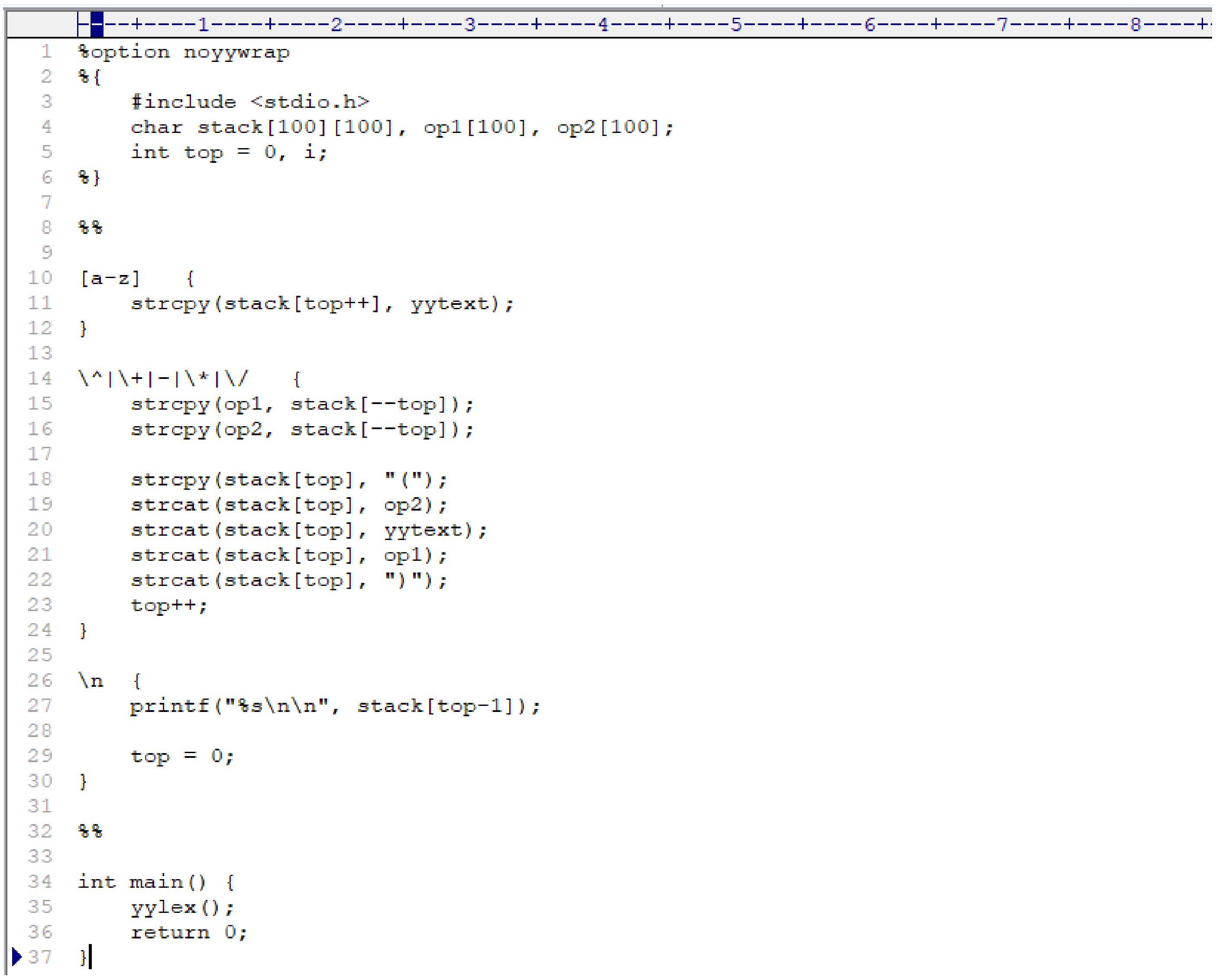
26/10/2021 **CD SPOT 7** **2019103573**

SPOT – EXERCISE

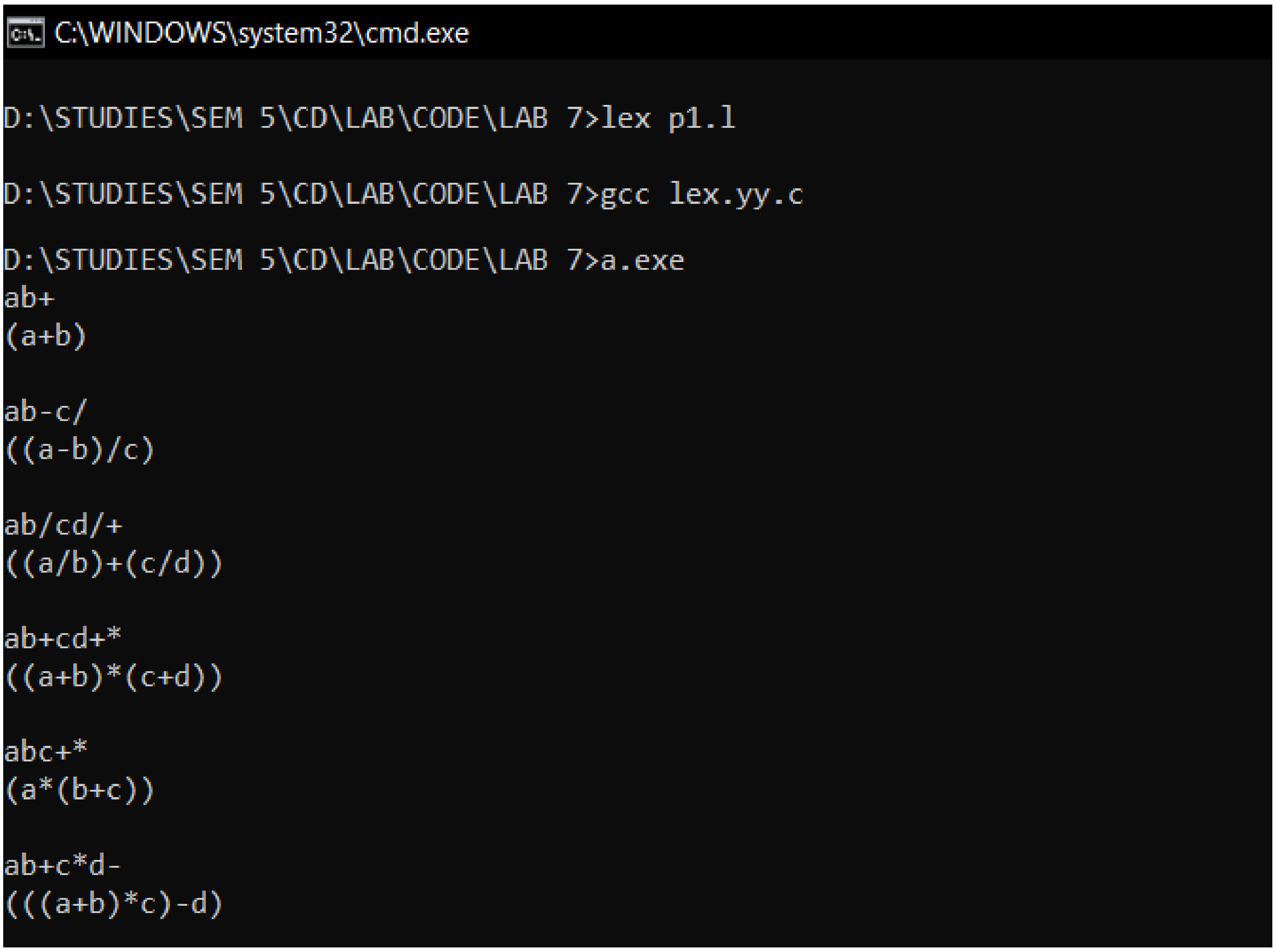
1.To write a program to convert the given postfix expression to infix expression.

CODE :-



```
%option noyywrap
%{
    #include <stdio.h>
    char stack[100][100], op1[100], op2[100];
    int top = 0, i;
%}

%%

[a-z]   {
    strcpy(stack[top++], yytext);
}

\^|\+|-|\*|\/    {
    strcpy(op1, stack[--top]);
    strcpy(op2, stack[--top]);

    strcpy(stack[top], "(");
    strcat(stack[top], op2);
    strcat(stack[top], yytext);
    strcat(stack[top], op1);
    strcat(stack[top], ")");
    top++;
}

\n  {
    printf("%s\n\n", stack[top-1]);

    top = 0;
}

%%

int main() {
    yylex();
    return 0;
}
```

