**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**

**on**

**COMPILER DESIGN**

*Submitted by*

**Manoj Patil(1BM21CS104)**

***Under the Guidance of***

**Prof. Prameetha Pai**

**Assistant Professor, BMSCE**

*in partial fulfilment 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**

**November 2023-February 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 **Manoj Patil(1BM21CS104)** , who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023-24.

The Lab report has been approved as it satisfies the academic requirements in respect of **Compiler Design- (22CS5PCCPD)** work prescribed for the said degree.

Prof. Prameetha Pai Dr. Jyothi Nayak

Assistant professor Professor and Head

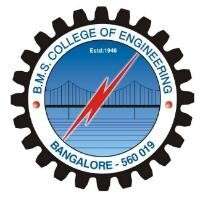
Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

**B. M. S. COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND**

**ENGINEERING**



***DECLARATION***

I, Manoj Patil(1BM21CS104), student of 5th Semester, B.E, Department of Computer Science and Engineering, B. M. S. College of Engineering, Bangalore, here by declare that, this lab report entitled " **Compiler Design**" has been carried out by me under the guidance of Prof. Prameetha Pai, Assistant Professor, Department of CSE, B. M. S. College of Engineering, Bangalore during the academic semester November-2023-February-2024.

I also declare that to the best of my knowledge and belief, the development reported here is not from part of any other report by any other students.

**TABLE OF CONTENTS**

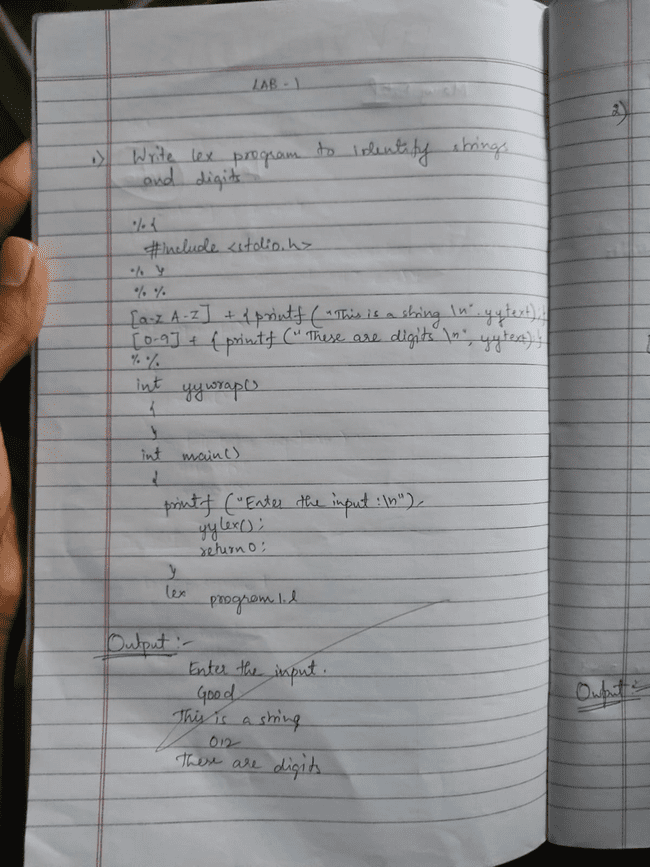
|  |  |  |
| --- | --- | --- |
| **Lab No** | **Title** | **Page No** |
| **1** |  | **6-7** |
| 1.1 | Write a program in LEX to recognize different tokens: Keywords, Identifiers, Constants, Operators and Punctuation symbols**.** | 6 |
| 1.2 | Write a program in LEX to count the number of vowels and consonants in a string. | 7 |
| **2** |  | **8-15** |
| 2.1 | Write a program in lex to count the number of words in a sentence. | 8 |
| 2.2 | Write a program in lex to demonstrate regular definition. | 9 |
| 2.3 | Write a program in lex to identify tokens in a program by taking input from a file and printing the output on the terminal. | 10-11 |
| 2.4 | Write a program in lex to identify tokens in a program by taking input from a file and printing the output in another file. | 12 |
| **3** |  | **14-22** |
| 3.1 | Write a program in LEX to recognize Floating Point Numbers. | 14-15 |
| 3.2 | Read and input sentence, and check if it is compound or simple. If a sentence has the word- and , or ,but ,because ,if ,then ,nevertheless then it is compound else it is simple. | 16-17 |
| 3.3 | Write a program to check if the input sentence ends with any of the following punctuation marks ( ? , fullstop , ! ) | 18 |
| 3.4 | Write a program to read an input sentence and to check if the sentence begins with  English articles (A, a,AN,An,THE and The). | 19 |
| 3.5 | Lex program to count the number of comment lines (multi line comments or single line) in a program. Read the input from a file called input.txt and print the count in a file called output.txt. | 20-21 |
| 3.6 | Write a program to read and check if the user entered number is signed or unsigned using appropriate meta character. | 22 |
| **4** |  | **23-33** |
| 4.1 | Write a LEX program that copies a file, replacing each nonempty sequence of white spaces by a single blank. | 23-24 |
| 4.2 | Write a LEX program to recognize the following tokens over the alphabets {0,1,..,9} | 25 |

|  |  |  |
| --- | --- | --- |
| 4.2.1 | The set of all string ending in 00. | 26 |
| 4.2.2 | The set of all strings with three consecutive 222’s. | 27 |
| 4.2.3 | The set of all string such that every block of five consecutive symbols contains at least two 5’s. | 29 |
| 4.2.4 | The set of all strings beginning with a 1 which, interpreted as the binary representation of an integer, is congruent to zero modulo 5. | 30 |
| 4.2.5 | The set of all strings such that the 10th symbol from the right end is 1. | 31 |
| 4.2.6 | The set of all four digits numbers whose sum is 9. | 32 |
| 4.2.7 | The set of all four digital numbers, whose individual digits are in ascending order from left to right. | 33 |
| **5** |  | **35** |
| 5.1 | Write a C program to design lexical analysis to recognize any five keywords, identifiers, numbers, operators and punctuations. | 35-37 |
| **6** |  | **38-40** |
| 6.1 | Write a program to perform recursive descent parsing on the following grammar: S->cAd  A->ab | a | 39-40 |
| **7** |  | **41-43** |
| 7.1 | Write a program in YACC to design a suitable grammar for evaluation of arithmetic expression having +, -, \* and /. | 41-43 |
| 7.2 | Write a program in YACC to recognize strings of the form {(a^n)b,n>=5}. | 44-45 |
| 7.3 | Write a program in YACC to generate a syntax tree for a given arithmetic expression. | 46-49 |
| **8** |  | **50-51** |
| 8.1 | Write a program in YACC to convert infix to postfix expression. | 50-51 |
| **9** |  | **52-55** |
| 9.1 | Write a program in YACC to generate three address code for a given expression. | 50-51 |

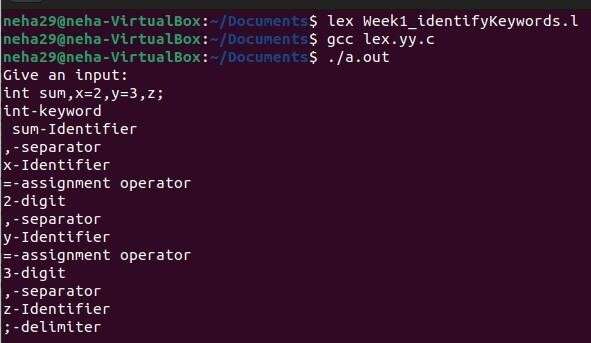
# Lab 1

**1.1 Write a program in LEX to recognize different tokens: Keywords, Identifiers, Constants, Operators and Punctuation symbols.**

