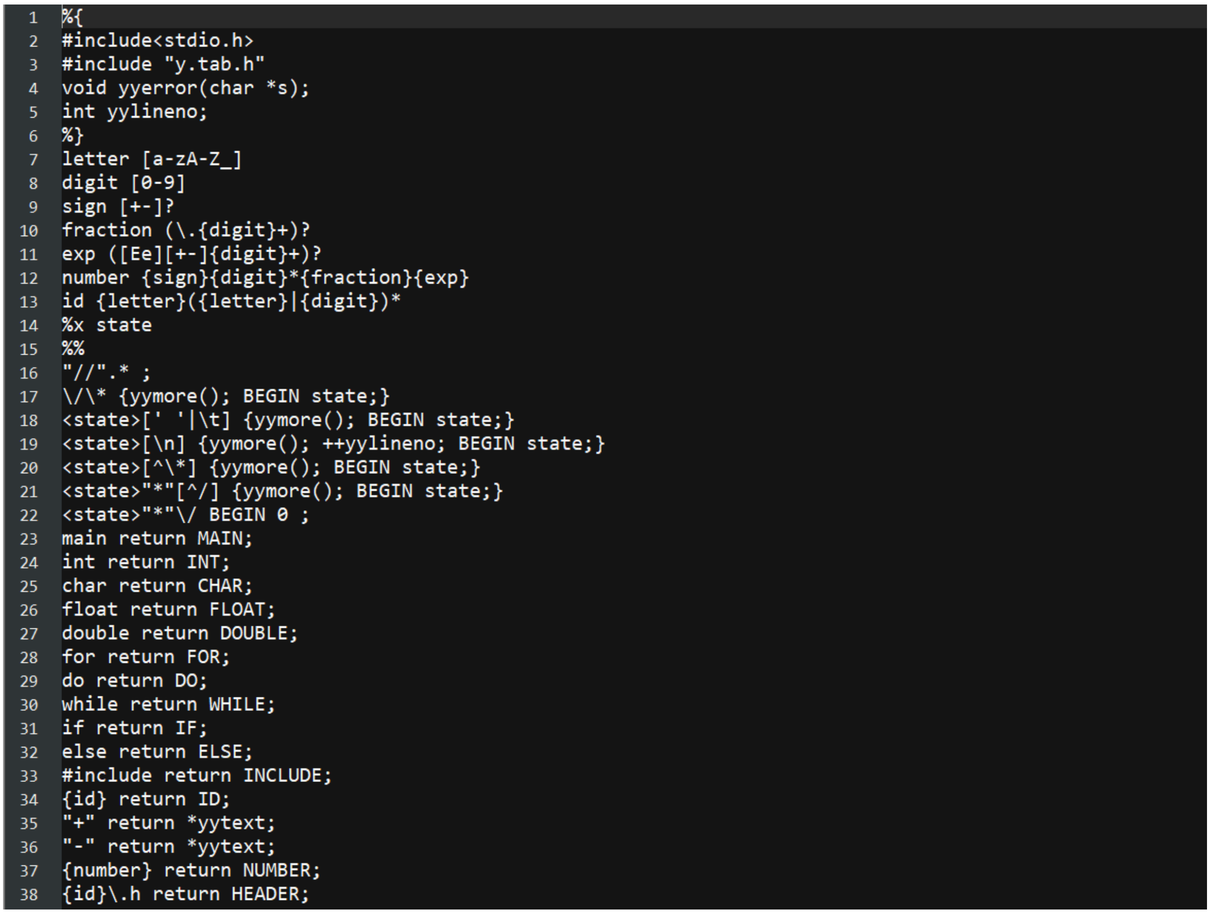


```
%{
#include<stdio.h>
#include<stdlib.h>
int yylex();
void yyerror(char *s);
extern int yylineno;
extern char *yytext;
%}
%token INT FLOAT DOUBLE CHAR FOR WHILE DO IF ELSE INCLUDE MAIN ID NUMBER HEADER
GREATEREQ LESSEREQ EQCOMP NOTEQ INC DEC ANDAND OROR
%left '+' '-'
%left '*' '/'
%%
Start : Prog { printf("Declarations are valid.\n"); YYACCEPT; };
Prog: INCLUDE '<' HEADER '>' Prog | MainF Prog | Declr ';' Prog | Assgn ';' Prog |
ArrayDecl ';' Prog | error ';' {yyerrok;yyclearin;} Prog |;
ArrayDecl: ID Bracket;
Bracket: '[' NUMBER ']' Bracket| '[' ID ']'Bracket| ;
Declr: Type ListVar;
ListVar: ListVar ',' ID | InitDeclr | ArrayDecl | ID;
InitDeclr: Assgn ',' InitDeclr | Assgn;
Type: INT | FLOAT | DOUBLE | CHAR;
Unary_operator: '&' | '*' | '+' | '-' | '~' | '!';
IncDec: INC | DEC ;
Assgn: ID '=' Expr | ID '=' Logical | ArrayDecl '=' Expr | ArrayDecl '=' Logical;
Logical: ID ANDAND Logical | ID OROR Logical | ID;
Expr: Expr Relop E | Unary_operator ID | ID IncDec | E;
Relop: '<' | '>' | LESSEREQ | GREATEREQ | EQCOMP | NOTEQ;
E: E '+' T | E '-' T | T;
T: T '*' F | T '/' F | F;
F: '(' Expr ')' | ID | NUMBER;
MainF: Type MAIN '(' Empty_ListVar ')' '{' Stmt '}';
Empty_ListVar: ListVar | ;
Stmt: SingleStmt Stmt | Block Stmt | ;
SingleStmt: Declr ';' | Assgn ';' | Cond ';' | IF '(' Cond ')' Stmt | IF '(' Cond
')' Stmt ELSE Stmt | WhileL | ForL | DoWhileL | error ';' {yyerrok;yyclearin;};
Block: '{' Stmt '}';
WhileL: WHILE '(' Cond ')' Loop_body;
```

