# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belgaum -590014, Karnataka.**

**LAB REPORT**
**On**

**COMPILER DESIGN**

**Submitted by**

**JYOTHIKA C N (1BM21CS083)**

**in partial fulfillment for the award of the degree of**
**BACHELOR OF ENGINEERING**
**in**
**COMPUTER SCIENCE AND ENGINEERING**

**B.M.S. COLLEGE OF ENGINEERING**
**(Autonomous Institution under VTU)**
**BENGALURU-560019**
**Oct 2023-Feb 2024**

**B. M. S. College of Engineering,**

**Bull Temple Road, Bangalore 560019**
**(Affiliated To Visvesvaraya Technological University, Belgaum)**
**Department of Computer Science and Engineering**

CERTIFICATE

This is to certify that the Lab work entitled **“COMPILER DESIGN”** carried out by **JYOTHIKA C N(1BM21CS083)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Compiler Design Lab - **(22CS5PCCPD )**work prescribed for the said degree.

| | |
|---|---|
| **Sunayana S** | **Dr. Jyothi S Nayak** |
| Assistant Professor | Professor and Head |
| Department of CSE | Department of CSE |
| BMSCE, Bengaluru | BMSCE, Bengaluru |

# INDEX

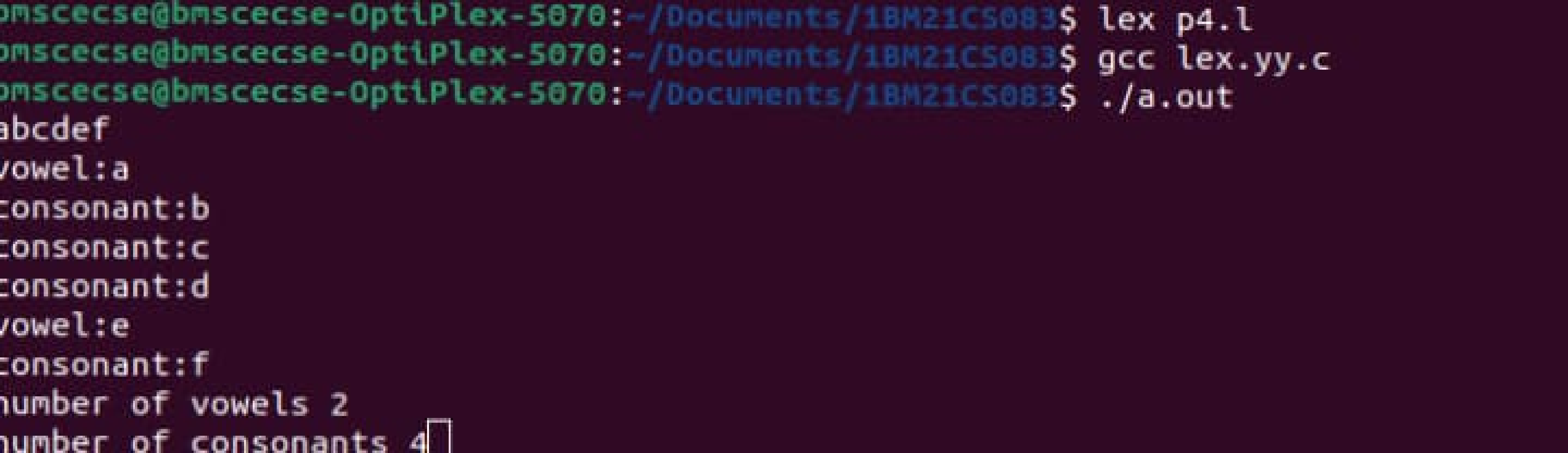
NAME: Jyothika.C.N STD.: SEC.: 5B ROLL NO.: SUB.: CD LAB

Write a lex program to identify each character as consonant or vowel in a given sentence and also count the vowel and consonants

```
%option noyywrap
%{
#include<stdio.h>
 int v=0;
 int c=0;
 %}
 %%
 [aeiouAEIOU] {v++; printf ("vowel: %s\n", yytext);}
 [a-zA-Z] {c++; printf ("consonant: %s\n", yytext);}
 [ \n] { printf ("number of vowels %d \n number of
                 consonants %d", v,c);}

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

Output

```
 airug
 vowel: a
 vowel: i
 consonant: r
 vowel :u
 consonant :g
 number of vowels 3
 number of consonants 2
```

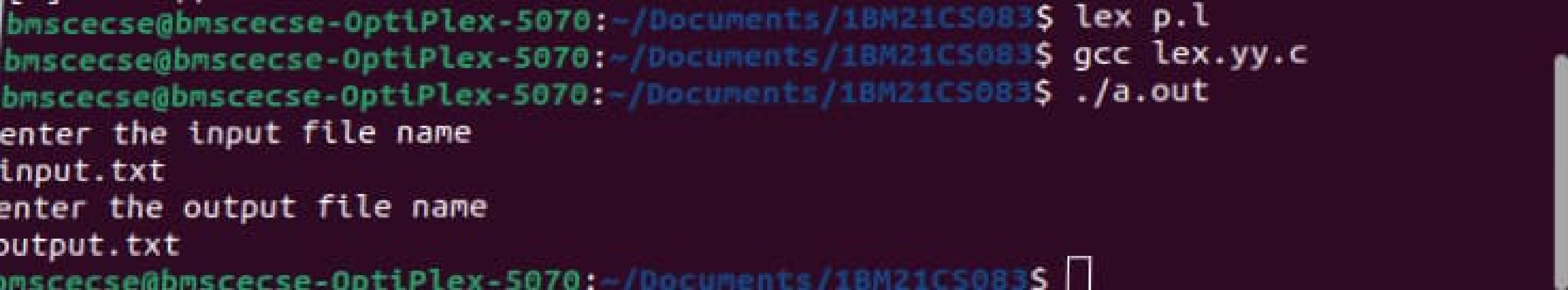


```
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex p4.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ gcc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
abcdef
vowel:a
consonant:b
consonant:c
consonant:d
vowel:e
consonant:f
number of vowels 2
number of consonants 4
```

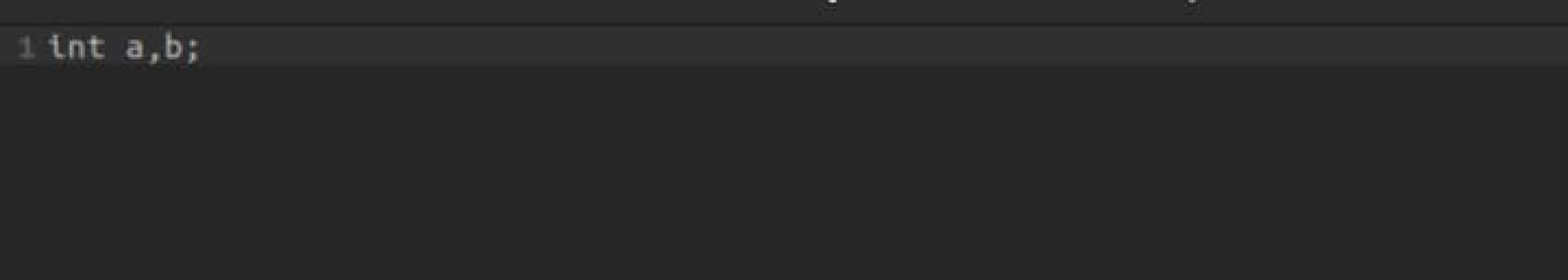
Write a lex program to read the following input from a file and print the valid token on the terminal

```
%{
char fname[20];
%}
%%
                          %s        yytext
int|float|char  {printf ("keywords");}
                   %s         yytext
[a-zA-Z]*  {printf ("Identifiers");}
               %s       yytext
,|; {printf ("seperator");}
%%
int yywrap()
{
}
```

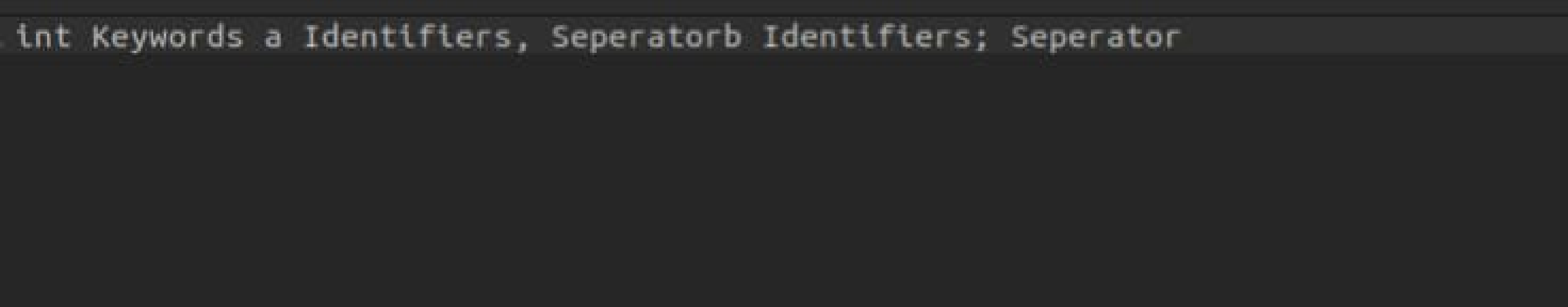
```
int yywrap()
 {
 }
void main()
 {
      printf("enter the input file name\n");
      scanf("%s", fname);
      printf("enter the output file name\n");
      scanf("%s", foname);
      yyin = fopen(fname, "r");
      yyout = fopen(foname, "w");
      yylex();
      fclose(yyin);
 }    fclose(yyin);
```