**Code:**

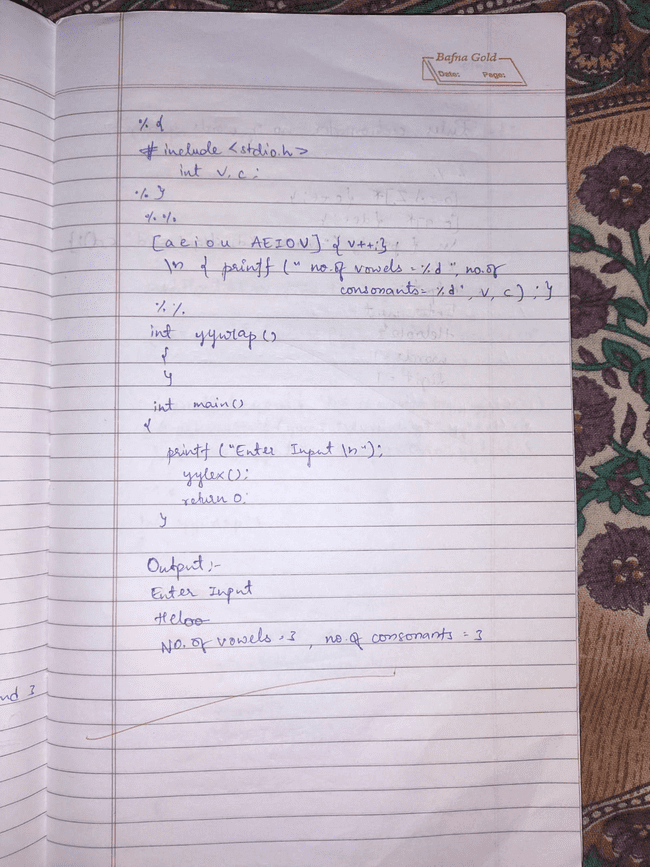


**Output**



**1.2 Write a program in LEX to count the number of vowels and consonants in a string.**

**Code**



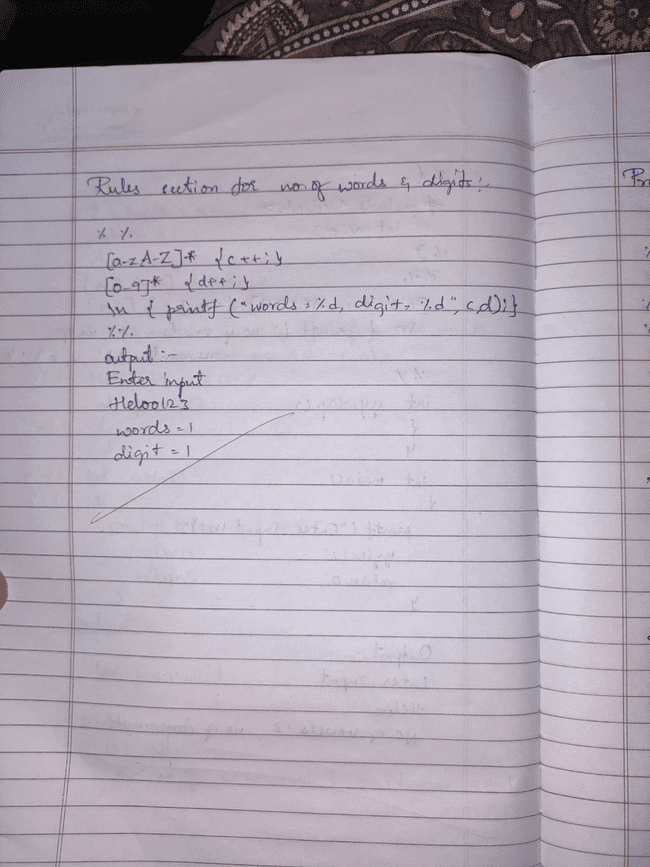
**Output**



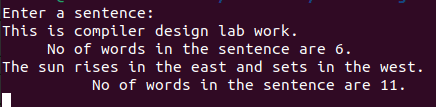
**Lab 2**

**2.1Write a program in lex to count the number of words in a sentence.**

**Code**



**Output**

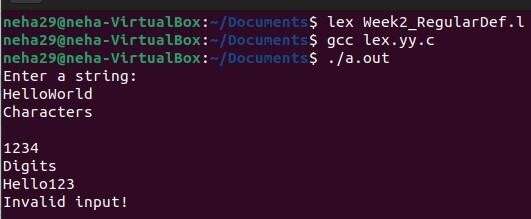


**2.2 Write a program in lex to demonstrate regular definition.**

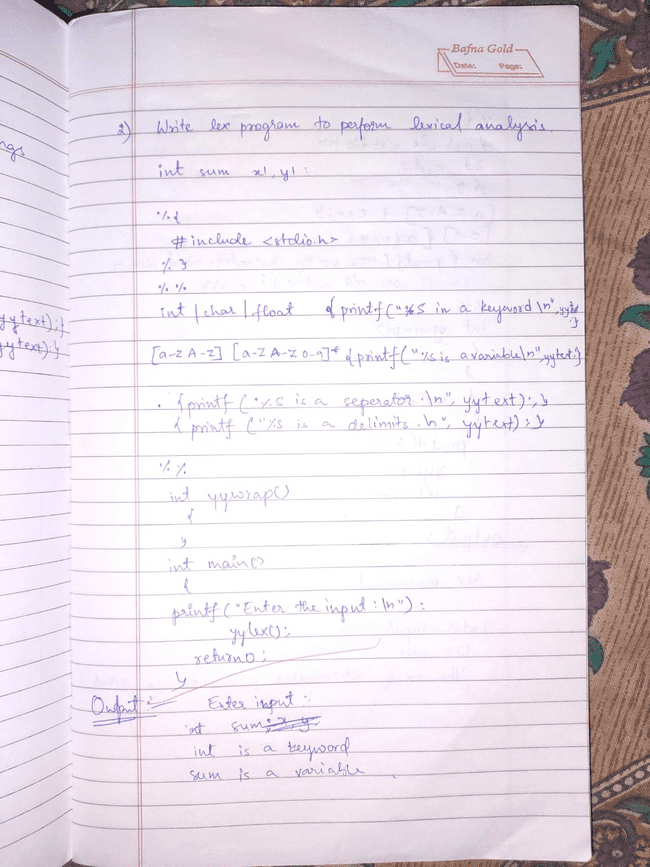
**Code**



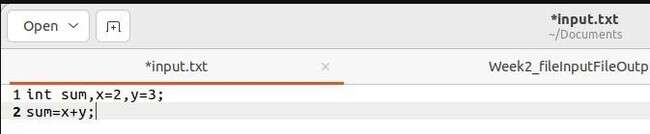
**Output**



**2.3 Write a program in lex to identify tokens in a program by taking input from a file and printing the output on the terminal.**

**Code** 

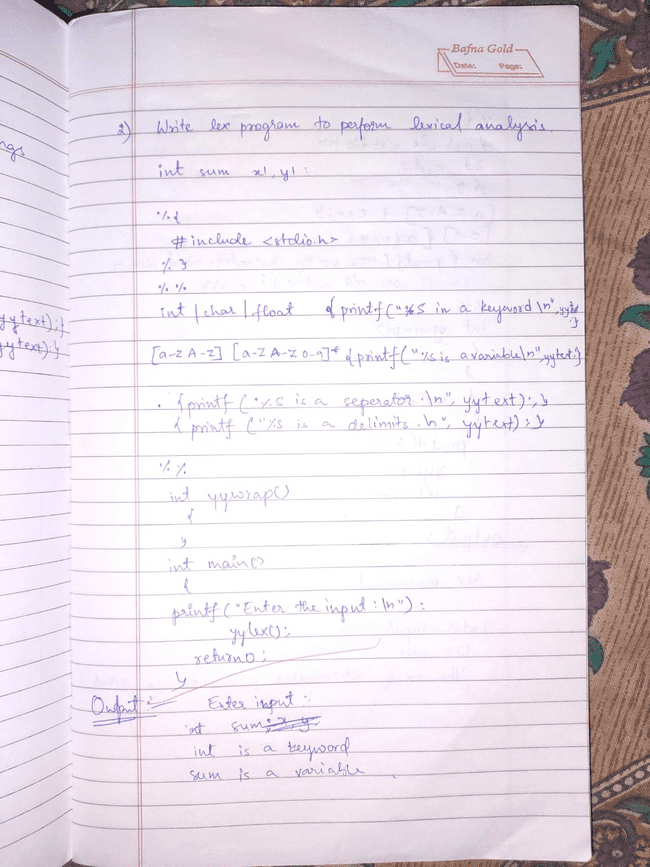
**Output**



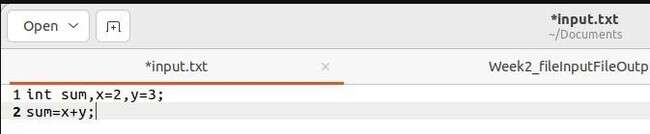


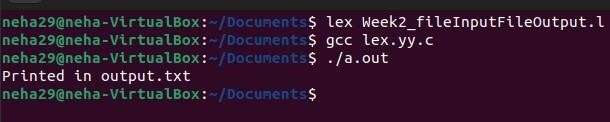
**2.4 Write a program in lex to identify tokens in a program by taking input from a file and printing the output in another file.**