OUTPUT :-

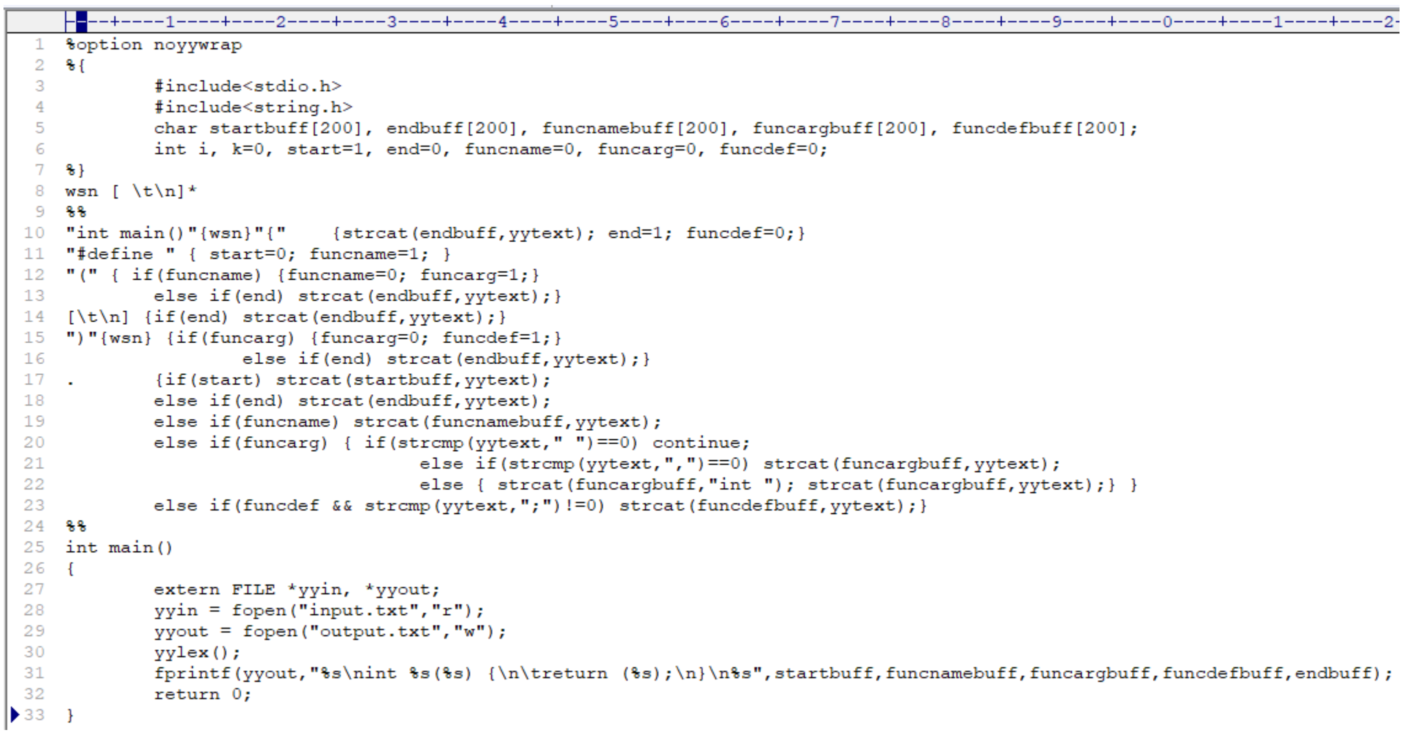


C:\WINDOWS\system32\cmd.exe

```
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>lex p1.l

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>gcc lex.yy.c

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>a.exe
ab+
(a+b)

ab-c/
((a-b)/c)

ab/cd/+
((a/b)+(c/d))

ab+cd+*
((a+b)*(c+d))

abc+*
(a*(b+c))

ab+c*d-
(((a+b)*c)-d)
```