```
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex p.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ gcc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
enter the input file name
input.txt
enter the output file name
output.txt
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ 
```



```
int a,b;
```



```
int Keywords a Identifiers, Seperatorb Identifiers; Seperator
```

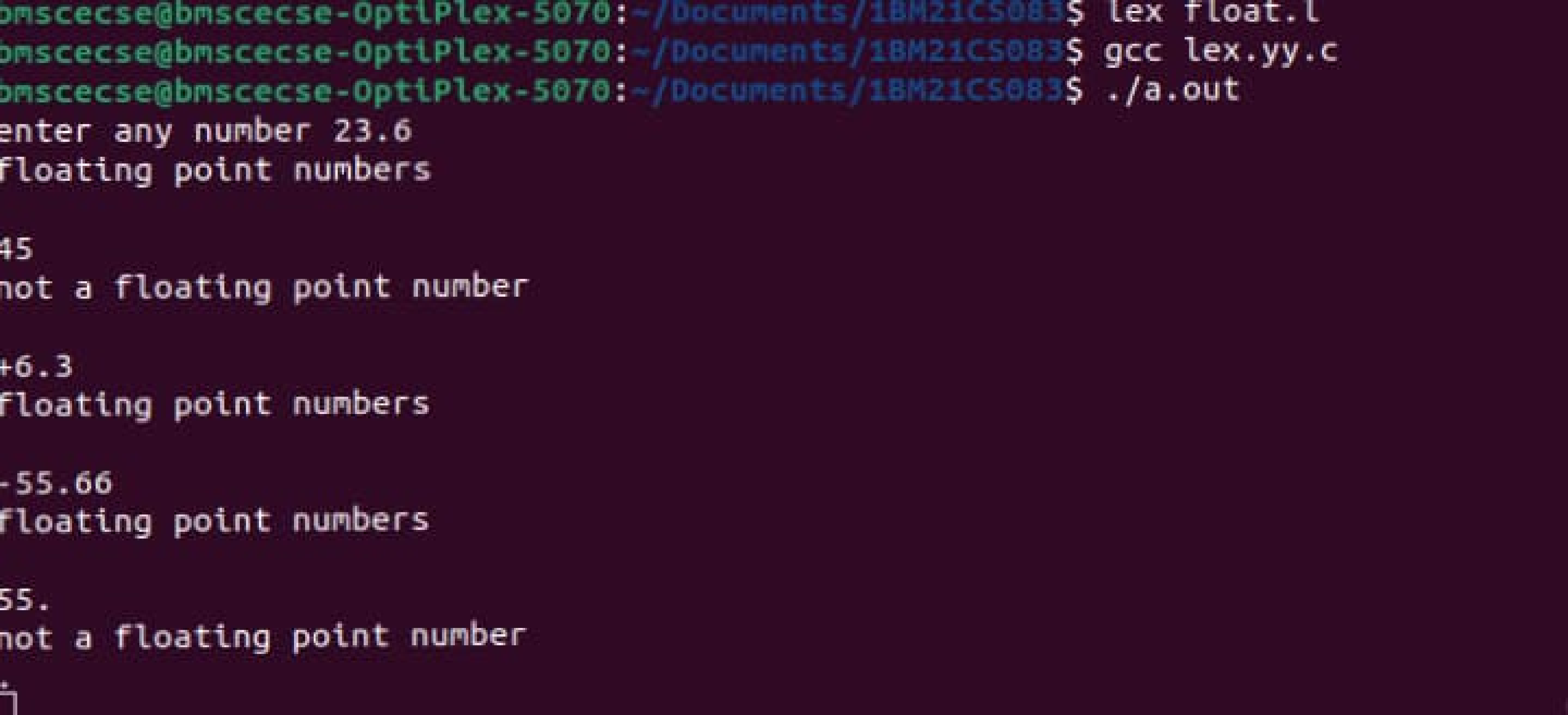
27/11/2023

Write a program in LEX to recognize Floating point number, check for all the following input cases

```
%{
%}
%%
[+-]?[0-9]*[.][0-9]+ {printf("floating point numbers\n");}
[+-]?[0-9]+ {printf("not a floating point number\n");}
%%
int yywrap()
{
}
void main()
{
    printf("enter any number");
    yylex();
}
```

Output

```
enter any number 23.6
 floating point number

45
not a floating point number

+6.3
 floating point number

-55.66
 floating point number

55.
  not a floating point number

.33
 floating point number
```

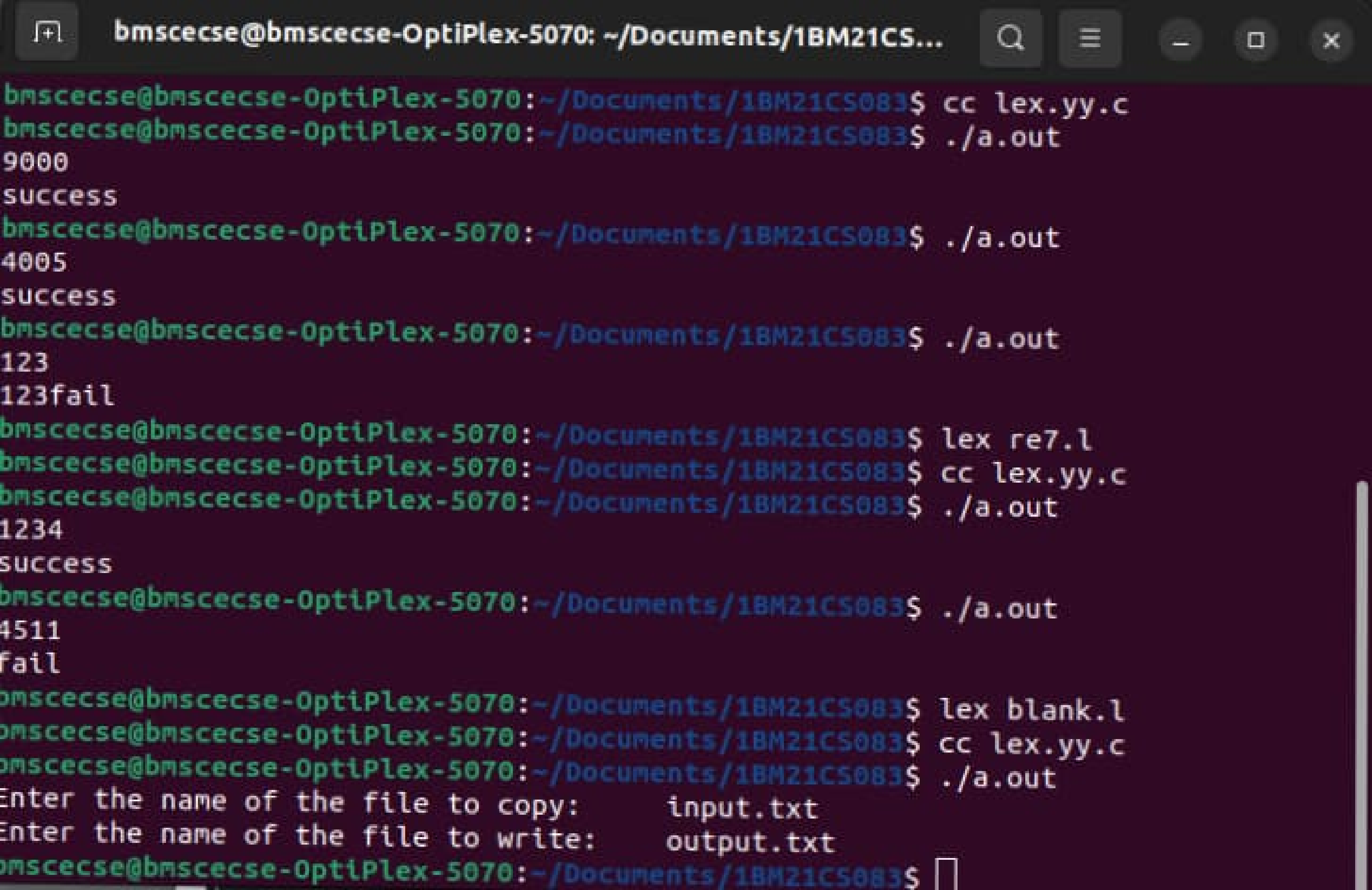


```
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex float.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ gcc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
enter any number 23.6
floating point numbers

45
not a floating point number

+6.3
floating point numbers

-55.66
floating point numbers

55.
not a floating point number

```

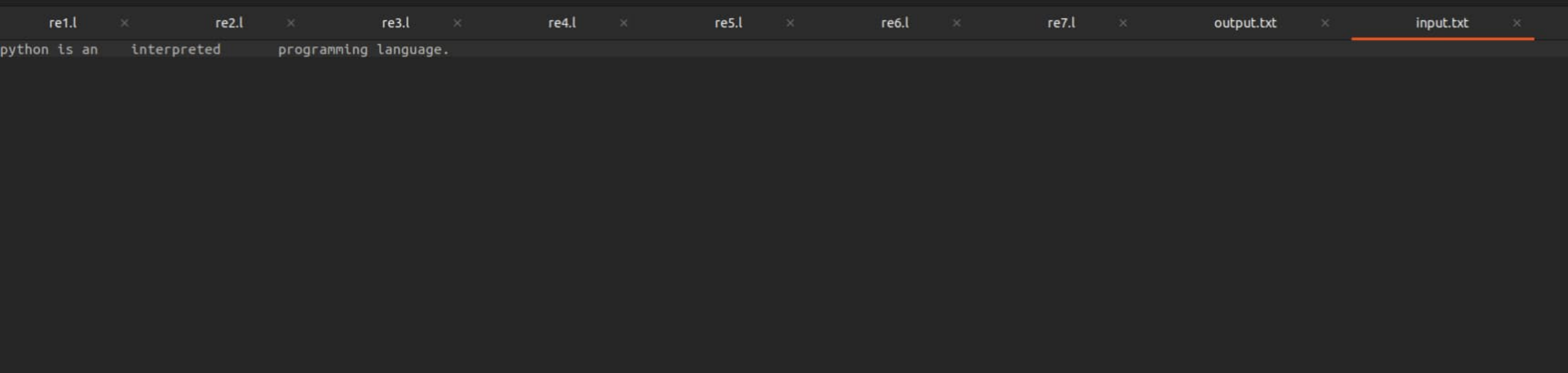
Write a LEX program that copies a file, replacing each nonempty sequence of white spaces by a single blank.