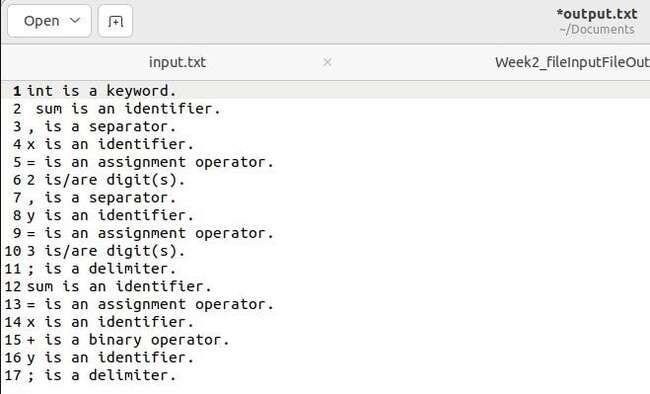
**Code**



**Output**



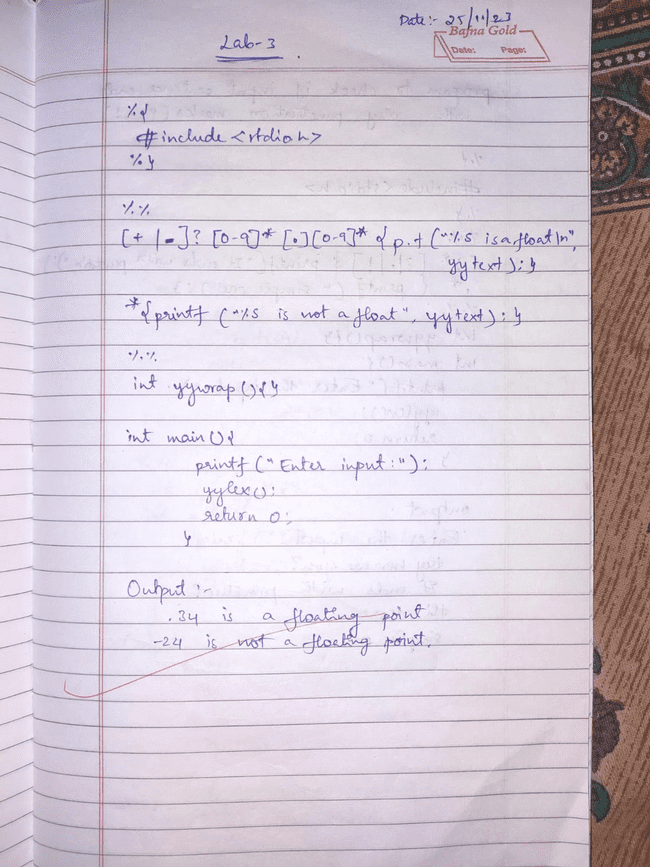




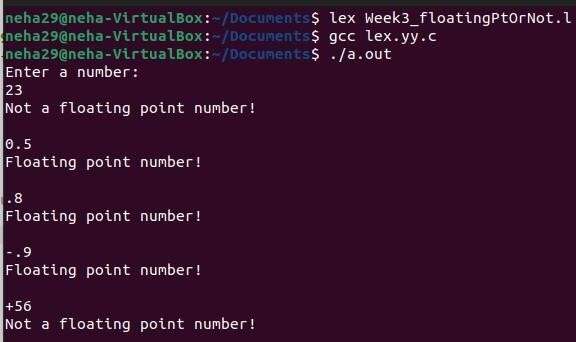
**Lab 3**

**3.1Write a program in LEX to recognize Floating Point Numbers.**

**Code**



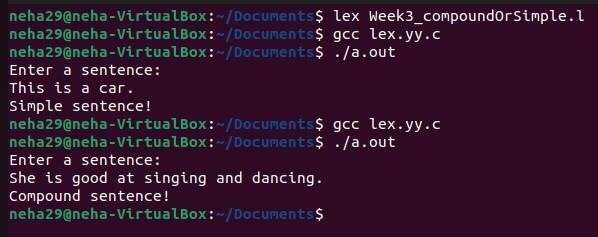
**Output**

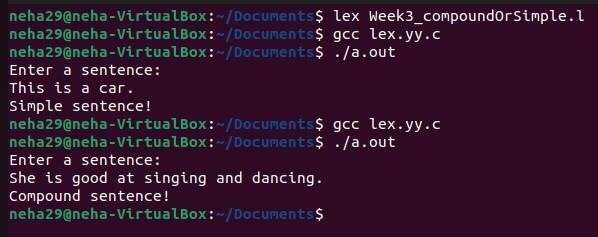


**3.2 Read and input sentence, and check if it is compound or simple. If a sentence has the word- and , or ,but ,because ,if ,then ,nevertheless then it is compound else it is simple.**



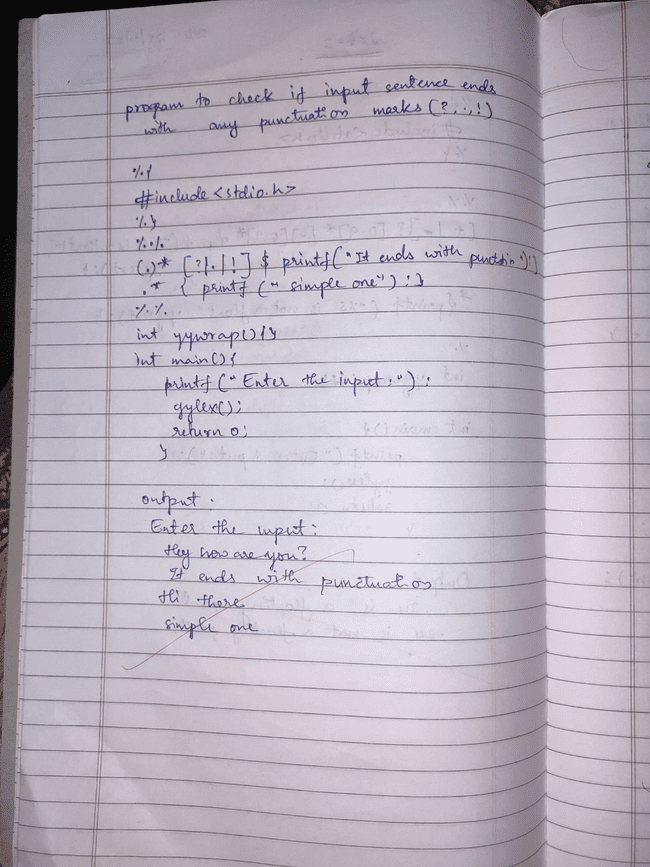
**Output**

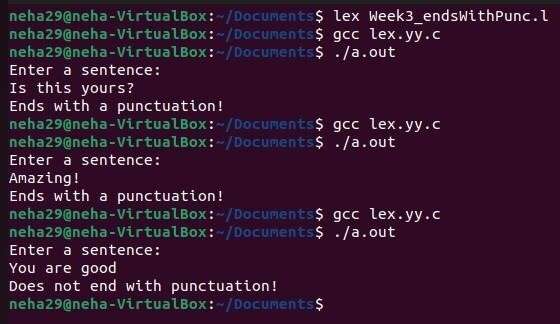


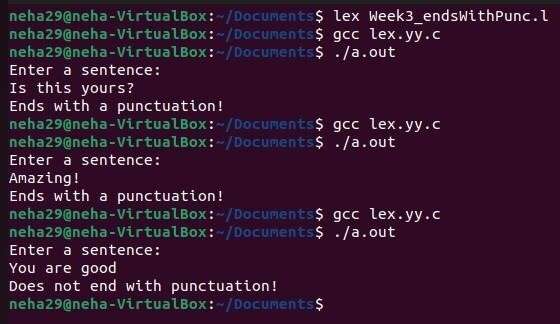


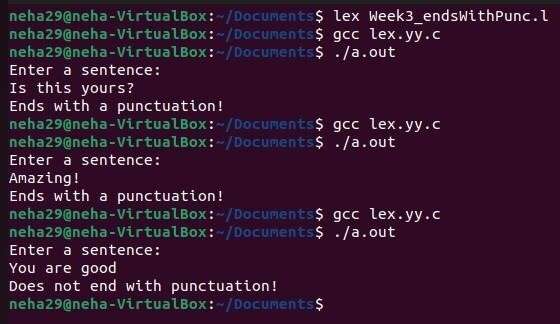
**3.3 Write a program to check if the input sentence ends with any of the following punctuation marks ( ? , fullstop , ! )**

**Code**

**Output**





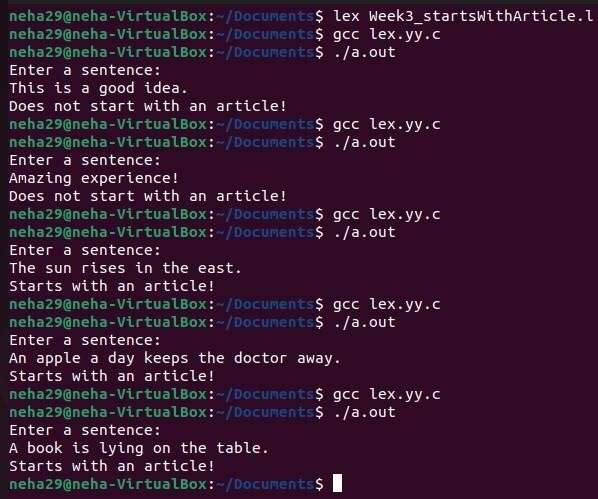
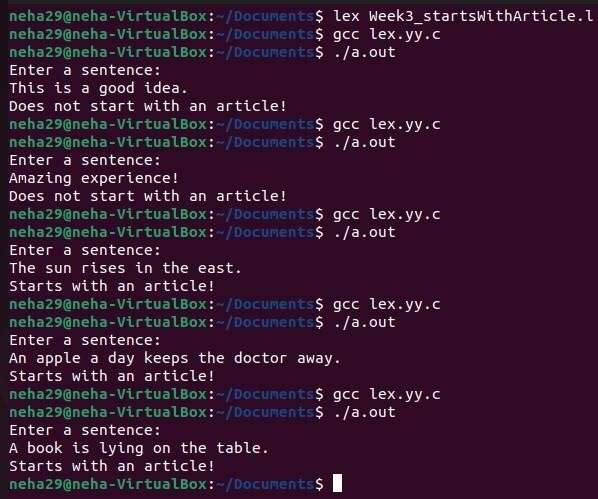
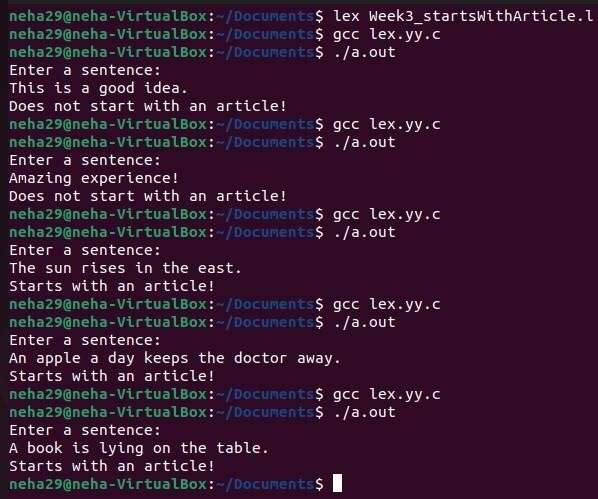
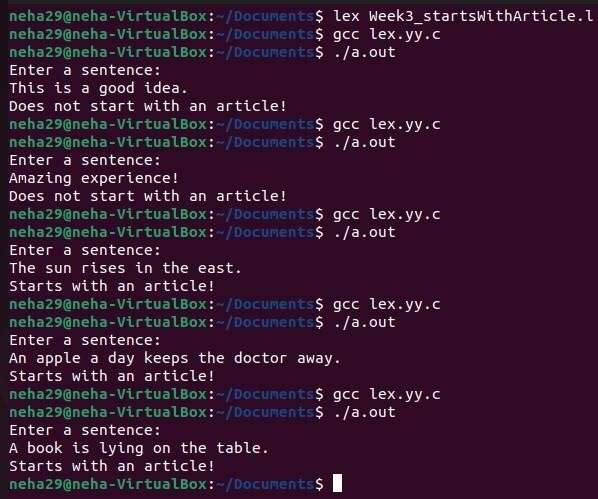
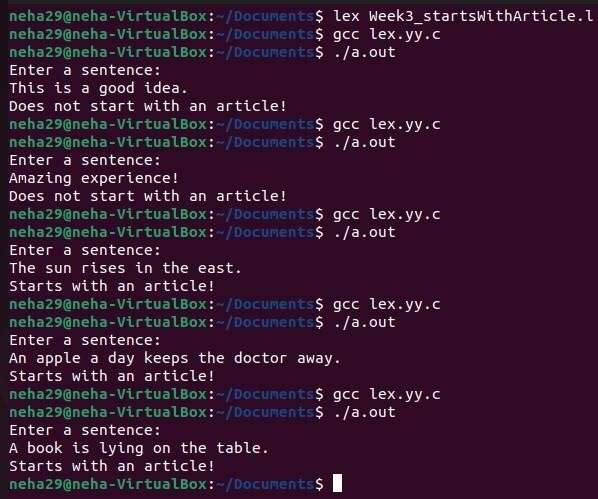


