UE20CS353-CD ASSIGNMENT-1

NAME	SRN	CLASS & SECTION
VIJAY J	PES2UG20CS815	6 - J

1.lexer.l code:

Open *lexer.l
~/PESU/6th Sem/CD/Assignment/PES2UG20CS815

```
%{
//Name: Vijay J
//SRN: PES2UG20CS815
//SECTION: J
#define YYSTYPE char*
#include "y.tab.h"
#include <stdio.h>
extern void yyerror(const char *);
%}

digit [0-9]
letter [a-zA-Z]
id {letter}({letter}|{digit})*
digits {digit}+
opFraction (\.{digits})?
opExponent ([Ee][+-]?{digits})?
number {digits}{opFraction}{opExponent}
%option yylineno
%%  // Separates definitions section from rules section

\/\/(.*)  ;
[\t\n]  ;
"int"     {return T_INT;}
"char"    {return T_CHAR;}
"double" {return T_DOUBLE;}
"float"   {return T_FLOAT;}
"while"   {return T_WHILE;}
"if"      {return T_IF;}
"else"    {return T_ELSE;}
"do"      {return T_DO;}
"#include"   {return T_INCLUDE;}
"main"   {return T_MAIN;}
"for" {return T_FOR;}
```

Open

*lexer.l
~/PESU/6th Sem/CD/Assignment/PES2UG20CS815

```
"--"    {return T_DECR;}
"≠"     {return T_NOTEQUAL;}
"≥"     {return T_GREATEREQ;}
"≤"     {return T_LESSEREQ;}
"("     {return *yytext;}
")"     {return *yytext;}
"["     {return *yytext;}
"]"     {return *yytext;}
"."     {return *yytext;}
","     {return *yytext;}
"{"     {return *yytext;}
"}"     {return *yytext;}
"*"     {return *yytext;}
"+"     {return *yytext;}
";"     {return *yytext;}
"-"     {return *yytext;}
"/"     {return *yytext;}
"="     {return *yytext;}
">"     {return *yytext;}
"<"     {return *yytext;}
"||"    {return T_OR;}
"&&"    {return T_AND;}
{number}        {
                        yylval=strdup(yytext);
                        return T_NUM;
                }
{id}\.h {return T_HEADER;}
{id}            {
                        yylval=strdup(yytext);
                        return T_ID;
                }
.               {}
%%
```

2.parser.y code:

Open — parser.y — ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 — Save

```
%{
//NAME: Vijay J
//SRN: PES2UG20CS815
//SECTION: J
        #include "sym_tab.c"
        #include <stdio.h>
        #include <stdlib.h>
        #include <string.h>
        #define YYSTYPE char*
        int type=-1;    //initial declaration of type for symbol table
        char* vval="~"; //initial declaration of value for symbol table
        int vtype=-1;   //initial declaration for type checking for symbol table
        int scope=0;    //initial declaration for scope
        int err=0;      //Initial declaration for error
        void yyerror(char* s); // error handling function
        int yylex(); // declare the function performing lexical analysis
        int arraysize=1; //initial size of array
        extern int yylineno; // track the line number
        extern char* yytext; // track the token

%}
/* declare tokens */
%token T_INT T_CHAR T_DOUBLE T_WHILE  T_INC T_DEC   T_OROR T_ANDAND T_EQCOMP T_NOTEQUAL T_GREATEREQ T_LESSEREQ T_LEFTSHIFT
T_RIGHTSHIFT T_PRINTLN T_STRING  T_FLOAT T_BOOLEAN T_IF T_ELSE T_STRLITERAL T_DO T_INCLUDE T_HEADER T_MAIN T_ID T_NUM T_FOR T_OR
T_AND T_INCR T_DECR

/* specify start symbol */
%start START


%%
START : PROG { if(err==0) printf("Valid syntax\n");
```

Open — parser.y — ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 — Save

```
PROG : T_INCLUDE '<' T_HEADER '>' PROG  /* include header */
        | MAIN PROG     /* main function */
        | DECLR ';' PROG    /* declarations */
        | ASSGN ';' PROG    /* assignments  */
        | DEC_ASGN';' PROG
        | FOR PROG
        | DO PROG
        | EXPR ';' PROG

        |       /* end of programs */
        ;

FOR : T_FOR '(' DEC_ASGN';'T_ID REL_OP EXPR ';' T_ID UNARY_OP ')' '{'{scope++;} STMT '}'{scope--;};
DEC_ASGN : {printf(" ");} TYPE LISTVAR;
/* Grammar for variable declaration */
DECLR : TYPE LISTVAR
        | TYPE ARRAY
        ;       /* always terminate with a ; */
ARRAY : T_ID BRACKET {
                if(check_sym_tab($1))   //if variable is in table then variable is being re-declared
                                {
                                        printf("Variable %s already declared\n",$1);
                                        yyerror($1);
                                }
                                else
                                {
                                insert_symbol($1,size(type)*arraysize,type,yylineno,scope);
                                arraysize=1;    //revert to default for checking
                                type=-1;
                                }
                }
        ;
```