```
%{
 #include<stdio.h>
 #include<string.h>
 #include<stdlib.h>
 char str1[200];
%}

%%
[\n]  {fprintf(yyout,"%s\n", str1); str1[0]='\0'; }
 [ ]*|[\t]{ fprintf(yyout,"%s", str1); str1[0]='\0';
           fprintf(yyout "%s"," "); }
 .       strcat(str1, yytext);
<<EOF>> {fprintf(yyout, "%s", str1); return 0; }
%%

int main()
 {
    extern FILE *yyin, *yyout;
    char filename[100];
    printf(" Enter the name of the file to copy:\t");
    scanf("%s", filename);
     yyin = fopen(filename,"r");
     if (yyin==NULL)
     {
          exit(0);
     }
    printf(" Enter the name of the file to write\t");
     scanf(" %s", filename);
     yyout= fopen(filename,"w");
     if (yyout==NULL)
     {
         exit(1);
     }
     yylex();
 }
~~int~~ int   yywrap(void)
{
 }
```

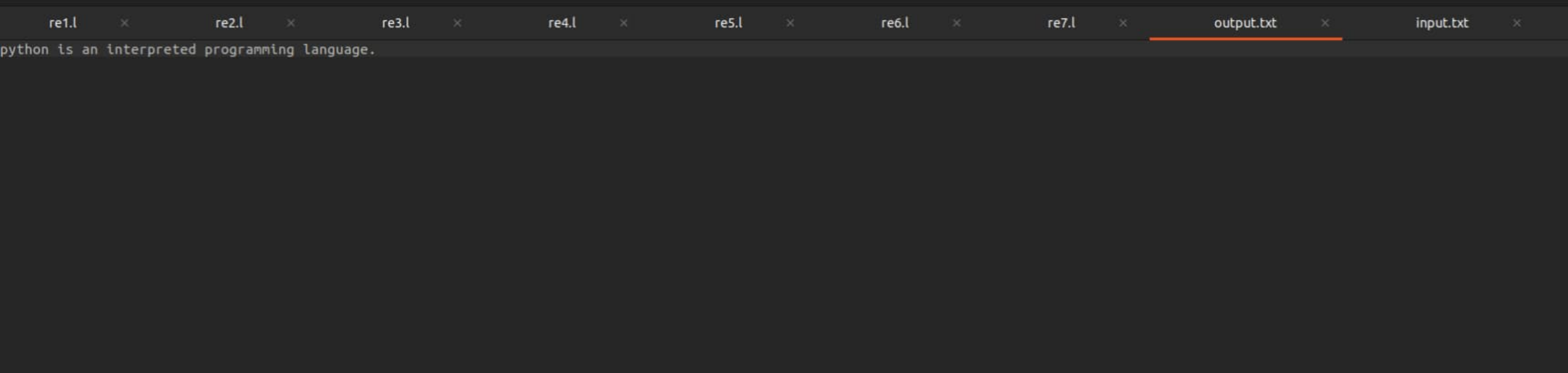
[illegible]
11/12/23



```
bmscecse@bmscecse-OptiPlex-5070: ~/Documents/1BM21CS...
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
9000
success
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
4005
success
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
123
123fail
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re7.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
1234
success
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
4511
fail
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex blank.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
Enter the name of the file to copy:    input.txt
Enter the name of the file to write:   output.txt
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ 
```



```
python is an    interpreted       programming language.
```



```
python is an interpreted programming language.
```

(i)

```
digits [0-9]
%%
{digits}*00 {printf("%s string ends with 00",yytext);}
{digits}* {printf("%s string does not end with 00",yytext);}
%%
```

Output

```
12300
12300 string ends with 00
```

(ii)

```
%%
{digits}*222{digits}* {printf("%s string has 222",yytext);}
{digits}*   {printf("%s string does not have 222",yytext);}
%%
```

Output

```
122234
122234 string has 222
```

(iii)

```
%{
 #include <math.h>
 int value=0, i,j=0, flag=0;
%}
%%
1[01]* { for (i=yyleng-1; i>=0; i--)
         {
            value += (yytext[i]-48)*pow(2,j);
            j++;
         }
         if(value%5==0)
         { flag=1;
         }
       }
[in]  return 0;
%%
```

Output

```
101
Success
```

(v)

```
%%
{digits}*1{digits}9 {printf ("%s 10th symbol from right end is
                              1",yytext);}
{digits}* {printf ("%s not", yytext);}
%%
```

Output

```
1023002245
1023002245  10th symbol from right end is 1
```

(vi)

```
%%
{digits}4 { for (i=yyleng-1; i>=0; i--)
            {
                value+ = (yytext [i] - 48);
            }
            if( value==9)
            {
                flag=1;
            }
          }
[\n] return 0;
%%
```

Output

