14: Intermediate code generator. P.vishnu 28 192011171

Aim:- To Implementation a simple intermediate code generation in C.

Program:-

```
#include <stdio.h>
#include <conio.h>
struct three
 {
 char dar[10],temp(3)
 }s(30);
FILE f1,f2
f1=fopen ("sum"+@+: r");
f2=fopen ("out", +m); w);
 int 0(y,adi7)
fprint (f2,"ie")
strcpy (d2,"f")
 j++;
 }
 close(f1);
 close(f2);
```

output:-

out=int1 + in2+in3-in4

out of t1=in1+in2-t2=t1+in3

t3=t2-in4 out=t3

Result:- simple intermediate is executed successfully.

P. Vishnu
192011179.

15. Back End of the compiler:-

Aim:- To Implement c program to Implement the back end to the compiler.

Program:-

```
#include<stdio.h>.
#include<conio.h>
struct three.
char d.t[1],d2[1]="t",int i=0; j=1, len=0
f1,f2=1f1,*f2;
f1 = fopen.("sum=text", "r").
f2 = fopen (f1, "%s", stlen,data != EOF)
i toa(j,d1 7)
strcat (d2,d1);
strcpy (s,(j], tempid2)];

}
close(f1);
fclose(f2);
```

output:-
out = initinztins -in4.

Result:-
c program to Implement the back end to compiler is Executed.

16. LEX PROGRAM for CAPITAL words

```
%{
#include<stdio.h>
%}
%%
[A-Z]+[a-z]+[A-Z]+ {ECHO;}
.;
%%
int yywrap(){}
int main()
{
printf("Enter the input string:\n");
yylex();
}
```

output:- Input.
IWDR.

Result:- lex program for capital words is executed.

EXP:- 17. lex - count:

Aim:- lex program for count comment lines.

program:-

```
%{
#include<stdio.h>
int nc=0
%}
```

31 Pushmu.
192011179.

```
%.%
"/"*[a-zA-Z0-9\n/t]*"*/"{ncp++}
"//"[a-zA-Z0-9/t]*\n "{nc++};
%.%
int yywrap(){}
int main(int argc,char*argv[])
{.
yyin =fopen(argv[1],"r");
yyout =fopen("output.c","w");
yylex();
printf("the no.of comment lines =%d\n",nc);
}.
```

output :- abc.

Result :- lex program for words if Executed successfully.

EXP :- 18. Email

Aim :- lex program for Email valid or not.

program :-

```
%.{
%.{
%.f
[a.zo-o-a]+@[a-z]+".com];in"{printf("%s)
Invalid.
```

```
%%
int yywrap() {}
int main()
{
printf("Enter the mail:");
yylex();
}
```

EX: 19 mobile no. valid not.

Aim: lex program for mobile number valid or not.

program:

```
%{
%{
%}
[0-9][0-9]{9}{printf("\n mobile number valid\n");
+printf("\n mobile number invalid or not");
yylex();
printf("\n");
return 0;
}
```

o/p: ! valid.

Result: lex program for mm is valid or not.

Executed.

EXP:- lex:- Pos & Neg :- 33 P.Vishnu, 192011179.

20. Aim :- lex program for positive & Negative number.

Program :-

```
%{
#include <stdio.h>
%}
%%
^[-][0-9]+{negative_no++;
printf("negative number = %s\n", yytext);}
[0-9]+{positive_no++;
printf("positive number = %s\n", yytext);}
%%
int yywrap(){}
int main()
{
yylex();
printf("no.of positive number = %d",
"no.of negative number = %d \n",
positive_no, negative_no);
return 0;
}
```

O/P. 2 -1 4 -1 8 → P.2 m:- 1.

Result :- lex program for pos & neg is Executed successfully.