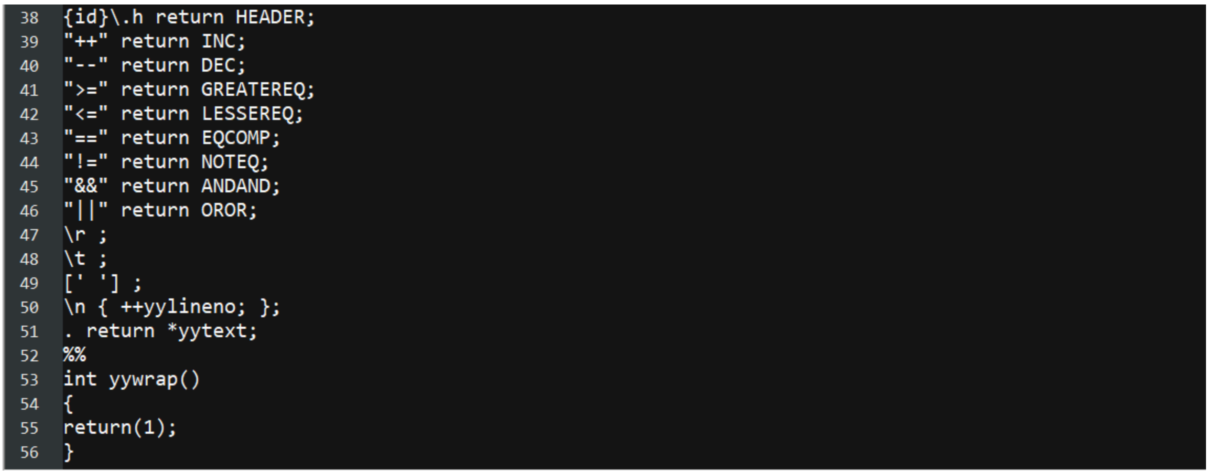


```
Block: '{' Stmt '}';
WhileL: WHILE '(' Cond ')' Loop_body;
Cond: Expr | Assgn | Logical;
Loop_body: '{' Stmt '}' | ;
multi_expression: Cond | Type Cond | multi_expression ',' Cond;
expression_statement : ';' | multi_expression ';';
ForL: FOR '(' expression_statement expression_statement multi_expression ')'
Loop_body;
DoWhileL: DO Loop_body WHILE '(' Cond ')' ';';
%%
void yyerror(char *s)
{
 printf("Error: %s, Line number: %d, Token: %s\n", s, yylineno, yytext);
}
int main()
{
 if(!yyparse())
 {
 printf("Parsing Successful\n");
 }
 else
 {
 printf("Unsuccessful\n");
 }
 return 0;
}
```