```
4005
success
```

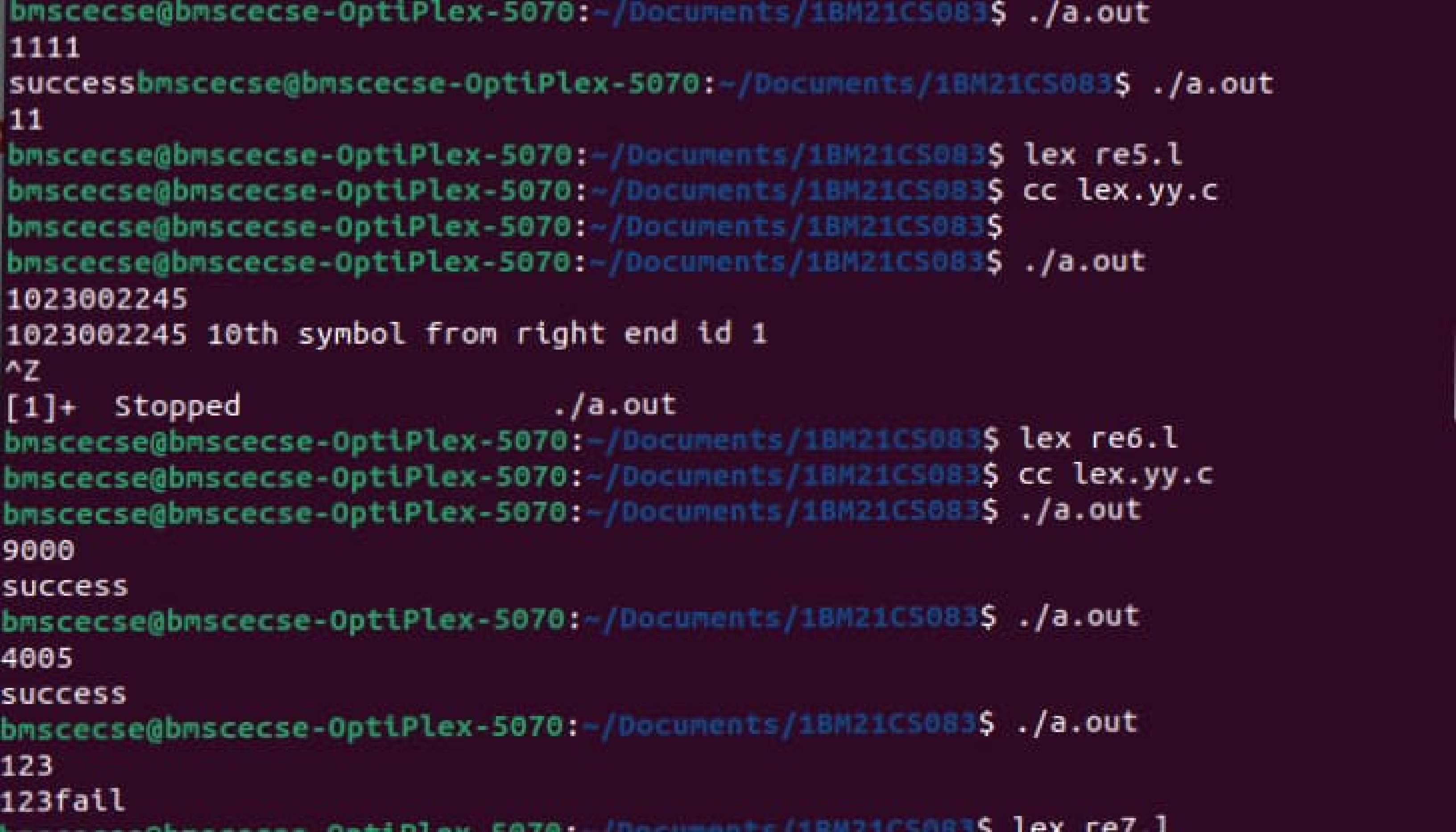
(vii)

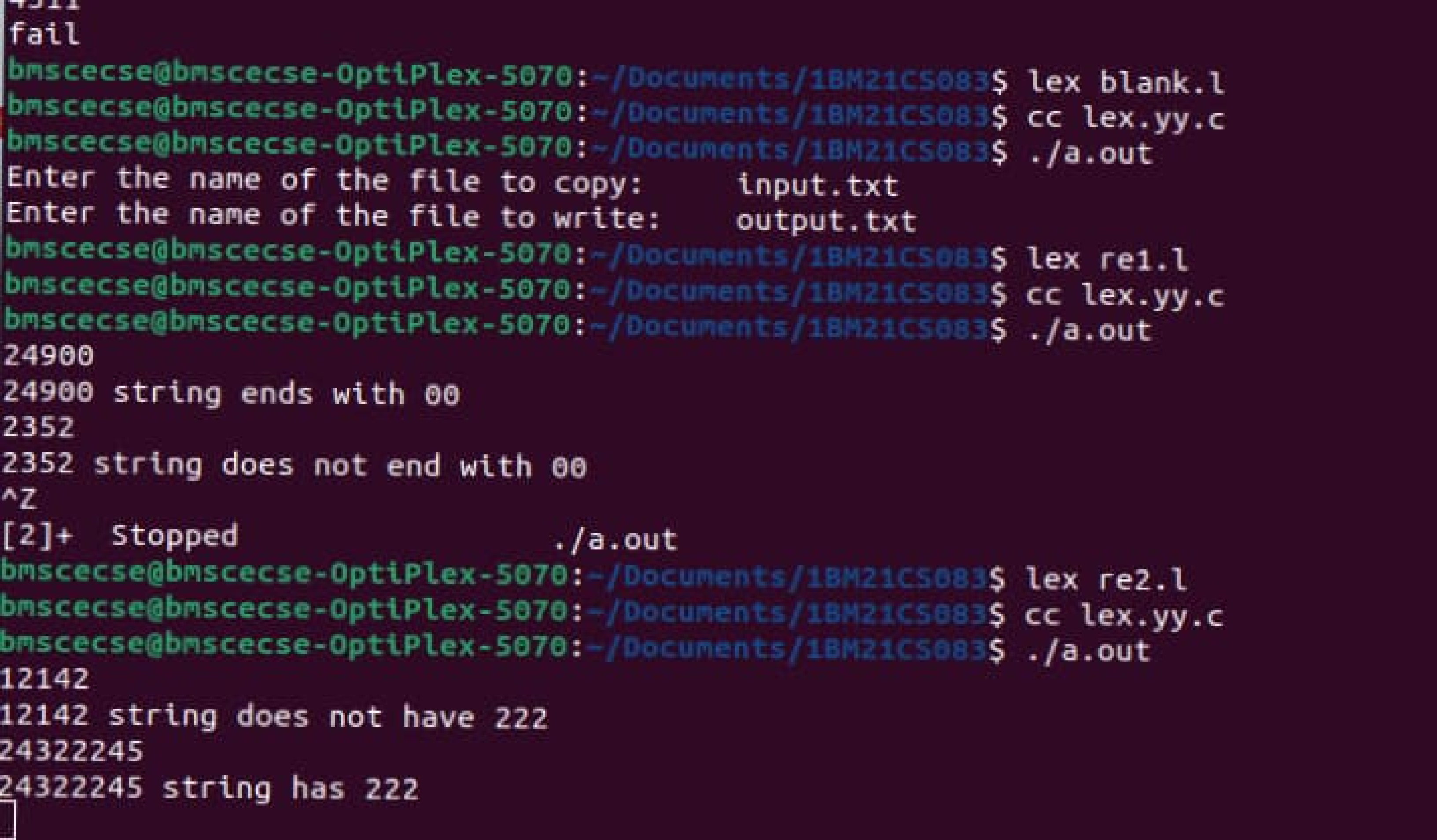
```
%%
{digits}4      { for (i=0; i<yyleng; i++)
                 {
                   if (yytext [i]> yytext[i+1])
                   {
                      flag=0;
                   }
                 }
               }
[\n] return 0;
%%
```

4/12/2023

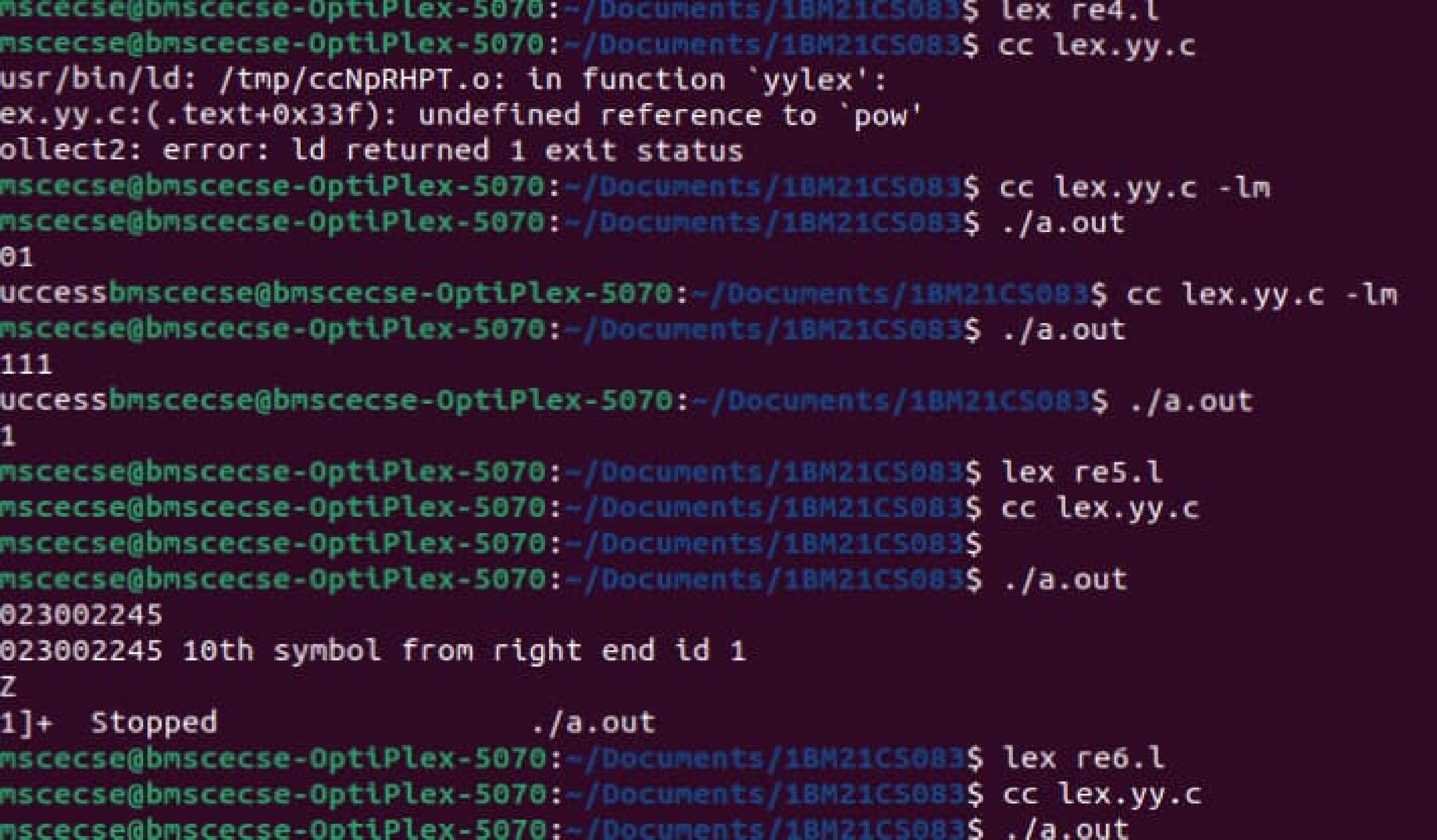
Output

```
1234
succe
```

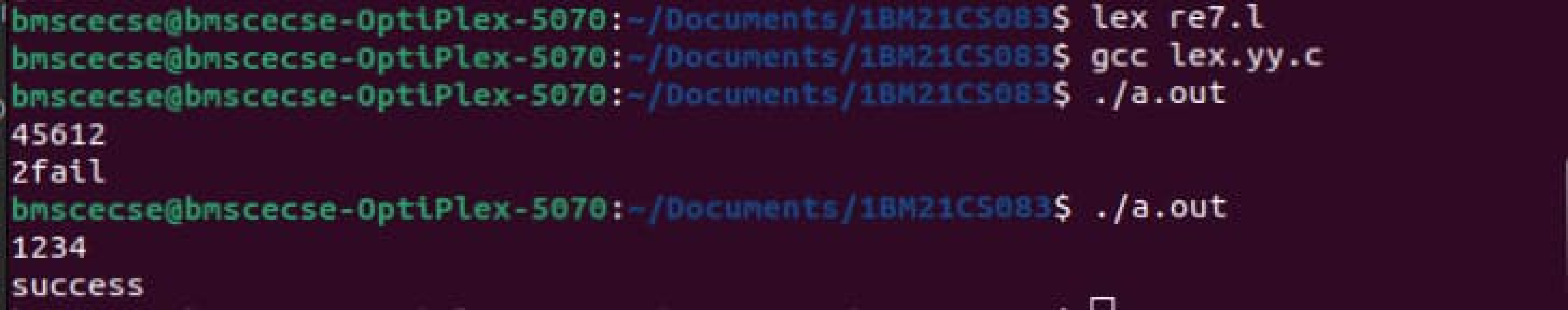

```
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
1111
successbmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
11
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re5.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
1023002245
1023002245 10th symbol from right end id 1
^Z
[1]+  Stopped                 ./a.out
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re6.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
9000
success
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
4005
success
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
123
123fail
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re7.l
```



