**Name:** Paryusha Shah

**Roll No:** 22bce234

**Course:** Principles of Compiler Design

**Course Code:** 4CS501CC25

***Practical 6: Write semantics for declaration statements to update the data type of symbols in the symbol table.***

**File:** prog6.l

%{

#include "y.tab.h"

#include <stdio.h>

%}

%%

"int"       { return INT; }

"float"     { return FLOAT; }

"char"      { return CHAR; }

[a-zA-Z\_][a-zA-Z0-9\_]\*  { yylval.str = strdup(yytext); return ID; }

[ \t\n]+    ;

","         { return COMMA; }

";"         { return SEMI; }

.           ;

%%

int yywrap() { return 1; }

**File:** prog6.y

%{

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int yylex(void);

extern FILE \*yyin;

struct Symbol {

    char name[30];

    char datatype[15];

    struct Symbol \*next;

};

struct Symbol \*head = NULL;

void insertSymbol(char \*name, char \*datatype) {

    struct Symbol \*temp = head;

    while (temp != NULL) {

        if (strcmp(temp->name, name) == 0) {

            printf("Warning: duplicate identifier %s\n", name);

            return;

        }

        temp = temp->next;

    }

    struct Symbol \*newNode = (struct Symbol\*)malloc(sizeof(struct Symbol));

    strcpy(newNode->name, name);

    strcpy(newNode->datatype, datatype);

    newNode->next = NULL;

    if (head == NULL) {

        head = newNode;

    } else {

        struct Symbol \*t = head;

        while (t->next != NULL) t = t->next;

        t->next = newNode;

    }

    printf("Inserted: %-10s | Type: %s\n", name, datatype);

}

void printTable() {

    struct Symbol \*temp = head;

    printf("\n=== SYMBOL TABLE ===\n");

    printf("%-15s %-15s\n", "Identifier", "Datatype");

    printf("--------------------------------\n");

    while (temp != NULL) {

        printf("%-15s %-15s\n", temp->name, temp->datatype);

        temp = temp->next;

    }

}

void yyerror(char \*s);

%}

%union {

    char\* str;

}

%token <str> ID

%token INT FLOAT CHAR

%token COMMA SEMI

%type <str> decl\_list decl T

%%

program:

    program decl\_list SEMI

    | decl\_list SEMI

    ;

decl\_list:

    decl { $$ = $1; }

    | decl\_list COMMA ID {

        insertSymbol($3, $1);

        $$ = $1;

    }

    ;

decl:

    T ID {

        insertSymbol($2, $1);

        $$ = $1;

    }

    ;

T:

    INT    { $$ = "int"; }

    | FLOAT { $$ = "float"; }

    | CHAR  { $$ = "char"; }

    ;

%%

void yyerror(char \*s) {

    fprintf(stderr, "Error: %s\n", s);

}

int main() {

    FILE \*fp = fopen("input.c", "r");

    if (!fp) {

        perror("Cannot open file input.c");

        return 1;

    }

    yyin = fp;

    yyparse();

    fclose(fp);

    printTable();

    return 0;

}

**File:** input.c

int a, b;

float c, d, e;

char ch;

**Commands for run program:**