**3.4 Write a program to read an input sentence and to check if the sentence begins with English articles (A, a,AN,An,THE and The).**

**Code**

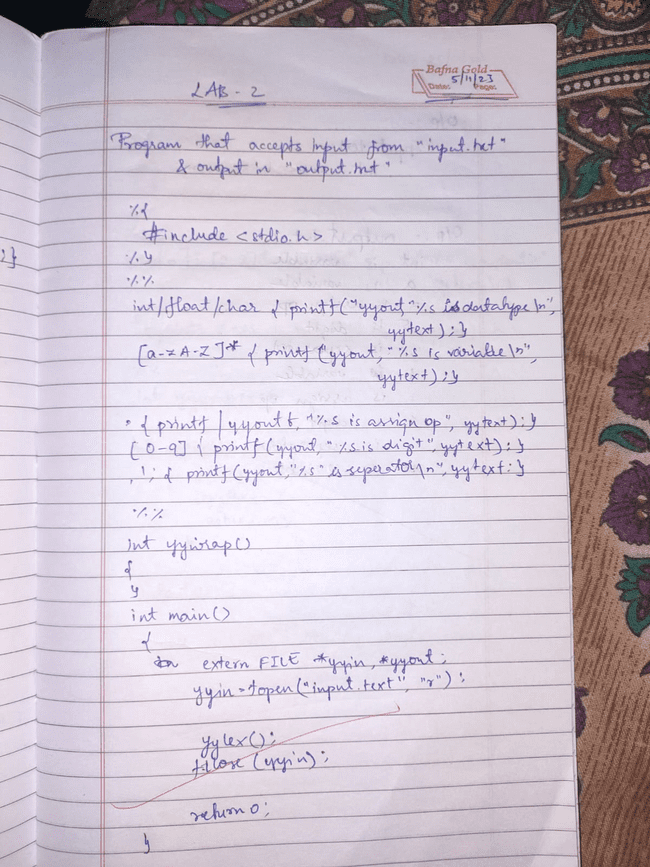


**Output**

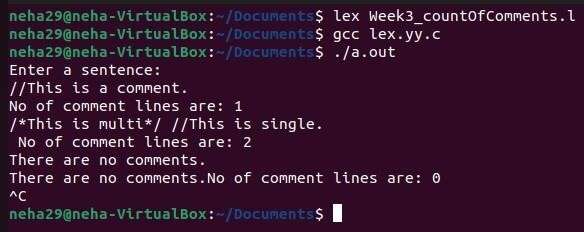


**3.5 Lex program to count the number of comment lines (multi line comments or single line) in a program. Read the input from a file called input.txt and print the count in a file called output.txt.**

**Code**



**Output**

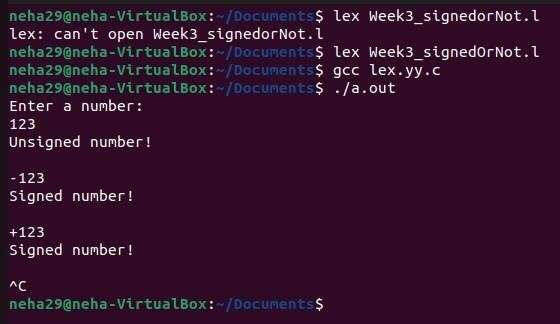


**3.6 Write a program to read and check if the user entered number is signed or unsigned using appropriate meta character.**

**Code**

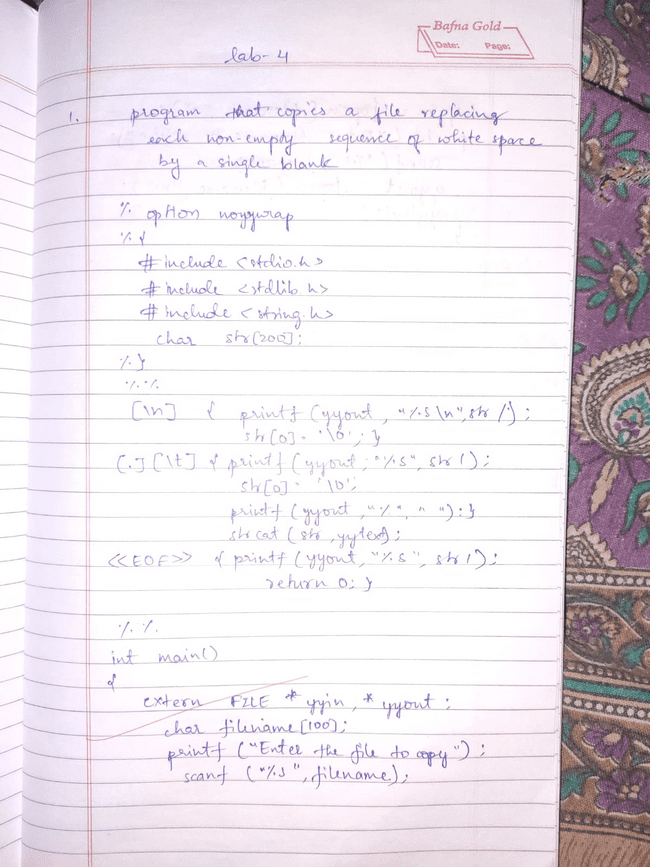


**Output**

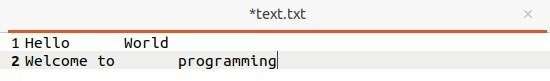


# Lab 4

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

**Code** 

**Output**

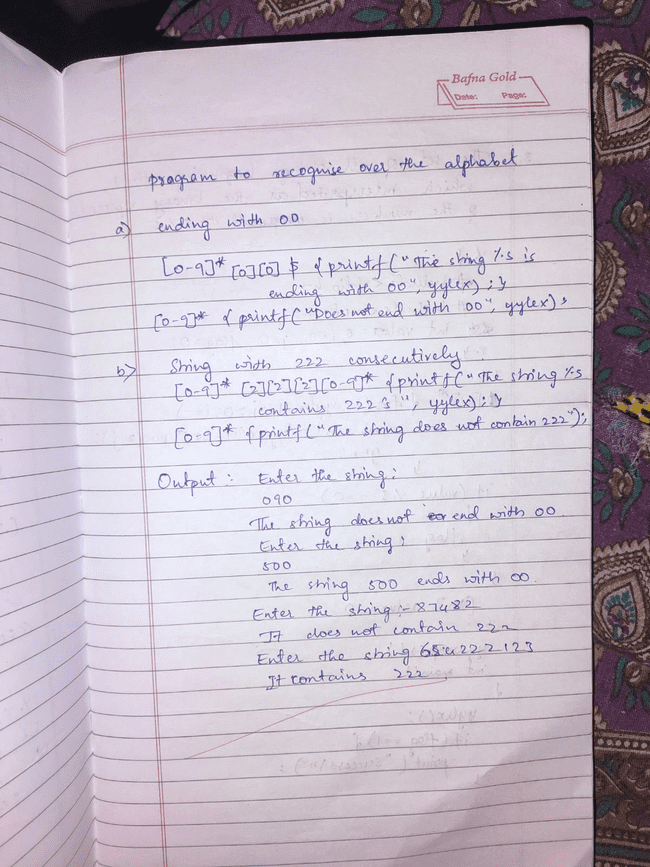




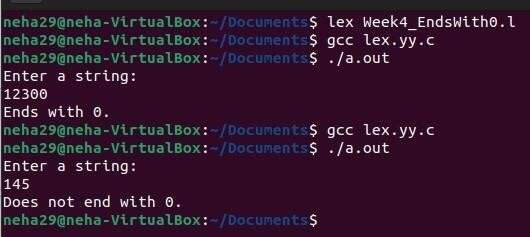


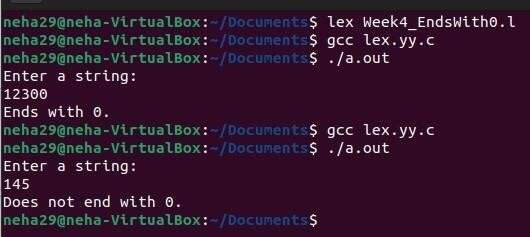
**4.2 Write a LEX program to recognize the following tokens over the alphabets {0,1,..,9}**