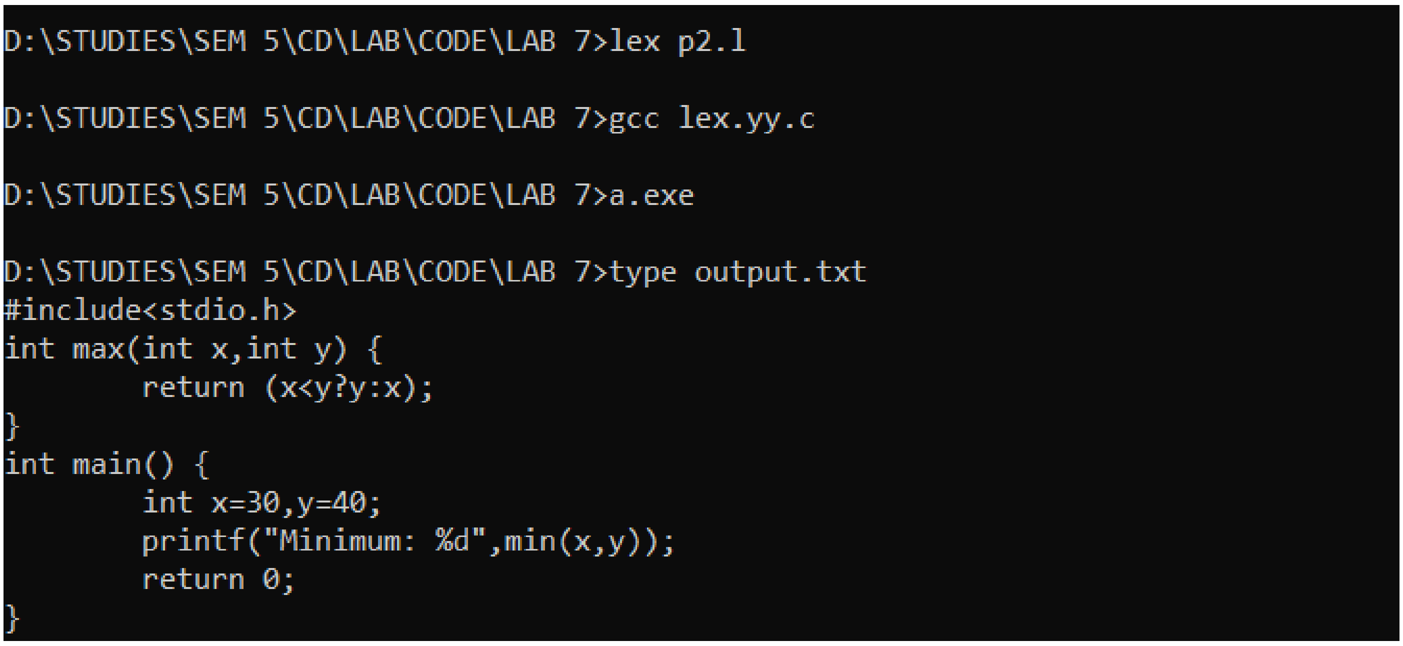
2.Write a program to convert the given macro to function.

CODE :-



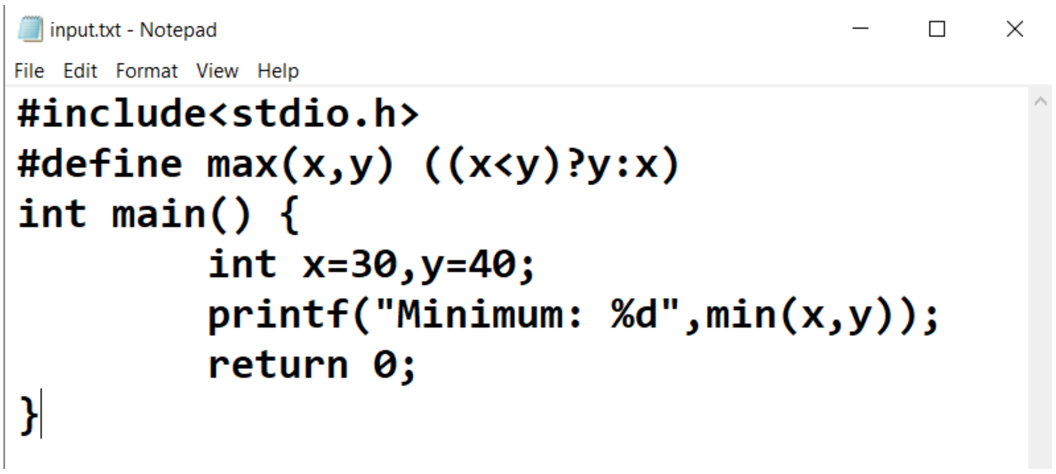
```
%option noyywrap
%{
        #include<stdio.h>
        #include<string.h>
        char startbuff[200], endbuff[200], funcnamebuff[200], funcargbuff[200], funcdefbuff[200];
        int i, k=0, start=1, end=0, funcname=0, funcarg=0, funcdef=0;
%}
wsn [ \t\n]*
%%
"int main()"{wsn}"{"    {strcat(endbuff,yytext); end=1; funcdef=0;}
"#define " { start=0; funcname=1; }
"(" { if(funcname) {funcname=0; funcarg=1;}
        else if(end) strcat(endbuff,yytext);}
[\t\n] {if(end) strcat(endbuff,yytext);}
")"{wsn} {if(funcarg) {funcarg=0; funcdef=1;}
                else if(end) strcat(endbuff,yytext);}
.       {if(start) strcat(startbuff,yytext);
        else if(end) strcat(endbuff,yytext);
        else if(funcname) strcat(funcnamebuff,yytext);
        else if(funcarg) { if(strcmp(yytext," ")==0) continue;
                                else if(strcmp(yytext,",")==0) strcat(funcargbuff,yytext);
                                else { strcat(funcargbuff,"int "); strcat(funcargbuff,yytext);} }
        else if(funcdef && strcmp(yytext,";")!=0) strcat(funcdefbuff,yytext);}
%%
int main()
{
        extern FILE *yyin, *yyout;
        yyin = fopen("input.txt","r");
        yyout = fopen("output.txt","w");
        yylex();
        fprintf(yyout,"%s\nint %s(%s) {\n\treturn (%s);\n}\n%s",startbuff,funcnamebuff,funcargbuff,funcdefbuff,endbuff);
        return 0;
}
```

