14. Intermediate Code Generator

AIM:- To implement a Simple intermediate code generator in C.

Program

```
#include<stdio.h>
#include<conio.h>
struct three
{
char dat[10], temp[10];
}st[30];
FILE *f1,*f2;
f1=fopen("sum","r");
f2=fopen("out","w");
int a(),ad1,z;
fprintf(f2,"lc")
strcpy(d2,"t")
i++;
}
close(f1);
fclose(f2);
```

Output

Out=int1+in2+in3-in4

Out-int1=in1+in2-t2=t1+in3

t3=t2-in4Out-t3.

Result:- Simple intermediate code is executed successfully

3/8/21

15 BACK END OF THE COMPILER

PVersion 501

AIM:- To implement C program to implement the back end of the complier

PROGRAM

```
#include stdio.h>
#include <conio.h>
struct three
{
char d1[7], d2[7] = "t", int i=0, j=1, len=0;
 FILE *f1, *f2;
f1 = fopen ("sum.txt", "r")
f2 = fopen (f1, "%s", strlen), data != EOF)
itoa (j, d1, 7)
strcat (d2, d1);
strcpy (s[i], temp, d2)];
}
close (f1);
fclose (f2);
```

output:-

```
out = in1 + in2 * in3 - in4
```

Result

C program to implement the back end of compiler is Executed.

16. LEX PROGRAM FOR CAPITAL WORDS:

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

Output INDRA
INDRA

RESULT:- Lex program for Capital words is Executed.

AIM:- Lex Program for Count comment lines.

Program

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

P Vishnu sai

```
%}
"/*"[a-zA-Z0-9\n\t]*"*/" {n++}
"//"[a-zA-Z0-9\t]*\n" {n++};
%%
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",n);
}
```

Output: abc

Result: lex program for comment words is executed successfully.

18. Aim:- Lex program for Email valid or not

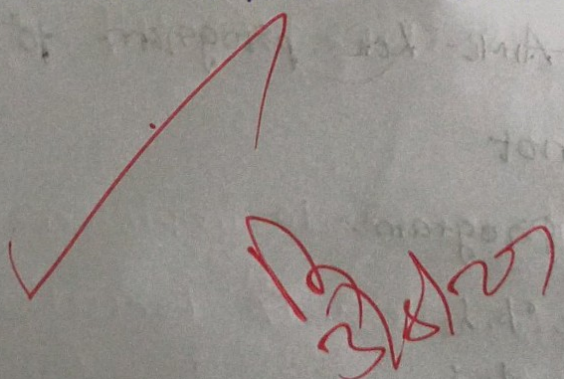
Program:-

```
%{
%}
%%
[a-z0-9]+@[a-z]+".com"|".in" {printf("%s valid");
```

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

19. 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

Rex1:- lex program for m.n is valid or not

Executed.

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

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 numbers=%d",
      "no of negative numbers=%d\n",
            positive_no, negative_no);
return 0;
}
```

O/P:- -2, 4, 8 → P. 2 N:-1

Result:- lex program for pos & neg is executed Successfully