Open | parser.y ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 | Save

```
67 BRACKET : '[' T_NUM ']' BRACKET { arraysize*=atoi($2);}
68         |
69         ;
70 LISTVAR : LISTVAR ',' VAR
71           | VAR
72           ;
73
74 VAR: T_ID '=' EXPR      {
75                                 if(check_sym_tab($1))   //if variable is in table then variable is being re-declared
76                                 {
77                                         printf("Variable %s already declared\n",$1);
78                                         yyerror($1);
79                                 }
80                                 else
81                                 {
82                                 insert_symbol($1,size(type),type,yylineno,scope);
83                                 insert_val($1,vval,yylineno);
84                                 vval="~";       //revert to default for checking
85                                 type=-1;
86                                 }
87                         }
88      | T_ID             {
89                                 if(check_sym_tab($1))   //if variable is in table then variable is being re-declared
90                                 {
91                                         printf("Variable %s already declared\n",$1);
92                                         yyerror($1);
93                                 }
94                                 else{
95                                 insert_symbol($1,size(type),type,yylineno,scope);
96                                 type=-1;        //revert to default for checking
97                                 }
98                         }
```

Open | parser.y ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 | Save

```
100 TYPE : T_INT {type = INT;}              //INT=2
101        | T_FLOAT {type = FLOAT;}        //FLOAT=3
102        | T_DOUBLE {type = DOUBLE;}      //DOUBLE=4
103        | T_CHAR {type = CHAR;}          //CHAR=1
104        ;
105
106 /* Grammar for assignment */
107 ASSGN : T_ID {type=retrieve_type($1);}'=' EXPR  {
108                                 if(!check_sym_tab($1))  //if variable not declared then value cannot be assigned
109                                 {
110                                         printf("Variable %s not declared\n",$1);
111                                         yyerror($1);
112                                 }
113                                 insert_val($1,vval,yylineno);
114                                 vval="~";               //to make sure previous values aren't inserted into other identifiers
115                                 type=-1;                //revert to default for checking
116                         }
117        ;
118
119 EXPR : EXPR REL_OP E
120        | E {vval=$1;}
121        | EXPR LOGICAL_OP E
122         | E UNARY_OP
123         | UN_OP E
124        ;
125
126 /* Expression Grammar */
127 E : E '+' T     {
128                         if(vtype==2)                            //integer
129                                 sprintf($$,"%d",(atoi($1)+atoi($3)));
130                         else if(vtype==3)                       //float or double
131                                 sprintf($$,"%lf",(atof($1)+atof($3)));
```

Open | parser.y ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 | Save

```
                        else
                        {
                                printf("Character used in arithmetic\n");
                                yyerror($$);
                                $$="~";
                        }
                }
    | E '-' T   {
                        if(vtype==2)                            //integer
                                sprintf($$,"%d",(atoi($1)-atoi($3)));
                        else if(vtype==3)                       //float or double
                                sprintf($$,"%lf",(atof($1)-atof($3)));
                        else
                        {
                                printf("Character used in arithmetic\n");
                                yyerror($$);
                                $$="~";
                        }
                }
    | T {$$=$1;}
    ;


T : T '*' F     {
                        if(vtype==2)                            //integer
                                sprintf($$,"%d",(atoi($1)*atoi($3)));
                        else if(vtype==3)                       //float or double
                                sprintf($$,"%lf",(atof($1)*atof($3)));
                        else
                        {
                                printf("Character used in arithmetic\n");
                                yyerror($$);
```

Open | parser.y ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 | Save

```
                        }
                }
    | T '/' F   {
                        if(vtype==2)                            //integer
                                sprintf($$,"%d",(atoi($1)/atoi($3)));
                        else if(vtype==3)                       //float or double
                                sprintf($$,"%lf",(atof($1)/atof($3)));
                        else
                        {
                                printf("Character used in arithmetic\n");
                                yyerror($$);
                                $$="~";
                        }
                }
    | F {$$=$1;}
    ;

F : '(' EXPR ')'
    | T_ID      {
                        if(check_sym_tab($1))           //check if variable is in symbol table
                        {
                                char* check=retrieve_val($1);
                                if(check=="~")          //if variable has no value then can't be used for assignment
                                {
                                        printf("Variable %s not initialised",$1);
                                        yyerror($1);
                                }
                                else
                                {
                                        $$=strdup(check);
                                        vtype=type_check(check);
                                        if(vtype!=type && type!=-1)     //checks for matching type
```