OUTPUT :-



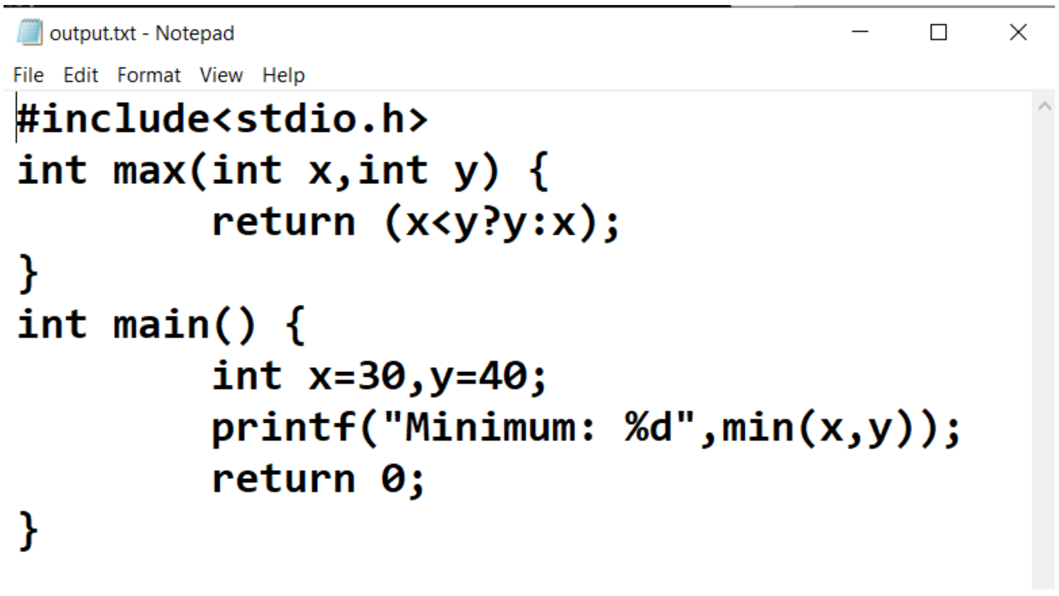
```
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>lex p2.l

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>gcc lex.yy.c

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>a.exe

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>type output.txt
#include<stdio.h>
int max(int x,int y) {
        return (x<y?y:x);
}
int main() {
        int x=30,y=40;
        printf("Minimum: %d",min(x,y));
        return 0;
}
```

Input.txt



```
#include<stdio.h>
#define max(x,y) ((x<y)?y:x)
int main() {
        int x=30,y=40;
        printf("Minimum: %d",min(x,y));
        return 0;
}
```

Output.txt



```
#include<stdio.h>
int max(int x,int y) {
        return (x<y?y:x);
}
int main() {
        int x=30,y=40;
        printf("Minimum: %d",min(x,y));
        return 0;
}
```