lexer.l

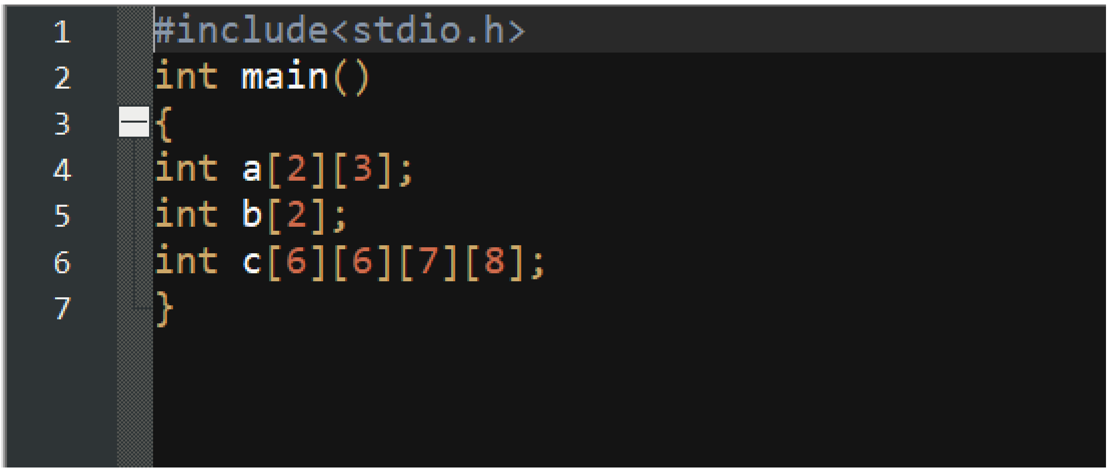


```
%{
#include<stdio.h>
#include "y.tab.h"
void yyerror(char *s);
int yylineno;
%}
letter [a-zA-Z_]
digit [0-9]
sign [+-]?
fraction (\.{digit}+)?
exp ([Ee][+-]{digit}+)?
number {sign}{digit}*{fraction}{exp}
id {letter}({letter}|{digit})*
%x state
%%
"//".* ;
\/\* {yymore(); BEGIN state;}
<state>[' '|\t] {yymore(); BEGIN state;}
<state>[\n] {yymore(); ++yylineno; BEGIN state;}
<state>[^\*] {yymore(); BEGIN state;}
<state>"*"[^/] {yymore(); BEGIN state;}
<state>"*"\/ BEGIN 0 ;
main return MAIN;
int return INT;
char return CHAR;
float return FLOAT;
double return DOUBLE;
for return FOR;
do return DO;
while return WHILE;
if return IF;
else return ELSE;
#include return INCLUDE;
{id} return ID;
"+" return *yytext;
"-" return *yytext;
{number} return NUMBER;
{id}\.h return HEADER;
```



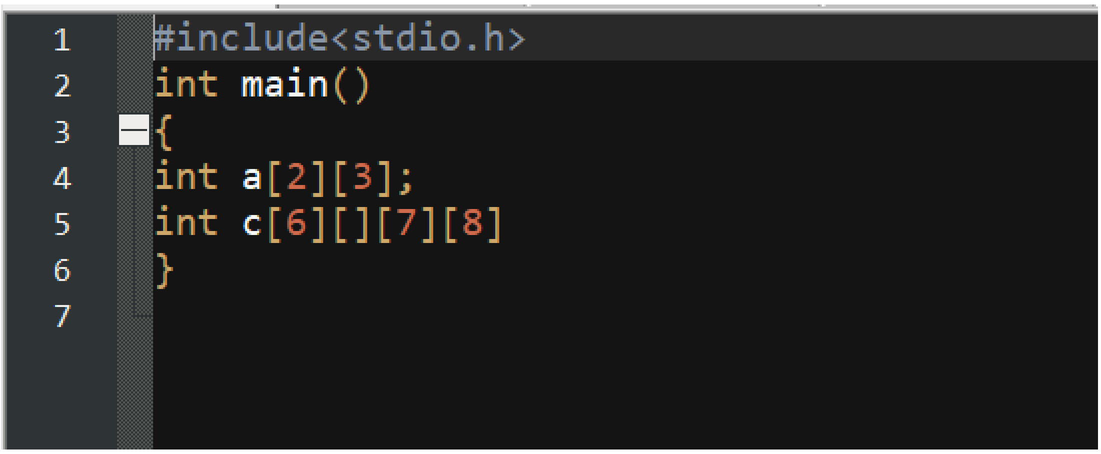
```
{id}\.h return HEADER;
"++" return INC;
"--" return DEC;
">=" return GREATEREQ;
"<=" return LESSEREQ;
"==" return EQCOMP;
"!=" return NOTEQ;
"&&" return ANDAND;
"||" return OROR;
\r ;
\t ;
[' '] ;
\n { ++yylineno; };
. return *yytext;
%%
int yywrap()
{
return(1);
}
```