**4.2.1 The set of all string ending in 00.**

**Code** 

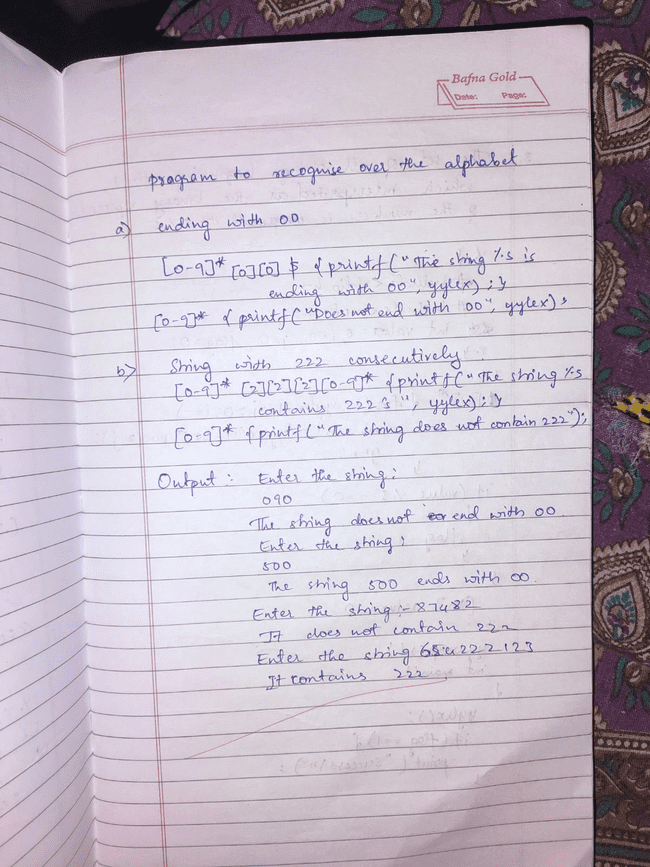
**Output**



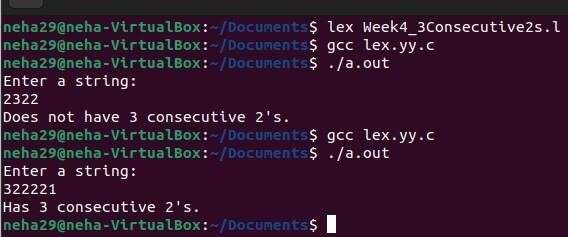


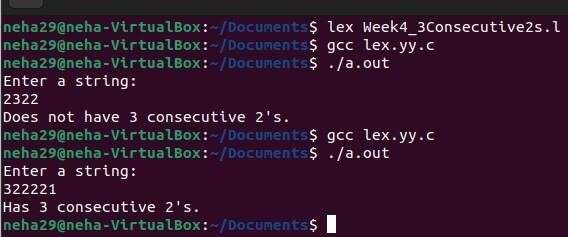
**4.2.2 The set of all strings with three consecutive 222’s.**