```
4311
fail
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex blank.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
Enter the name of the file to copy:     input.txt
Enter the name of the file to write:    output.txt
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re1.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
24900
24900 string ends with 00
2352
2352 string does not end with 00
^Z
[2]+  Stopped                 ./a.out
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re2.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
12142
12142 string does not have 222
24322245
24322245 string has 222
```



```
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re4.l
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
usr/bin/ld: /tmp/ccNpRHPT.o: in function `yylex':
ex.yy.c:(.text+0x33f): undefined reference to `pow'
ollect2: error: ld returned 1 exit status
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c -lm
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
01
uccessbmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c -lm
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
111
uccessbmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
1
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re5.l
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
023002245
023002245 10th symbol from right end id 1
Z
1]+  Stopped                 ./a.out
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re6.l
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ cc lex.yy.c
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
```



```
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex re7.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ gcc lex.yy.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
45612
2fail
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
1234
success
```

Write a program to design lexical analyzer in C/C++/Java/Python language (to recognize any five keywords, identifiers, numbers, operators, and punctuations)

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
void lexicalAnalyzer (char input_code[])
{  char *keywords[] = {"if", "else", "while", "for", "return",
                       "int" "float"};
   char *operators[] = {"+", "-", "*", "/", "=", "==", "<", ">",
                        "<=" ">=", "!="};
   char *punctuations [] = {",", ";", "(", ")", "{", "}"};
   char *tokens = strtok (input_code, " \t\n");
   while ( token != NULL)
   {
       if (isdigit (token [0]))
       {
           printf ("Number :%s\n", token);
       }
       else if (isalpha(token[0]) || token[0] == '_')
       {
           int isKeyword = 0;
           for(int i=0; i< sizeof(keywords)/ sizeof (keywords[0]);
                                                              i++)
           {
               if(strcmp (token, keywords[i] == 0)
               {
                   printf ("Keyword : %s\n", token);
                   isKeyword = 1; break;
               }
           }
           if (! isKeyword)
           {
               printf(" Identifier : %s\n", token);
           }
       }
       else if (strchr ("+-*/=<>;(),!", token[0]) != NULL)
       {
           printf("Punctuation /Operator : %s\n", token);
       }
       token = strtok (NULL, " \t\n"); //next token
   }
}
```

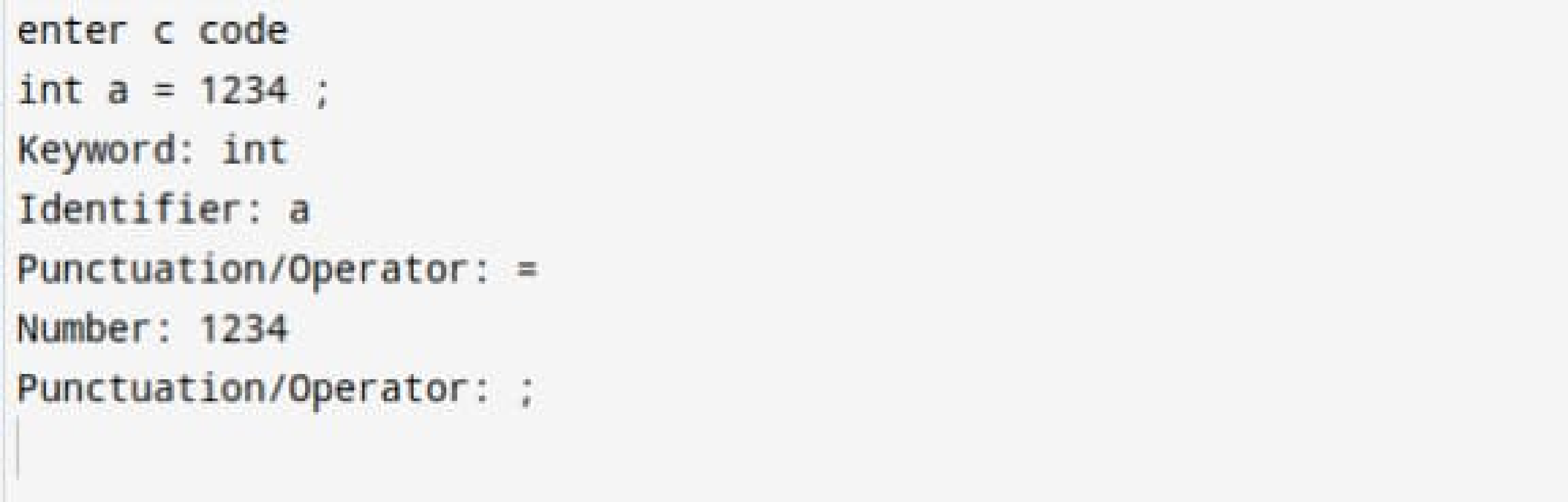
```
int main()
{
        char input_code [200];
        printf ("enter c code\n");
        fgets (input_code, 200, stdin);
        lexicalAnalyzer (input_code);
        return 0;
}
```

Output

```
enter c code
int a = 1234;
Keyword :int
Identifier : a
Punctuation/Operator : =
Number: 1234
Punctuation/operator : ;
```

9/10

[illegible]
18/12/2023



```
enter c code
int a = 1234 ;
Keyword: int
Identifier: a
Punctuation/Operator: =
Number: 1234
Punctuation/Operator: ;
```

write a program to perform Recursive descent on the following grammar

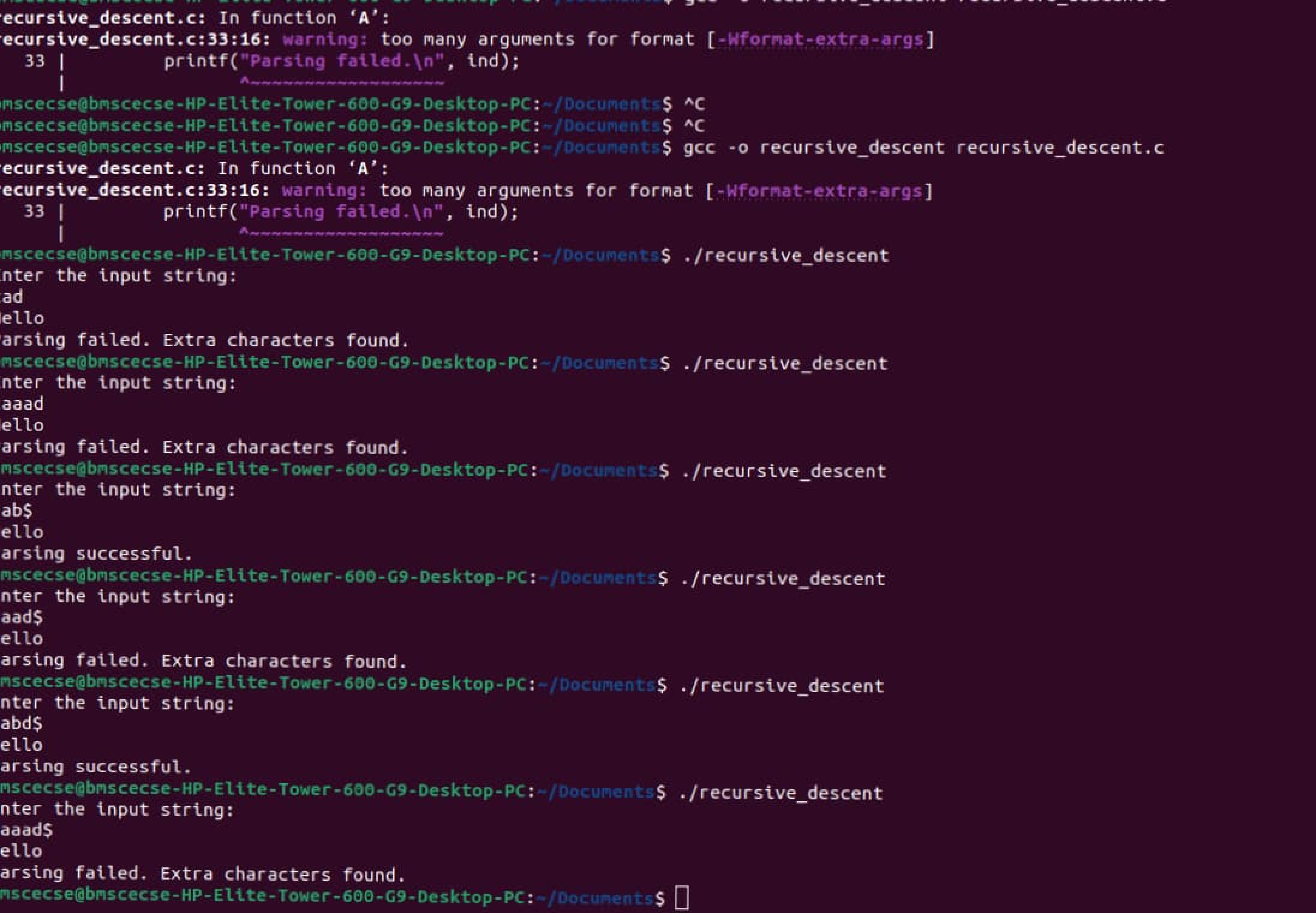
S → cAd,
A → ab/a

```
#include<stdio.h>
#include<stdlib.h>
char input [100];
int ind=0;
void match (char expected)
{
    if (input[ind] == expected)
    {
        ind++;
    }
}
void A();
void S()
{
    match ('c');
    A();
    match ('d');
}
~~void A();~~
~~void S()~~
{
    ~~match ('c');~~
void A()
{
    if(input[ind]=='a')
    {
        printf ("Hello\n");
        match('a');
        match ('b');
    }
    else
    {
        printf("Parsing failed.\n", ind);
        exit(1);
    }
}
```

```
int main()
{
    printf("Enter the input string :\n");
    scanf ("%s", input);

    S();
    if (input[ind] == '$')
    {
        printf("Parsing succesfull.\n");

    }
    else
    {
        printf ("Parsing failed. Extra character found.\n");

    }
    return 0;
}
```

