Lexical Analysis

a) Program must be written in the lex programming language

b) .L extension (lex source format)

c) Mike Lesk & Eric Schmidt in 1975 invented it

d) It changes into transitional diagram called lex.yy.c (after generating)

e) Three phases of conversion

File → $\boxed{\text{Lex compiler}}$ → lex.yy.c

lex.yy.c → $\boxed{\text{c.compiler}}$ → a.out

Input → $\boxed{\text{a.out}}$ → stream of tokens

Lex - Out+YAcc

→ scans input string character by character

Format of the Transition diagram generated in three part sections

1) {declaration} rules separately

" %% "

2) {Transition Rules}

Translation

%%

3) {Auxilary form}

<u>Syntax</u>

```
%{ #include <stdio.h>
      int c;
  %}


        %%

  pattern(---rules--)

        %%

  main() → auxilary
```

<u>Functions</u>

1) <u>yywrap()</u>
- called when input is exhausted
- return 1 if input finished else 0.

2) <u>yylex()</u>
- read input string & generates tokens according to regular exp

3) <u>yytext</u>
- not a func
- pointer to input string

Example

```
% { #include<stdio.h>
   %}

        %%
"hi" printf("bye");
     %%
    main()
{   {printf("Entrainput");
      yylex();
    }
    int yywrap():
      { return1;
      }
```