**Code**

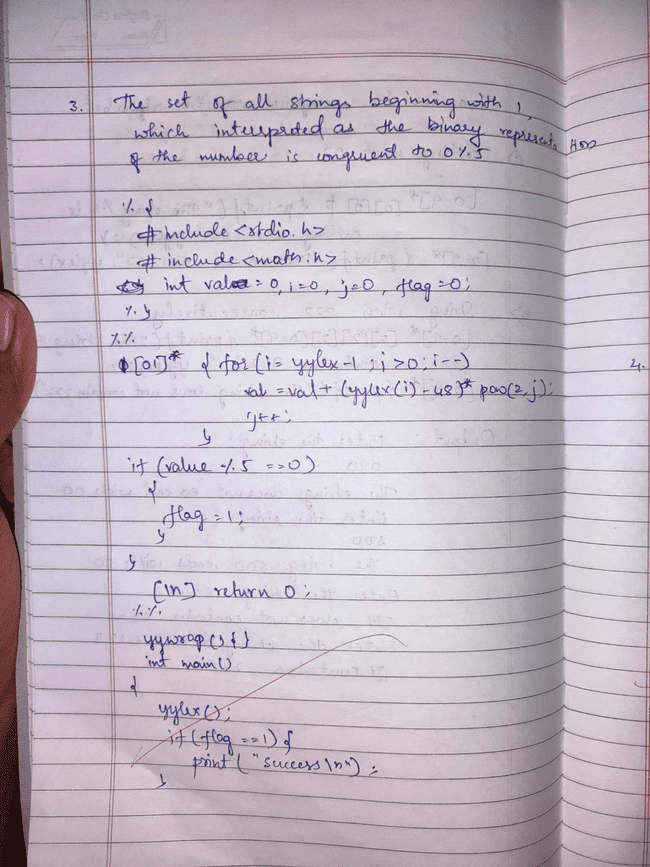


**Output**





**4.2.3 The set of all string such that every block of five consecutive symbols contains at least two 5’s.**

**Code** 

**Output**

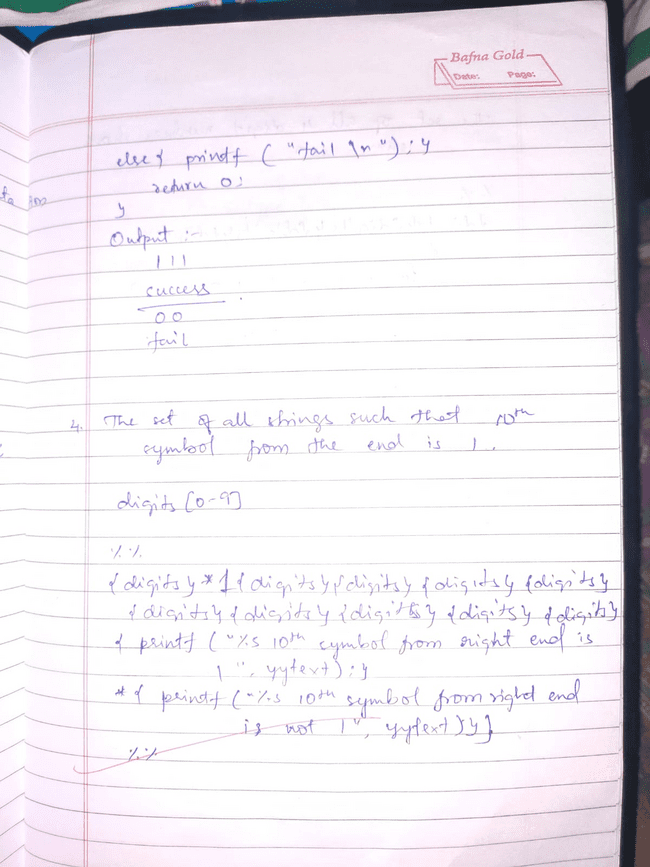


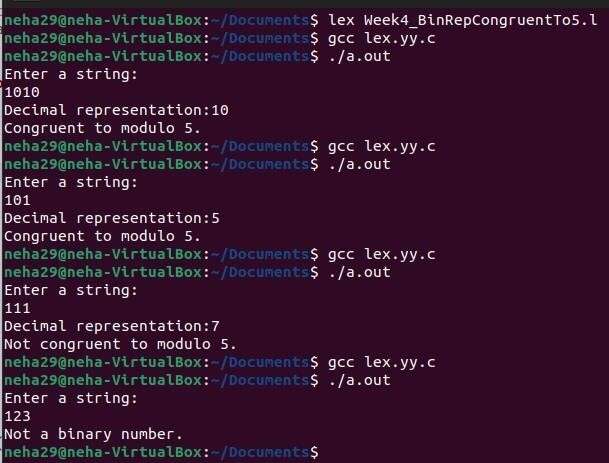


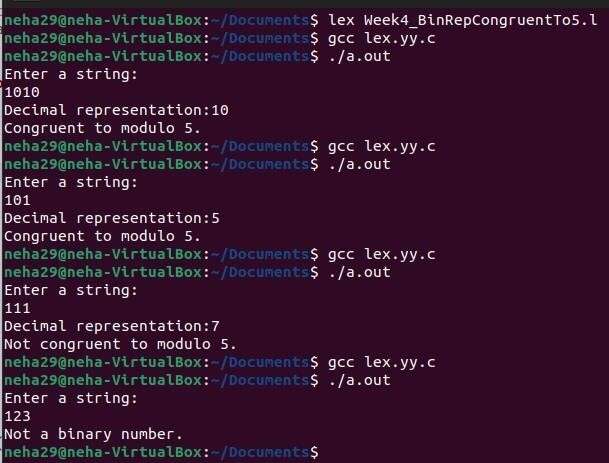
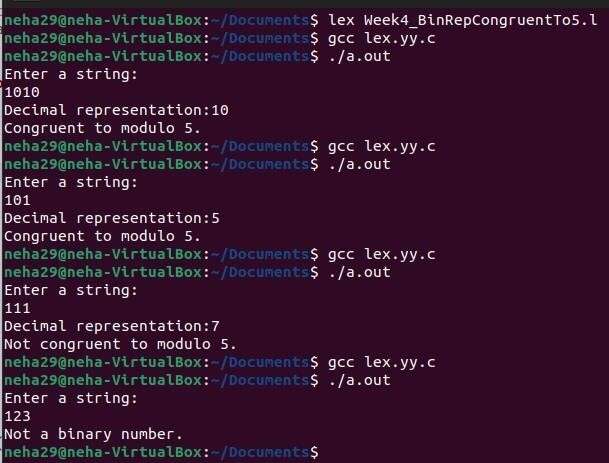
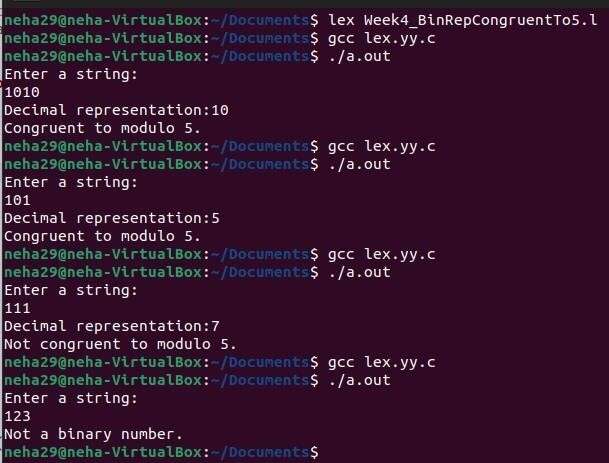


**4.2.4 The set of all strings beginning with a 1 which, interpreted as the binary representation of an integer, is congruent to zero modulo 5.**

**Code**

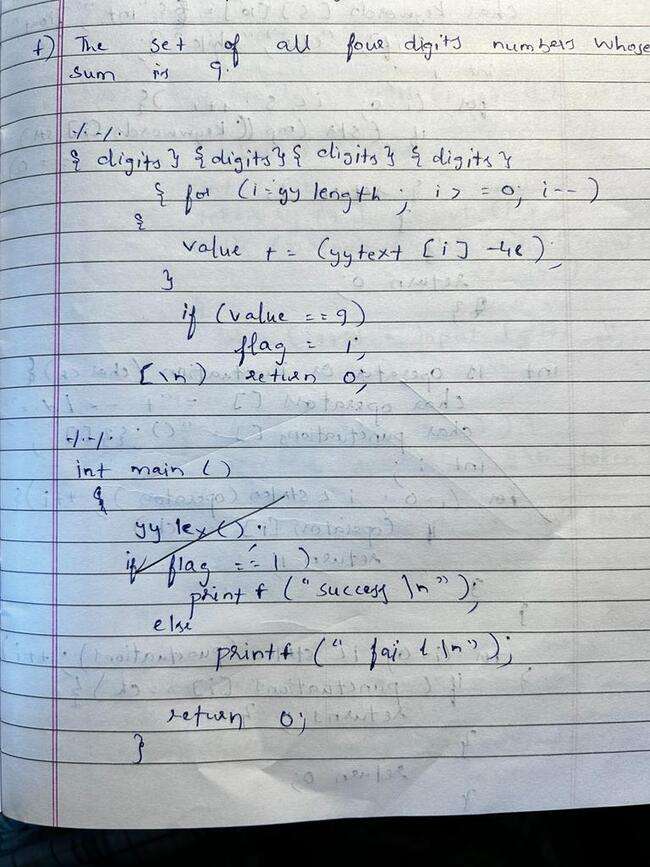


**Output**



**4.2.6 The set of all four digits numbers whose sum is 9.**

**Code**



**Output**

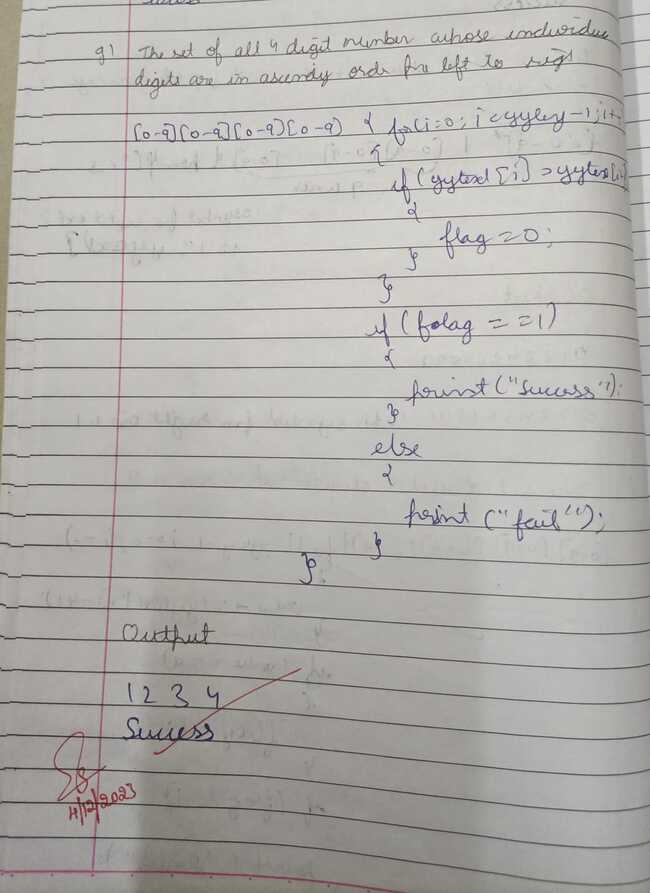




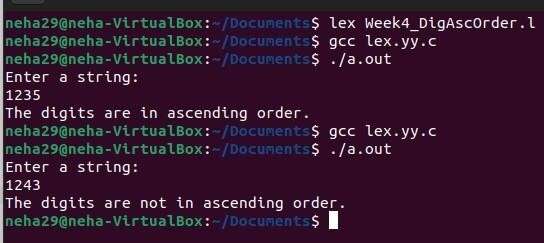


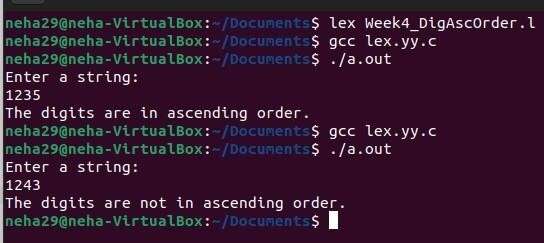
**4.2.7 The set of all four digital numbers, whose individual digits are in ascending order from left to right.**