Output

```
Enter the input string
 cabd $
 Hello
Parsing Succesful.
```



```
ecursive_descent.c: In function ‘A’:
ecursive_descent.c:33:16: warning: too many arguments for format [-Wformat-extra-args]
   33 |         printf("Parsing failed.\n", ind);
      |                ^~~~~~~~~~~~~~~~~~~
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ^C
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ^C
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ gcc -o recursive_descent recursive_descent.c
ecursive_descent.c: In function ‘A’:
ecursive_descent.c:33:16: warning: too many arguments for format [-Wformat-extra-args]
   33 |         printf("Parsing failed.\n", ind);
      |                ^~~~~~~~~~~~~~~~~~~
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./recursive_descent
nter the input string:
ad
ello
arsing failed. Extra characters found.
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./recursive_descent
nter the input string:
aaad
ello
arsing failed. Extra characters found.
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./recursive_descent
nter the input string:
ab$
ello
arsing successful.
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./recursive_descent
nter the input string:
aad$
ello
arsing failed. Extra characters found.
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./recursive_descent
nter the input string:
abd$
ello
arsing successful.
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./recursive_descent
nter the input string:
aaad$
ello
arsing failed. Extra characters found.
mscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ 
```

Write a yacc program to parse strings grammar { $a^nb$ $n \geq 5$

anbn.l

```
%{
#include <stdio.h>
#include <stdlib.h>
#include "y.tab.h"
extern int yyval;
%}
%%
[aA]  {yylval =yytext[0]; return A;}
[bB]  {yyval =yytext [0]; return B;}
\n   {return NL;}
.    {return yytext[0];}
%%
int yywrap()
{
  return 1;
}
```

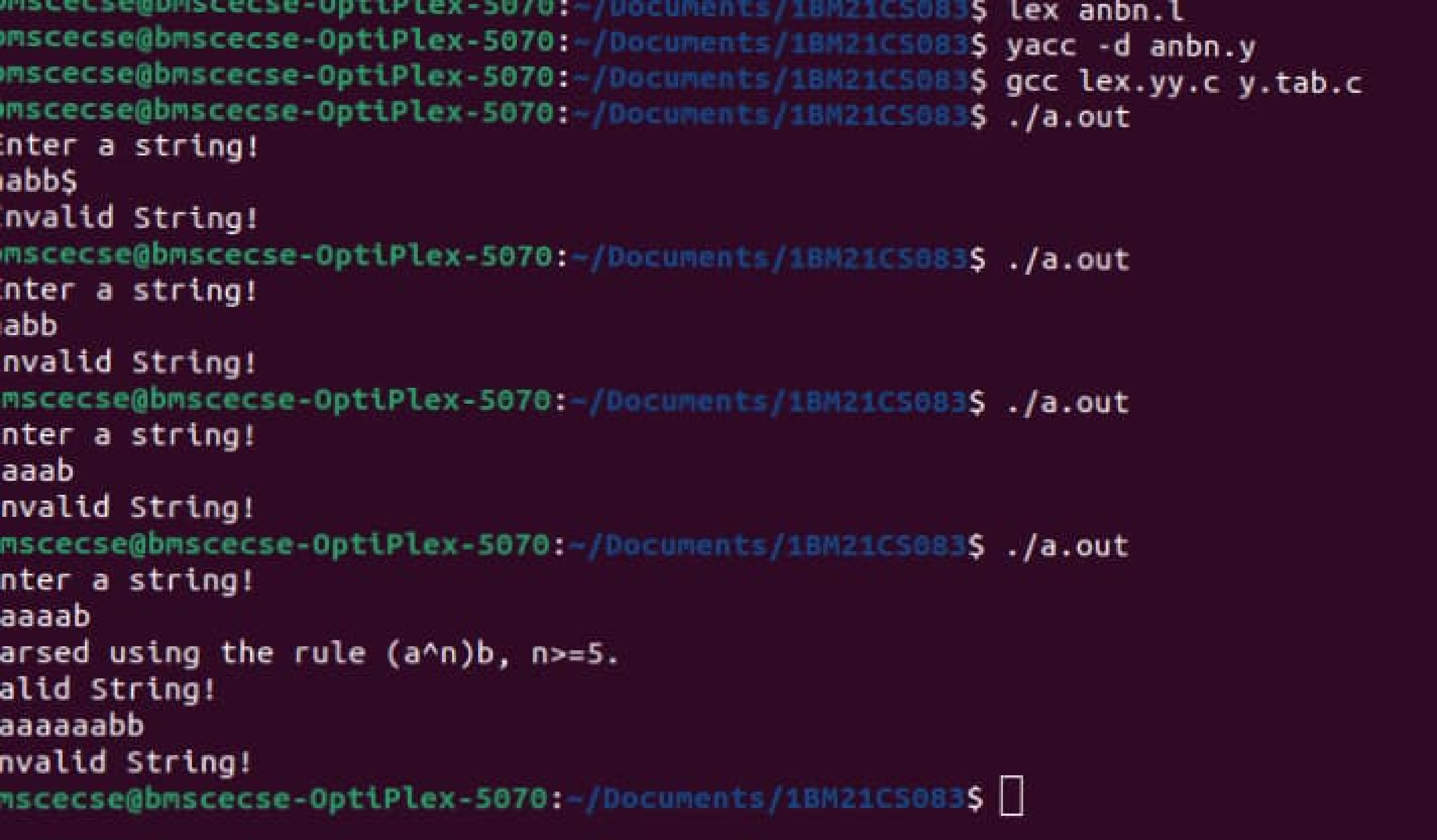
anbn.y

```
%{
#include <stdio.h>
#include <stdlib.h>
int yyerror (char *s).
int yylex (void).
%}
%token A
%token B
%token NL
%%
smtr : AAAAASB NL {printf ("Parsed using the rule (a^n b),
;         n>=5, Invalid String !\n"); }
;
S: SA
|
;
%%
```

```
void main()
{
   printf("Enter a string!\n");
   yyparse();
}
int yyerror(char *s)
{
   printf("Invalid string!\n");
   return 0;
}
```

Output

```
$ lex anbn.l
$ yacc -d anbn.y
$ gcc lex.yy.c y.tab.c
$ ./a.out
Enter a string!
aaaaab
Parsed using the rule (a^n)b, n>=5
Valid string!
aabb
Invalid string!
```



```
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex anbn.l
omscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ yacc -d anbn.y
omscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ gcc lex.yy.c y.tab.c
omscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
Enter a string!
aabb$
Invalid String!
omscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
nter a string!
abb
nvalid String!
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
nter a string!
aaab
nvalid String!
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
nter a string!
aaaab
arsed using the rule (a^n)b, n>=5.
alid String!
aaaaaabb
nvalid String!
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ 
```

Design a suitable grammar for evaluation of arithmetic expression having +, -, *, /, %, ^ operator.

^ highest precedence and right
% second highest and left
*, / second highest and left
+, - lowest precedente and left

prool.l

```
%option noyywrap
%{
#include "y.tab.h"
%}
%%
[0-9]+ {yylval = atoi (yytext); return NUM; }
[\t] ;
\n return 0.
. return yytext[0].
%%
```

