Compiler Design Practical-5

**PRACTICAL-5**

**Aim : Write a program for validation of arithmetic statement.**

**Code:**

%{

#include<stdio.h>

#include<string.h>

int noprt=0, nopnd=0, valid=1, top=-1, m, l=0,i=-1, j=0;

char opnd[10][10], oprt[10][10], a[100];

%}

%%

"(" { top++; a[top]='(' ; }

"{" { top++; a[top]='{' ; }

"[" { top++; a[top]='[' ; }

")" { if(a[top]!='(')

{

valid=0;

return 0;

}

else

top--;

}

"}" { if(a[top]!='{')

{

valid=0;

return 0;

}

else

top--;

}

"]" { if(a[top]!='[')

{

valid=0;

return 0;

}

17012011056-Dharmay Sureja IoT1-CE

Compiler Design Practical-5

else

top--;

}

"+"|"-"|"\*"|"/"|"%" { noprt++;

strcpy(oprt[l], yytext);

l++;

}

[0-9]+|[a-zA-Z][a-zA-Z0-9\_]\* {nopnd++; strcpy(opnd[j],yytext);j++;} %%

int yywrap()

{

return 1;

}

int main()

{

int k;

printf("Enter the expression.. at end press ^d\n");

yylex();

if(valid==1 && i==-1 && (nopnd-noprt)==1)

{

printf("The expression is valid\n");

printf("The operators are\n");

for(k=0;k<l;k++){

printf("%s\n",oprt[k]);

}

}

else

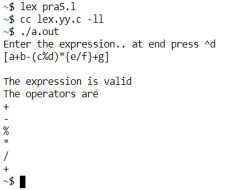
printf("The expression is invalid");

}

17012011056-Dharmay Sureja IoT1-CE

Compiler Design Practical-5

**Output**

****

17012011056-Dharmay Sureja IoT1-CE