**Code**



**Output**

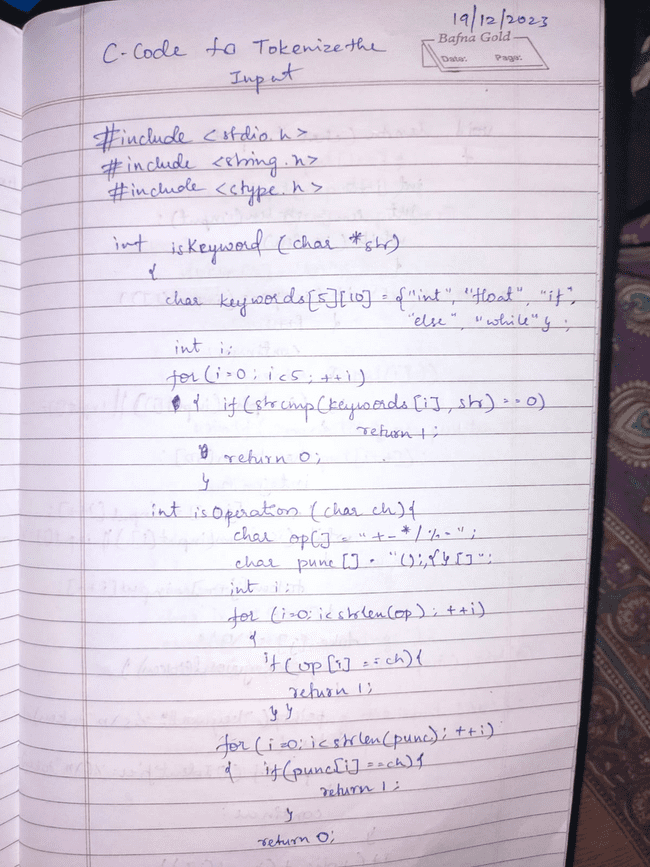


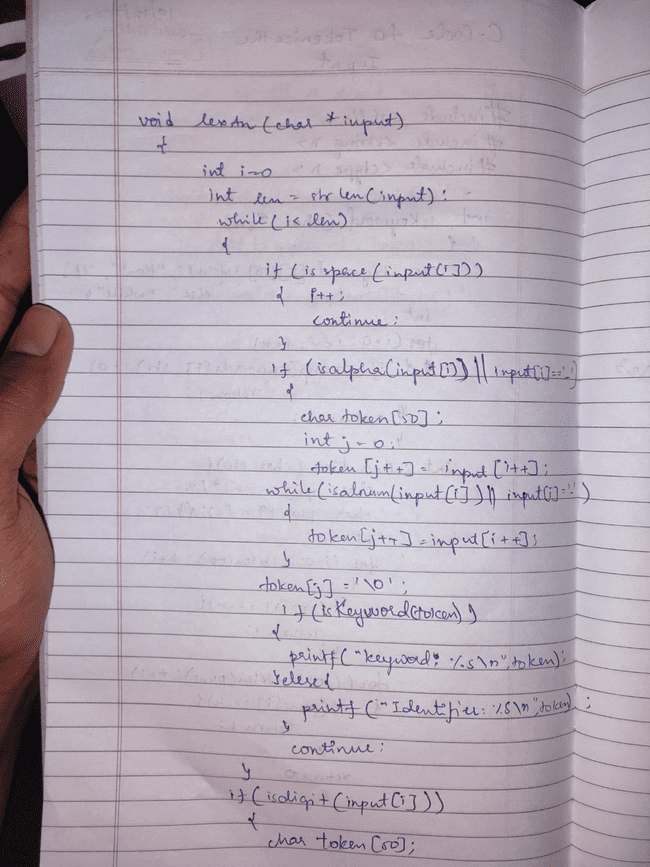


# Lab 5

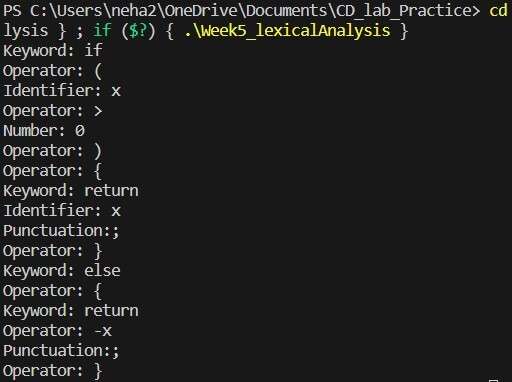
**Write a C program to design lexical analysis to recognize any five keywords, identifiers, numbers, operators and punctuations.**

**Code**





**Output**



# Lab 6

**Write a program to perform recursive descent parsing on the following grammar:**

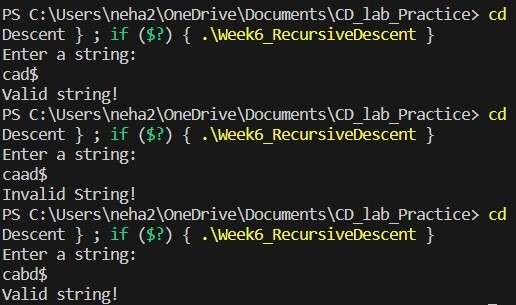
**S->cAd**

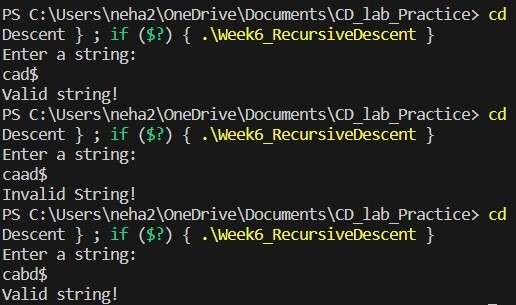
**A->ab | a**

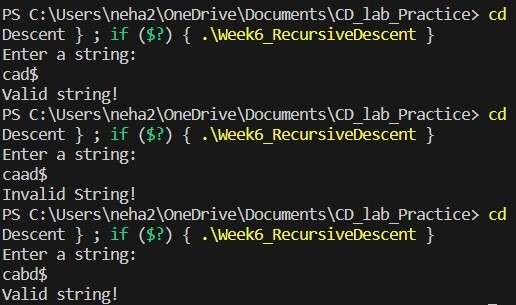
**Code**



**Output**



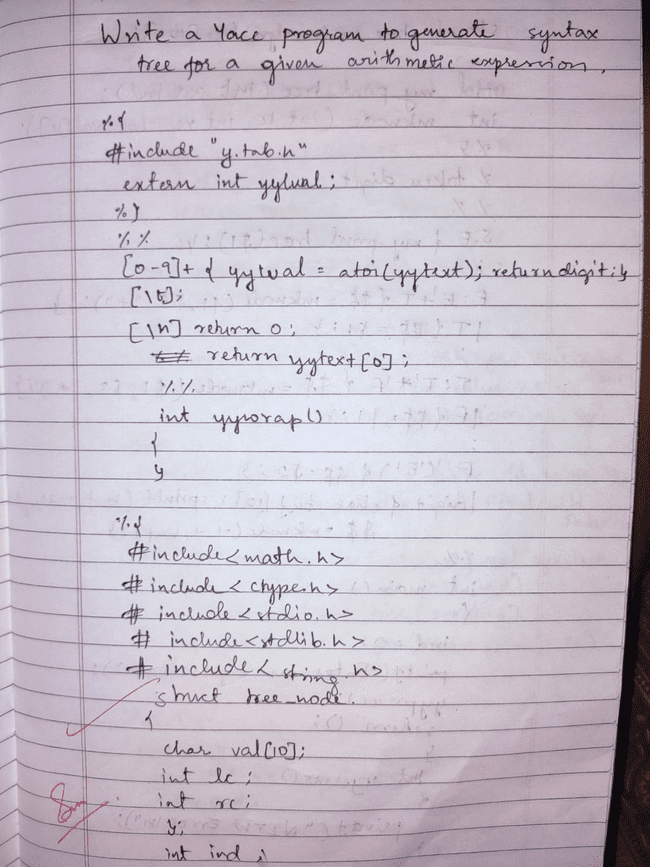




**Lab 7**

**7.1Write a program in YACC to design a suitable grammar for evaluation of arithmetic expression having +, -, \* and /.**