parser.y
~/PESU/6th Sem/CD/Assignment/PES2UG20CS815

```
                                        {
                                                printf("Mismatch type\n");
                                                yyerror($1);
                                        }
                                }
                        }
                        else
                        {
                                        printf("Variable %s not declared\n",$1);
                                        yyerror($1);
                        }
                }
    | T_NUM     {
                        $$=strdup($1);
                        vtype=type_check($1);
                        if(vtype!=type && type!=-1)     //checks for matching type
                        {
                                printf("Mismatch type\n");
                                yyerror($1);
                        }
                }
    | T_STRLITERAL {
                        $$=strdup($1);
                        vtype=1;
                        if(vtype!=type)         //checks for matching type
                        {
                                printf("Mismatch type\n");
                                yyerror($1);
                        }
                }
    ;
```

parser.y
~/PESU/6th Sem/CD/Assignment/PES2UG20CS815

```
REL_OP :   T_LESSEREQ
           | T_GREATEREQ
           | '<'
           | '>'
           | T_EQCOMP
           | T_NOTEQUAL
           ;

LOGICAL_OP:T_AND
        |T_OR
        |'!'
        ;

UNARY_OP: T_INCR
        |T_DECR
        ;

UN_OP   : T_INCR
        |T_DECR
        |'-'
        |'+'
        |'!'
        ;
/* Grammar for main function */
MAIN : TYPE T_MAIN '(' EMPTY_LISTVAR ')' '{' {scope++;} STMT '}' {scope--;};


/* argument list can be empty, or have a list of variables */
EMPTY_LISTVAR : LISTVAR
                |       /* similar to lambda */
                ;
```

Open parser.y ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 Save

```
STMT : STMT_NO_BLOCK STMT
        | BLOCK {scope++;} STMT {scope--;}
        | FOR PROG
        | DO
        | EXPR ';'
        | DEC_ASGN ';'
        |
        ;

%nonassoc T_IFX;
%nonassoc T_ELSE;

STMT_NO_BLOCK : DECLR ';'
        | ASSGN ';'
        | T_IF COND {scope++;} STMT {scope--;} %prec T_IFX   /* if loop */
        | T_IF COND {scope++;} STMT {scope--;} T_ELSE STMT   /* if else loop */
        | WHILE
        ;

DO    : T_DO BLOCK WHILE
      ;
BLOCK : '{' {scope++;} STMT '}' {scope--;};


/* Condition can be an expression or an assignment */
COND : EXPR
       | ASSGN
       ;

/* Grammar for while loop */
WHILE : T_WHILE '(' COND ')' WHILE_2;
```

Open parser.y ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 Save

```
/* Condition can be an expression or an assignment */
COND : EXPR
        | ASSGN
        ;

// while loop may or may not have block of statements
WHILE_2 : '{'{scope++;} STMT '}'{scope--;}
        | ';'
        ;

%%


/* error handling function */
void yyerror(char* s)
{
        err+=1;
        printf("Error :syntax error,line number:%d,token:%s \n",yylineno,yytext);
}
int yywrap()
{
    return(1);
}

/* main function - calls the yyparse() function which will in turn drive yylex() as well */
int main(int argc, char* argv[])
{
        t=init_table();
        //printf("here \n");
        yyparse();
        display_sym_tab();
        return 0;

}
```

3.run.sh code:

Open run.sh ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815

```
#!/bin/bash

lex lexer.l
yacc -d parser.y -Wno
gcc -g y.tab.c lex.yy.c

```

4.Input codes And Output :

Forloop_valid.c

forloop_valid.c

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

Output

```
~/P/6/C/A/PES2UG20CS815

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 67ms
λ ./run.sh

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 1s
λ ./a.out<forloop_valid.c
 Valid syntax
Name    size    type    lineno  scope   value
a       2       2       3       1       1
i       2       2       4       1       0

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 2ms
λ _
```

Forloop_invalid.c

forloop_invalid.c

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

Save the current file

Output

```
~/P/6/C/A/PES2UG20CS815

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 60ms
λ ./run.sh

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 537ms
λ ./a.out<forloop_invalid.c
 Error :syntax error,line number:5,token:i
Name    size    type    lineno  scope   value
count   2       2       4       1       0
i       2       2       5       1       0

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 2ms
λ _
```