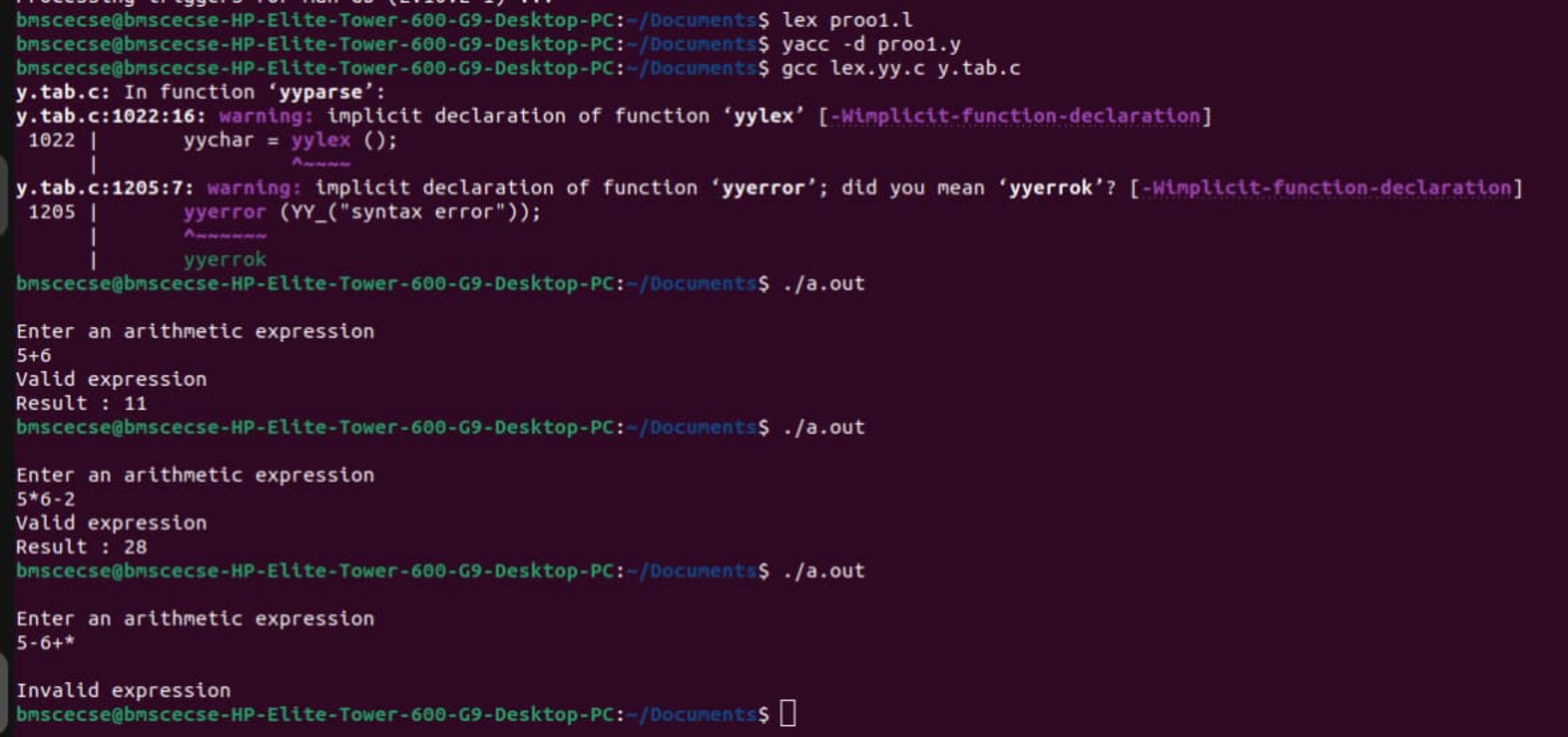
prool.y

```
%{
#include<stdio.h>
%}
%token NUM
%left '+' '-'
%left '*' '/'
%left '%'
%right '^'
%%
expr : e { ~~printf("valid exprerion \n");~~ printf("Result :%d \n", $$);
                                                   return 0; }

e : e '+' e  { $$ = $1 + $3; }
  | e '-' e  { $$ = $1 - $3; }
  | e '*' e  { $$ = $1 * $3; }
  | e '/' e  { $$ = $1 / $3 ; }
  | e '%' e  { $$ = $1 % $3; }
```

```
$ | e '^'e      { $$ = $3 ^ $1; }
  | NUM         { $$ = $1; }
  ;
%%
int main()
{
   printf("\nEnter an arithmetic expression \n");
   yyparse();
   return 0;
}
int yyerror()
{
   printf("\nInvalid expression \n");
   return 0;
}
```

Output

```
Enter an arithmetic expression
 5+6
Valid expression
Result : 11
Enter an arithmetic expression
 5 + 6+*
Invalid expression
```



```
Processing triggers for man-db (2.10.2-1) ...
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ lex proo1.l
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ yacc -d proo1.y
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ gcc lex.yy.c y.tab.c
y.tab.c: In function ‘yyparse’:
y.tab.c:1022:16: warning: implicit declaration of function ‘yylex’ [-Wimplicit-function-declaration]
 1022 |       yychar = yylex ();
      |                ^~~~~
y.tab.c:1205:7: warning: implicit declaration of function ‘yyerror’; did you mean ‘yyerrok’? [-Wimplicit-function-declaration]
 1205 |       yyerror (YY_("syntax error"));
      |       ^~~~~~~
      |       yyerrok
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./a.out

Enter an arithmetic expression
5+6
Valid expression
Result : 11
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./a.out

Enter an arithmetic expression
5*6-2
Valid expression
Result : 28
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./a.out

Enter an arithmetic expression
5-6+*

Invalid expression
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ 
```

Write a yacc program to generate syntax tree for the given arithmetic expression.

p.l

```
%{
#include "y.tab.h"
extern int yylval;
%}
%%
[0-9]+  {yylval = atoi (yytext ); return digit;}
[\t]; 
[\n] return 0;
. return yytext [0];
%%
int yywrap()
{
}
```

P.y

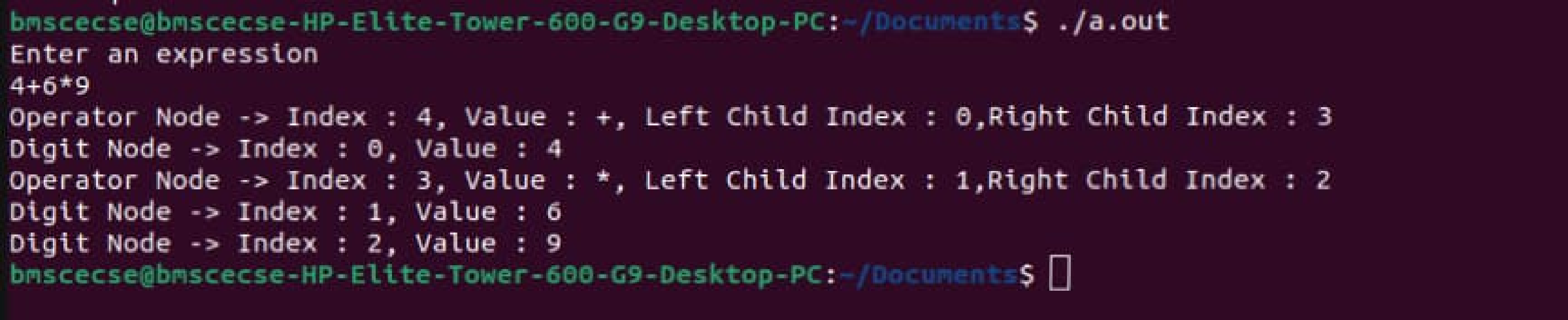
```
%{
#include <math.h>
#include<ctype.h>
#include <stdio.h>
#include< stdlib.h>
#include <string.h>
struct tree-node
{
    char val[10];
    int lc;
    int rc;
};
int ind;
struct ~~tree_node~~ syn_tree [100];
void ~~my~~ print_tree (int cur_ind);
int mknode (int lc, int rc, char val [10]);
%}
% token digit
```

```
%%
S:E {my_print_tree($1);}
;
E:E'+'T {$$ = mknode($1, $3, "+");; }
|T {$$ = $1; }
;
T:T'*'F {$$ = mknode($1, $3, "*");; }
|F {$$ = $1; }
;
F: '('E')' {$$ = $2; }
|digit {char buff[10]; sprintf(buf,"%d",yylval);
                $$ = mknode(-1, -1, buf);}
%%
int main()
{
    ind=0;
    printf("Enter an expression\n");
    yyparse();
    return 0;
}
int yyerror()
{
    printf("NITW Error\n");
}
int mknode(int lc, int rc, char val[10])
{
    strcpy(syn_tree[ind].val, val);
    syn_tree[ind].lc = lc;
    syn_tree[ind].rc = rc;
    ind++; return ind-1;
}
void my_print_tree(int cur_ind)
{
    if(cur_ind == -1)
        return;
    if (syn_tree[cur_ind].lc == -1 && syn_tree[cur_ind].rc==
        printf("Digit Node -> Index : %d" Value : %s\n",
            cur_ind, syn_tree[cur_ind].val);
    else
printf("Operator Node → Index : %d, Value %s , Left
    child Index : %d, Right child Index : %d\n",
    curr_ind, syn_tree[curr_ind].val, syn_tree[curr_ind].lc,
    syn_tree[curr_ind].rc);

    my_print_tree (syn_tree[curr_ind].lc);
    my_print_tree (syn_tree[curr_ind].rc);

}
```

Output :

```
Enter an expression
4+6*9
Operator Node → Index : 4, Value : + , Left child Index : 0,
                                 Right child Index : 3
Digit Node → Index : 0 , Value : 4
Operator Node → Index : 3, Value : * , Left child Index : 1,
                                 Right child Index : 2
Digit Node → Index : 1 , Value : 6
Digit Node → Index : 2, Value : 9
```

[illegible]
11/1/2024



```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ ./a.out
Enter an expression
4+6*9
Operator Node -> Index : 4, Value : +, Left Child Index : 0,Right Child Index : 3
Digit Node -> Index : 0, Value : 4
Operator Node -> Index : 3, Value : *, Left Child Index : 1,Right Child Index : 2
Digit Node -> Index : 1, Value : 6
Digit Node -> Index : 2, Value : 9
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Documents$ 
```