array_valid.c



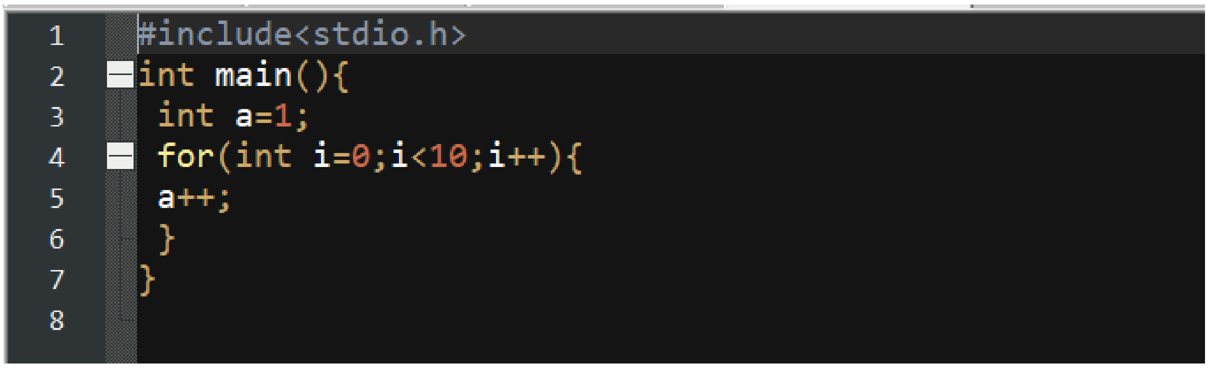
```
#include<stdio.h>
int main()
{
int a[2][3];
int b[2];
int c[6][6][7][8];
}
```

array_invalid.c



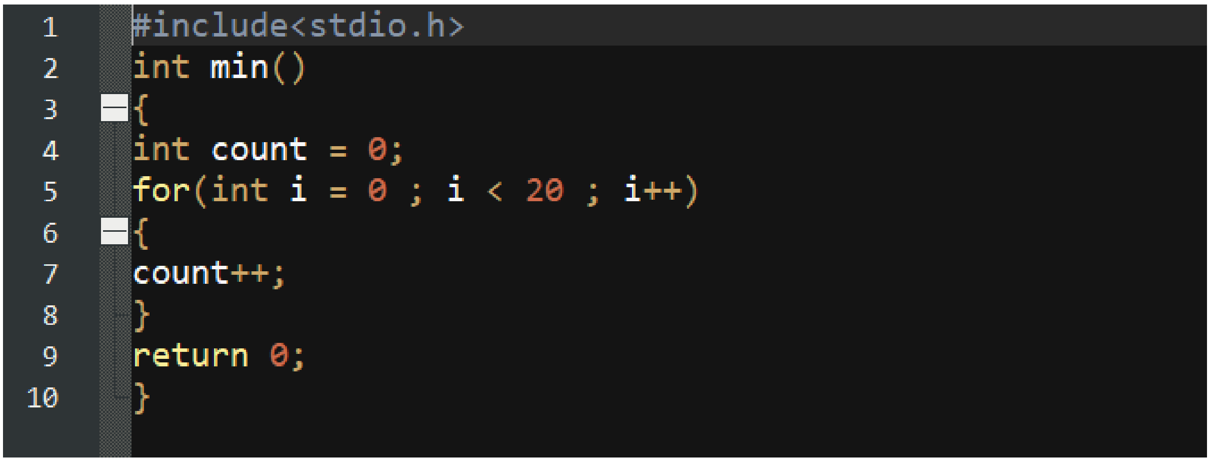
```
#include<stdio.h>
int main()
{
int a[2][3];
int c[6][][7][8]
}
```