While_valid.c

Open | while_valid.c — ~/PESU/6th Sem/CD/Assignment/PES2UG20... | Save

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

Output

```
~/P/6/C/A/PES2UG20CS815

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 2ms
λ ./run.sh

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 501ms
λ ./a.out<while_valid.c
Valid syntax
Name    size    type    lineno  scope   value
i       2       2       4       1       0

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 2ms
λ _
```

While_invalid.c

Open ▾ | while_invalid.c — ~/PESU/6th Sem/CD/Assignment/PES2UG20... | Save

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

Output

~/P/6/C/A/PES2UG20CS815

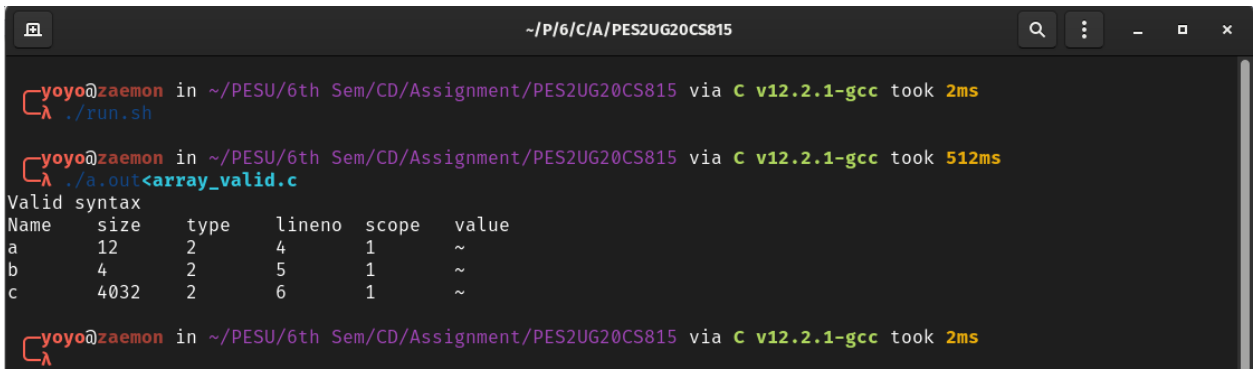
```
yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 2ms
λ ./run.sh

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 516ms
λ ./a.out<while_invalid.c
Error :syntax error,line number:9,token:}
Name    size    type    lineno  scope   value
i       2       2       4       1       0

yoyo@zaemon in ~/PESU/6th Sem/CD/Assignment/PES2UG20CS815 via C v12.2.1-gcc took 1ms
λ _
```

Array_valid.c

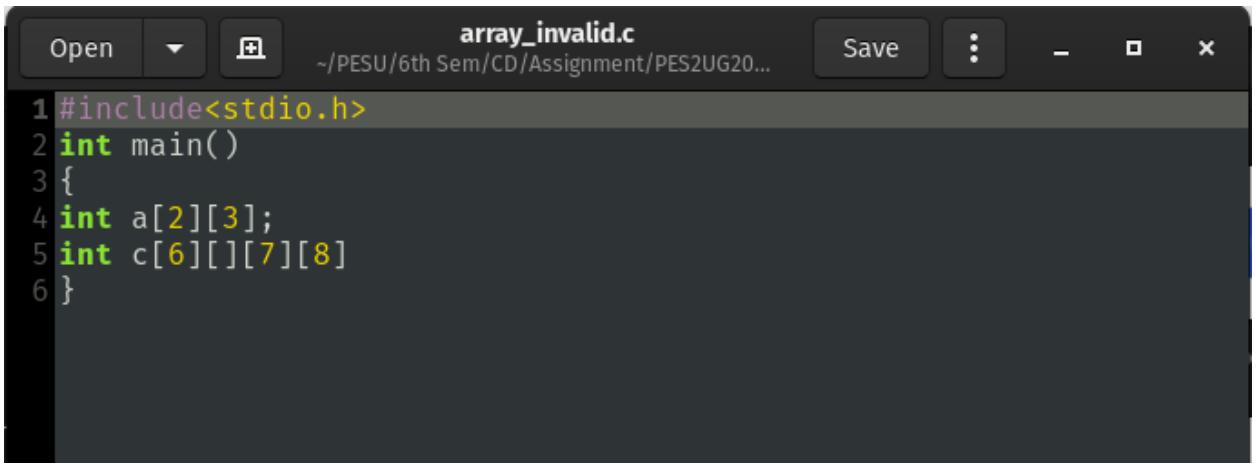
Open ▾ | array_valid.c — ~/PESU/6th Sem/CD/Assignment/PES2UG20... | Save

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

Output



Array_invalid.c



Output