**Compiler Design Lab**

**Assignment-7**

**Generation of Intermediate Code using Lex and Yacc**

**Name : Vikraman S**

**Reg No. : 185001195**

**Code:**

**intermediate.l**

**%{**

#include<stdio.h>

#include "y.tab.h"

extern YYSTYPE yylval**;**

**%}**

id **[**a**-**zA**-**Z\_**]+**

no **[**0**-**9**]+**

ro **(**"<"**|**"<="**|**">"**|**">="**|**"=="**|**"!="**)**

nl "\n"

sp "\t"**|**" "

**%%**

"begin" **{return** BEG**;}**

"end" **{return** END**;}**

"if" **{return** IF**;}**

"then" **{return** THEN**;}**

"else" **{return** ELSE**;}**

"endif" **{return** ENDIF**;}**

**{**id**}** **{**yylval**.**var**=**strdup**(**yytext**);return** ID**;}**

**{**no**}** **{**yylval**.**val**=**atoi**(**yytext**);return** NUM**;}**

**{**ro**}** **{return** RELOP**;}**

**{**nl**}** **{return** NL**;}**

**{**sp**}** **;**

**.** **return** yytext**[**0**];**

**%%**

int yywrap**(){return** 1**;}**

**intermediate.y**

**%{**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int yyerror**(**char **\***err**);**

int yylex**(**void**);**

int c**=**0**;**

char res**[**100**],**temp**[**50**];**

**%}**

**%**token ID NUM RELOP BEG END IF THEN ELSE ENDIF NL

**%**union**{**

int val**;**

char**\*** var**;**

**}**

**%**type**<**val**>** NUM E F T

**%**type**<**var**>** ID

**%**left '<' '>'

**%**right '\*' '/'

**%**left '+' '-'

**%**left '(' ')'

**%%**

G**:** BEG NL S NL END **{**sprintf**(**temp**,**"\n\nSyntactically correct\n"**);**strcat**(**res**,**temp**);**printf**(**"%s"**,**res**);return** 1**;}**

S**:** IF C THEN NL A NL ELSE NL A NL ENDIF

C**:** X RELOP X

X**:** ID **|** NUM

A**:** ID '=' E ';' **{**sprintf**(**temp**,**"\n%s = t%d"**,**$1**,**$3**);**strcat**(**res**,**temp**);}**

E**:** E '\*' T **{**sprintf**(**temp**,**"\nt%d = t%d %s t%d"**,++**c**,**$1**,**"\*"**,**$3**);**strcat**(**res**,**temp**);**$$**=**c**;}**

**|** E '/' T **{**sprintf**(**temp**,**"\nt%d = t%d %s t%d"**,++**c**,**$1**,**"/"**,**$3**);**strcat**(**res**,**temp**);**$$**=**c**;}**

**|** E '%' T **{**sprintf**(**temp**,**"\nt%d = t%d %s t%d"**,++**c**,**$1**,**"%"**,**$3**);**strcat**(**res**,**temp**);**$$**=**c**;}**

**|** T **{**$$**=**$1**;}**

T**:** T '+' F **{**sprintf**(**temp**,**"\nt%d = t%d %s t%d"**,++**c**,**$1**,**"+"**,**$3**);**strcat**(**res**,**temp**);**$$**=**c**;}**

**|** T '-' F **{**sprintf**(**temp**,**"\nt%d = t%d %s t%d"**,++**c**,**$1**,**"-"**,**$3**);**strcat**(**res**,**temp**);**$$**=**c**;}**

**|** F **{**$$**=**$1**;}**

F**:** ID **{**sprintf**(**temp**,**"\nt%d = %s"**,++**c**,**$1**);**strcat**(**res**,**temp**);**$$**=**c**;}**

**|** NUM **{**sprintf**(**temp**,**"\nt%d = %d"**,++**c**,**$1**);**strcat**(**res**,**temp**);**$$**=**c**;}**

**%%**

void main**()**

**{**

yyparse**();**

printf**(**"\n"**);**

**}**

int yyerror**(**char **\***err**)**

**{**

printf**(**" Invalid - %s\n"**,**err**);**

exit**(**0**);**

**}**

**Screenshot:**