forloop_valid.c



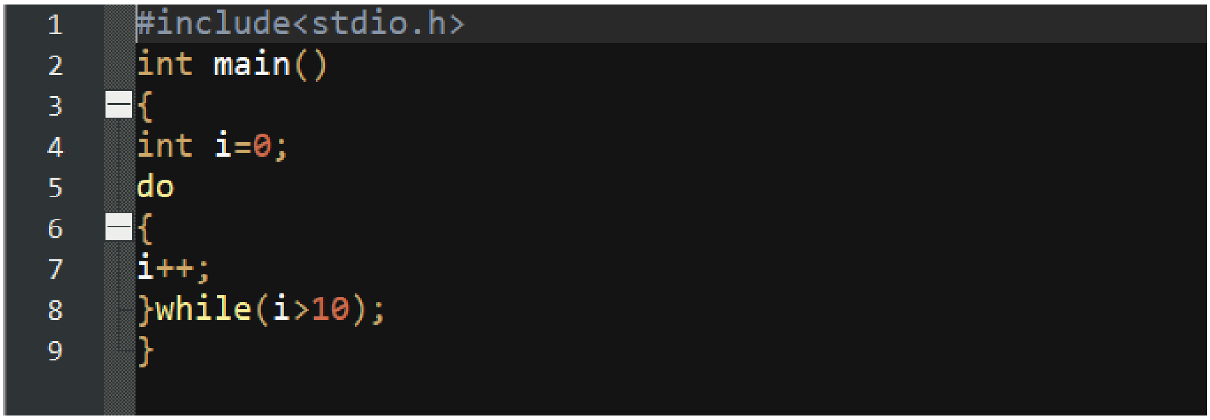
```c
#include<stdio.h>
int main(){
 int a=1;
 for(int i=0;i<10;i++){
 a++;
 }
}

```

forloop_invalid.c



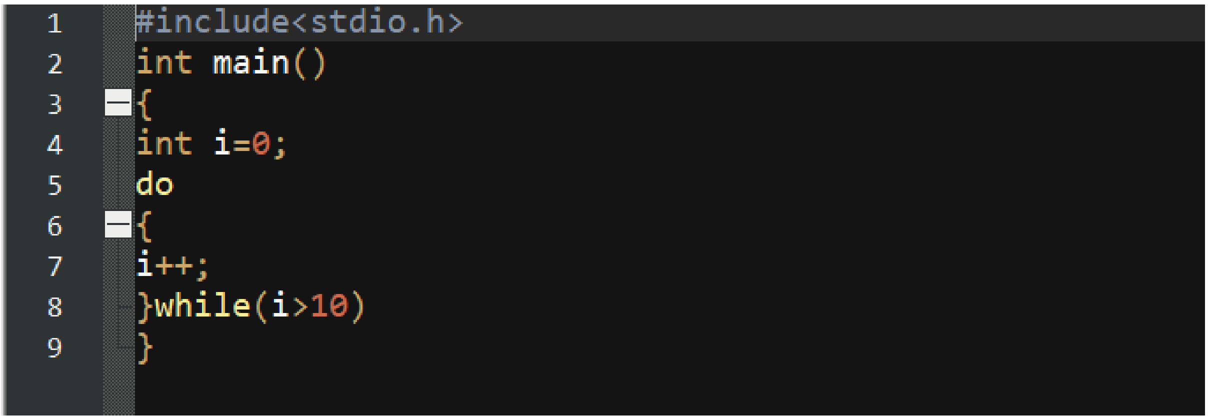
```c
#include<stdio.h>
int min()
{
int count = 0;
for(int i = 0 ; i < 20 ; i++)
{
count++;
}
return 0;
}
```