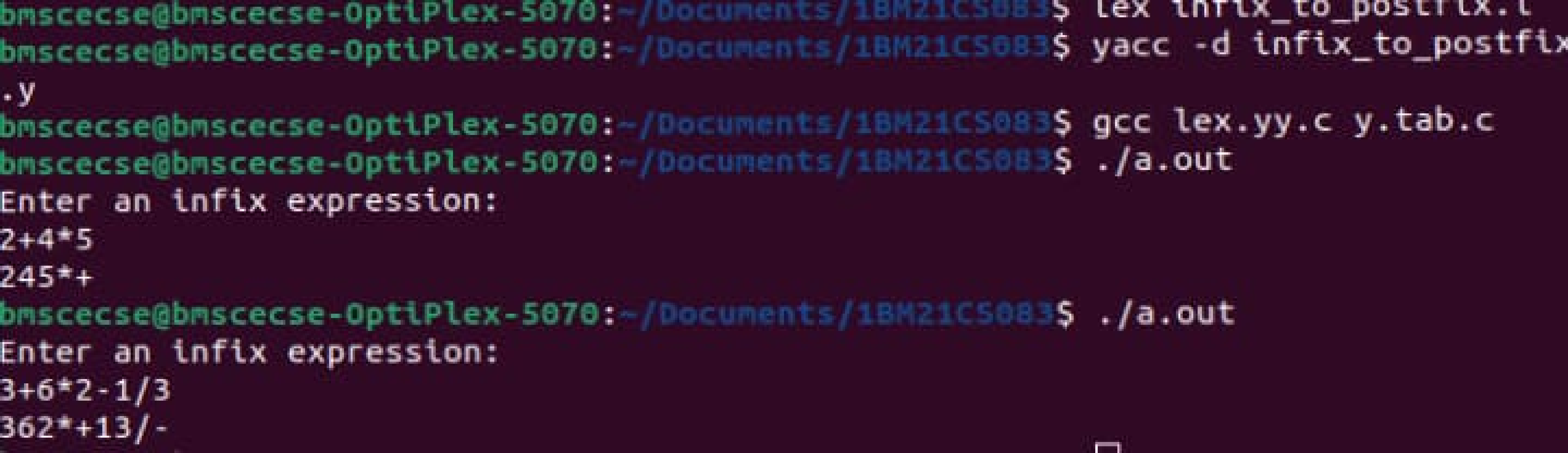
Use YACC to convert : Infix expression to Postfix expression

```
infix_to_postfix.l
%{
#include <stdio.h>
#include <stdlib.h>
#include "y.tab.h"
extern int yylval;
%}
%%

[0-9]+ {yylval =atoi(yytext); return num;}
[\t ];
\n {~~r~~return 0;}
. { return yytext [0];}
%%
int yywrap()
{
}

infix_to_prefix.y
%{
#include <stdio.h>
#include <stdlib.h>
int yyerror (const char *s);
int yylex(void);
%}

%token num

%left '+''-'
%left '*''/'
%left '('
%left ')'
%right '^'
%%
```

```
s: e {printf("\n");}
;
e: e'+'t {printf("+");}
 | e'-'t {printf("-");}
 |t
;
t: t'*'h {printf("*");}
 |t '/'h {printf("/");}
 |h
;
h: f '^'h {printf("^");}
 | f
;
f: '('e')'
 |num {printf("%d", $1);}
;
%%
void main()
{
  printf("Enter an infix expression:\n");
  yyparse();
}
int yyerror(const char *s)
{
  printf("Invalid infix expression.\n");
  return 0;
}
```

Output

```
Enter an infix expression:
2+4*5
245*+

Enter an infix expression:
3+6*2-1/3
362*+13/-
```



```
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex infix_to_postfix.l
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ yacc -d infix_to_postfix
.y
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ gcc lex.yy.c y.tab.c
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
Enter an infix expression:
2+4*5
245*+
bmscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
Enter an infix expression:
3+6*2-1/3
362*+13/-
```

Use YACC to generate 3-Address code for a given expression.

3addcode.l

```
%{
#include <stdio.h>
#include <stdlib.h>
#include "y.tab.h"
extern int yylval;
extern char iden[20];
%}
d [0-9]+
a [a-z A-z]+
%%
{d}     {yylval= atoi(yytext); return digit;}
{a}     { strcpy (iden, yytext ); yylval=1; return id;}
[ \t]   {;}
\n      return 0;
.       return yytext[0];
%%
int yywrap()
{
  return 1;
}
```

3addcode.y

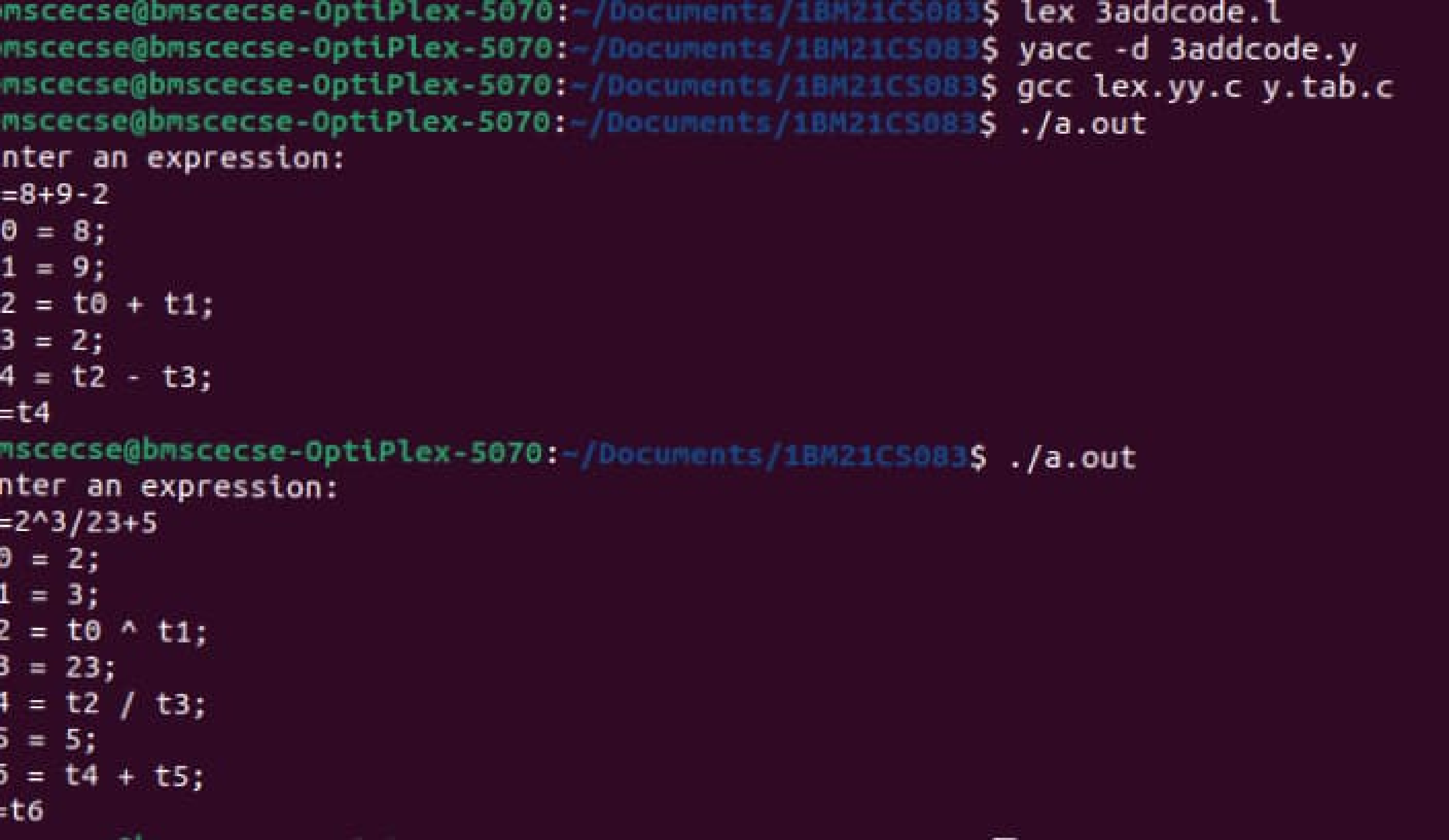
```
%{
#include <math.h>
#include <ctype.h>
#include <stdio.h>
int yyerror (char *s);
int yylex (void);
int. var_cnt=0;
char iden[20];
%}
```

```
%token id
%token digit
%%
S: id '=' E {printf("%s= t%d\n", idon, var_cnt-1);}
E: E '+' T {$$ = var_cnt; var_cnt++; printf("t%d= t%d + t%d;\n", $$, $1, $3);}
|E '-' T {$$= var_cnt; var_cnt++; printf("t%d = t%d - t%d;\n", $$, $1, $3);}
|T {$$ = $1;}
;
T: T '*' F {$$ =var_cnt; var_cnt++; printf("t%d= t%d * t%d ;\n", $$, $1, $3);}
|T '/' F {$$ = var_cnt; varcnt++; printf("t%d = t%d / t%d ;\n", $$, $1, $3);}
|F {$$ = $1;}

;
F: P '^' F {$$ = var_cnt; var_cnt++; printf("t%d= t%d ^t%d;\n", $$, $1, $3);}
|P {$$, $1;}
;
P: '(' E ')' {$$=$2;}
|digit {$$ =var_cnt; var_cnt++; printf("t%d= %d;\n", $$, $1);}
;
%%
int main()
{
    var_cnt=0;
    printf("Enter an expression:\n");
    ~~yyparse();~~
    return 0;
}
int yyerror (char *s)
{
   printf ("Invalid expression!");
   return 0;
}
```

Output

```
Enter an expression:
a = 8+9-2
 t0 = 8;
 t1 = 9;
 t2 = t0+t1;
 t3 = 2;
 t4 = t2-t3;
a = t4
```

[illegible]
29/1/2024



```
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ lex 3addcode.l
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ yacc -d 3addcode.y
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ gcc lex.yy.c y.tab.c
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
nter an expression:
=8+9-2
0 = 8;
1 = 9;
2 = t0 + t1;
3 = 2;
4 = t2 - t3;
=t4
mscecse@bmscecse-OptiPlex-5070:~/Documents/1BM21CS083$ ./a.out
nter an expression:
=2^3/23+5
0 = 2;
1 = 3;
2 = t0 ^ t1;
3 = 23;
4 = t2 / t3;
5 = 5;
6 = t4 + t5;
=t6
```