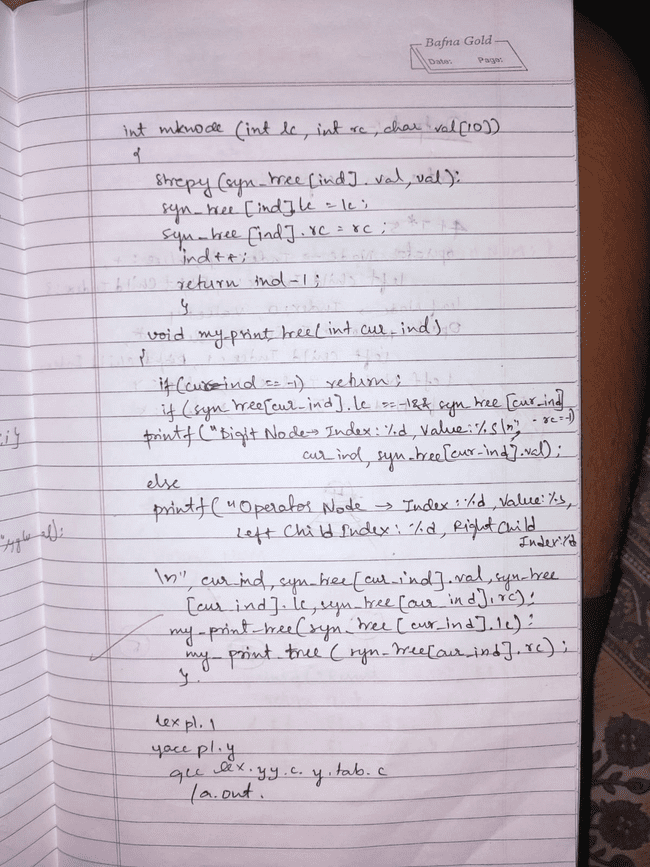
**Code**



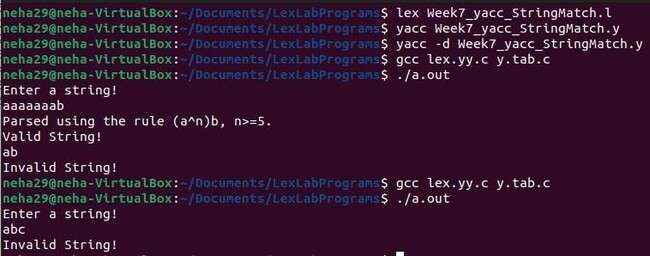
**Output**



**7.2 Write a program in YACC to recognize strings of the form {(a^n)b , n>=5}.**

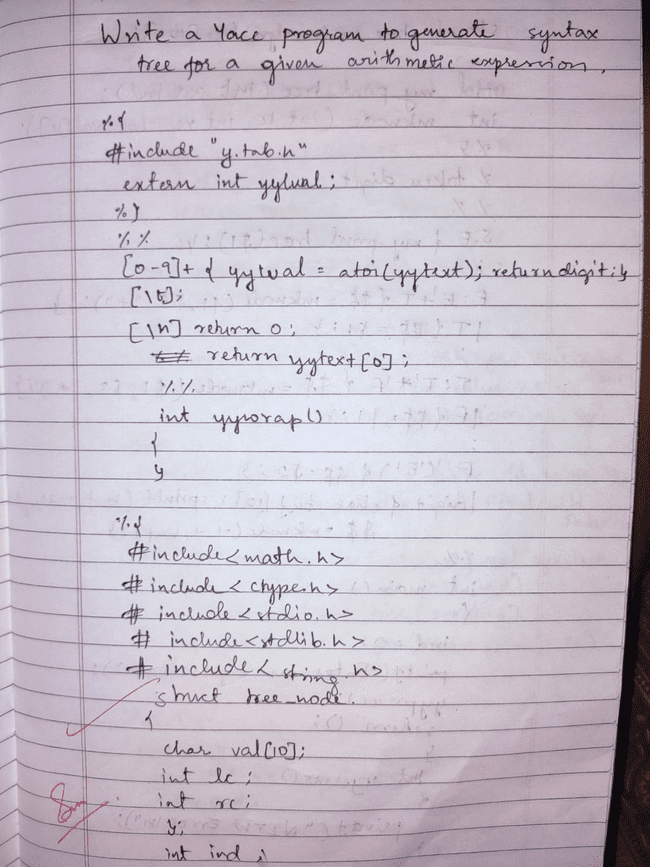
**Code** 

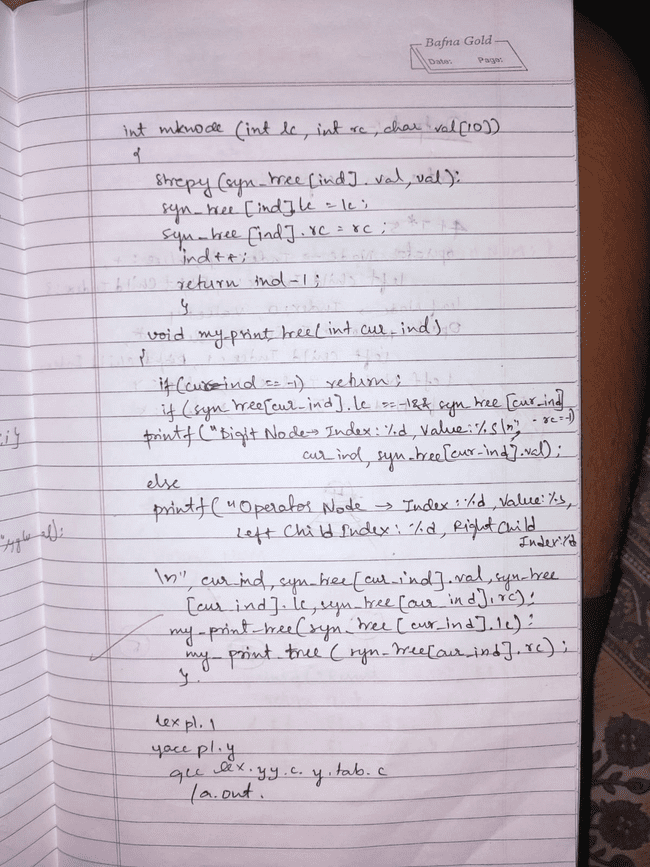
**Output**



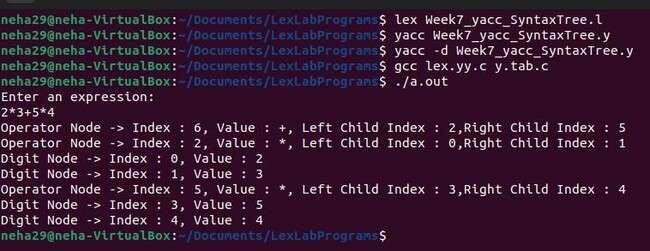
**7.3 Write a program in YACC to generate syntax tree for a given arithmetic expression.**

**Code**





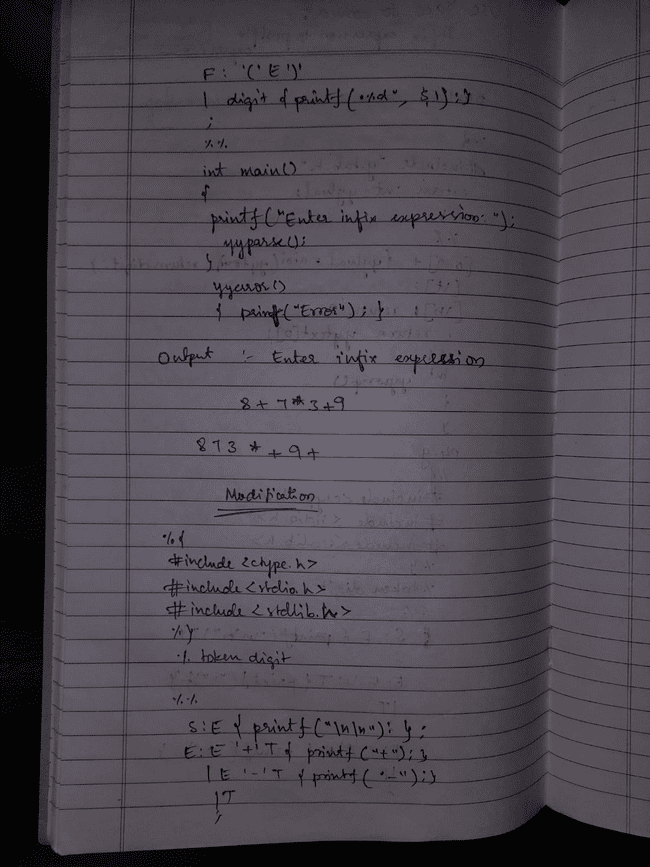
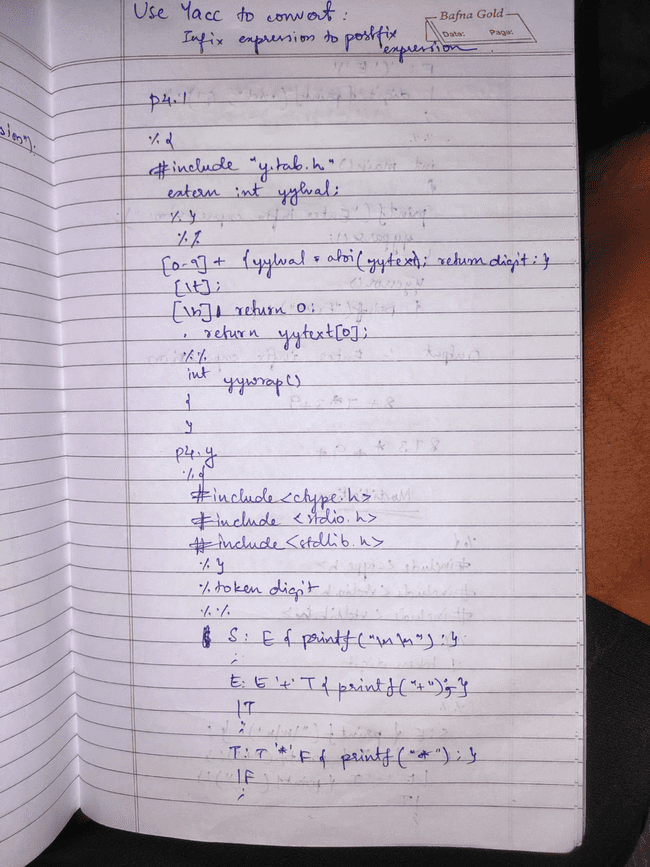
**Output**

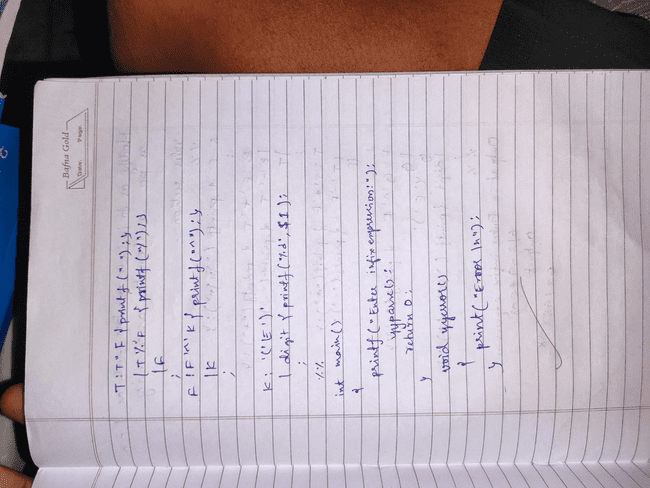


# Lab 8

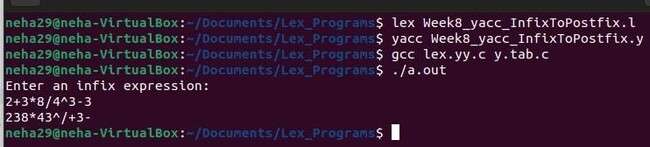
**8.1Write a program in YACC to convert infix to postfix expression.**

**Code**



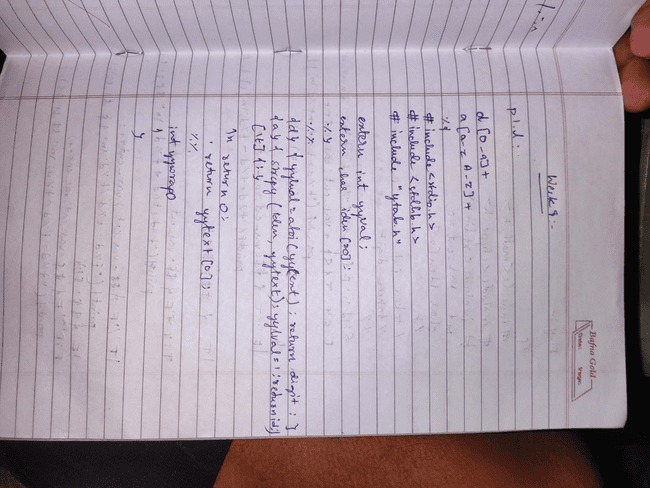


**Output**



**Lab 9**

**9.1Write a program in YACC to generate three address code for a given expression.**



**Output**