while_valid.c



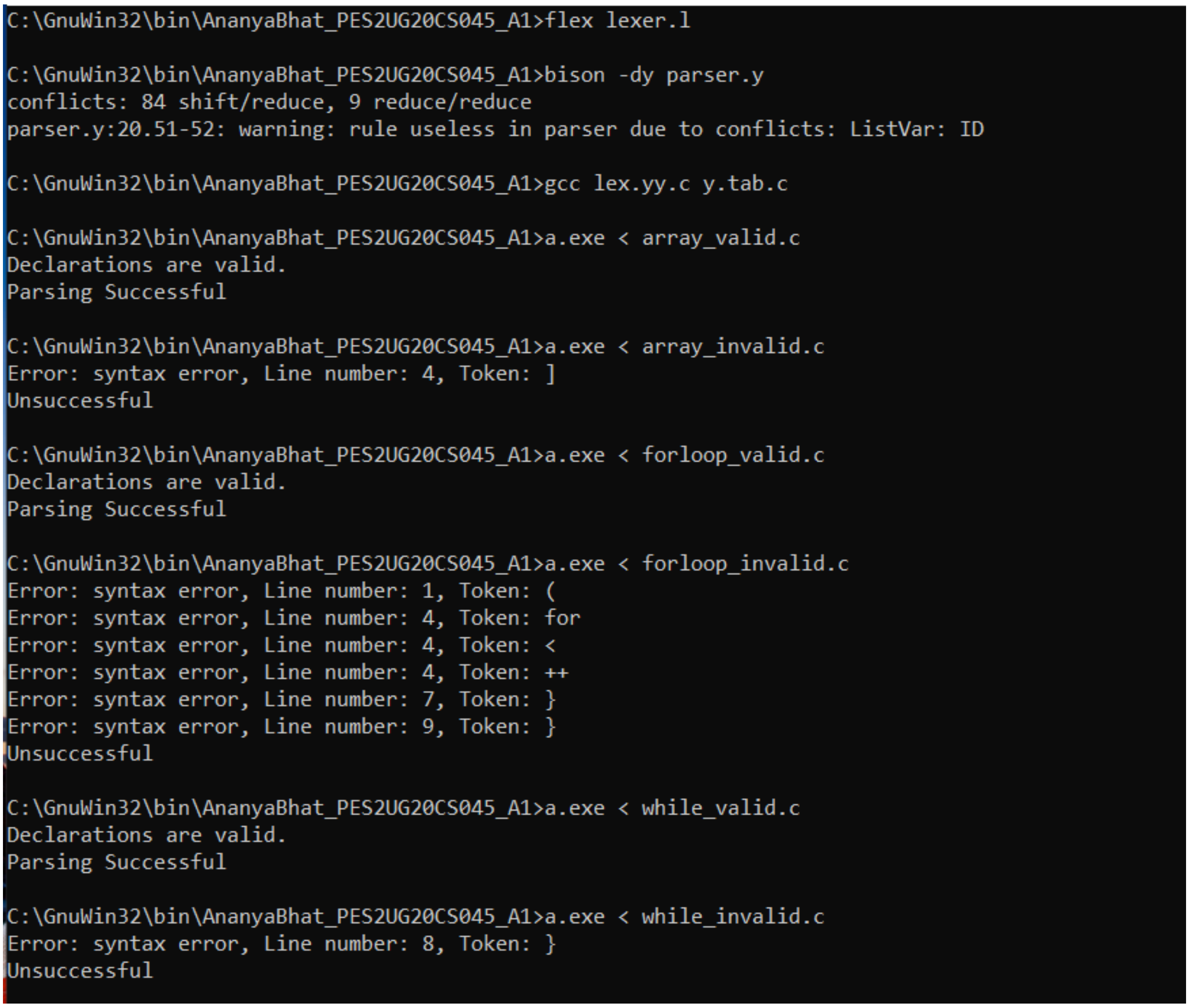
```c
#include<stdio.h>
int main()
{
int i=0;
do
{
i++;
}while(i>10);
}
```

while_invalid.c



```
#include<stdio.h>
int main()
{
int i=0;
do
{
i++;
}while(i>10)
}
```

Output:



```
C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>flex lexer.l

C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>bison -dy parser.y
conflicts: 84 shift/reduce, 9 reduce/reduce
parser.y:20.51-52: warning: rule useless in parser due to conflicts: ListVar: ID

C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>gcc lex.yy.c y.tab.c

C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>a.exe < array_valid.c
Declarations are valid.
Parsing Successful

C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>a.exe < array_invalid.c
Error: syntax error, Line number: 4, Token: ]
Unsuccessful

C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>a.exe < forloop_valid.c
Declarations are valid.
Parsing Successful

C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>a.exe < forloop_invalid.c
Error: syntax error, Line number: 1, Token: (
Error: syntax error, Line number: 4, Token: for
Error: syntax error, Line number: 4, Token: <
Error: syntax error, Line number: 4, Token: ++
Error: syntax error, Line number: 7, Token: }
Error: syntax error, Line number: 9, Token: }
Unsuccessful

C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>a.exe < while_valid.c
Declarations are valid.
Parsing Successful

C:\GnuWin32\bin\AnanyaBhat_PES2UG20CS045_A1>a.exe < while_invalid.c
Error: syntax error, Line number: 8, Token: }
Unsuccessful
```