1. LEX program to accept language (a + b)\* abb

%{

#include <stdio.h>

int flag=0;

%}

%%

[ab]\*abb { flag=1;}

. |

\n {;}

%%

main()

{

printf("Enter string :");

yylex();

if (flag)

printf("valid");

else

printf("invalid -------");

return 0;

}

1. LEX program to accept language at least one ‘a’ on ∑={a,b}

%{

#include <stdio.h>

int flag=0;

%}

%%

[ \t]+ {;}

[ab]\*[a][ab]\* { flag =1;printf("valid");}

. |

\n {;}

%%

main()

{

printf("Enter string :");

yylex();

if (flag)

printf("valid");

else

printf("invalid -------");

return 0;

}

1. LEX program to accept language exactly one ‘a’ on ∑={a,b}

%{

#include <stdio.h>

int flag=0;

%}

%%

[ \t]+ {;}

[ab]\*[a][ab]\* { flag =1;printf("valid");}

. |

\n {;}

%%

main()

{

printf("Enter string :");

yylex();

if (flag)

printf("valid");

else

printf("invalid -------");

return 0;

}

1. LEX program to validate integer number

%{

#include <stdio.h>

int flag =0;

%}

%%

[ \t]+ { ; }

[0-9]+ { flag =1; return 0; }

. |

\n {;}

%%

main()

{

printf(" Enter number : ");

printf("\n before flag - %d ",flag);

yylex();

printf(" flag -%d",flag);

if (flag)

printf("valid");

else

printf("Invalid");

}

yyerror()

{

printf("\n invalid ");

}

1. LEX programs for to count identifiers, Constants and operators in an arithmetic expression

%{

#include <stdio.h>

int id=0, con=0, op=0;

%}

%%

[ \t]+ ;

[\+\-\\*\/] { op++;}

[0-9]+ { con++;}

[a-zA-z][a-zA-z0-9]\* { id++;}

. | \n ;

%%

main()

{

printf("Enter expression : ");

yylex();

printf("Total Number of operators : %d \n",op);

printf("Total Number of Identifiers : %d \n",id);

printf("Total Number of constants : %d \n",con);

}

1. LEX program to validate integer and decimal number.

%{

#include <stdio.h>

%}

%%

[ \t]+ ;

\+?[0-9] { printf("\n %s is an integer",yytext);}

\+?[0-9]+\.[0-9]+ { printf("\n %s is float",yytext);}

. | \n ;

%%

main()

{

printf("\n Enter Text : ");

yylex();

return;

}

1. LEX program to validate and count integer and decimal number.

%{

#include <stdio.h>

int i=0, d=0;

%}

%%

\+?[0-9]+ { printf("\n %s is an integer",yytext);i++;}

\+?[0-9]+\.[0-9]+ { printf("\n %s is a Decimal ",yytext);d++;}

%%

main()

{

printf("Enter numbers : ");

yylex();

printf("Total Int Numbers : %d \n",i);

printf("Total Decimal Numbers : %d ",d);

}

1. LEX program to count positive and Negative numbers.

%{

#include <stdio.h>

int p=0, n=0;

%}

%%

\+?[0-9]+ { p++;}

\-[0-9]+ { n++;}

%%

main()

{

printf("Enter numbers : ");

yylex();

printf("Positive Numbers : %d \n Negative Numbers : %d ",p,n);

}

1. LEX program to validate integer number

%{

#include <stdio.h>

int flag=0;

%}

num [0-9]

%%

{num}+ { flag =1;}

[^0-9]+ { flag =0;}

[ \t]+ ;

%%

yyerror()

{

printf(" Invalid \n");

}

int main()

{

printf("Enter a Number : ");

yylex();

if(flag==1)

printf("valid\n");

else

printf("INVALID\n");

return 0;

}

1. LEX program for check sentence is Simple or Compound

%{

#include <stdio.h>

int comp=0;

%}

%%

and |

but |

or |

neither |

either {comp=1;}

[a-zA-Z]+ ;

%%

main()

{

printf("Enter a Sentence : ");

yylex();

if(comp)

printf("\n Sentence is Compound statement\n",yytext);

else

printf("\n Sentence is not a Compound statement\n",yytext);

}

1. LEX program to identify verbs in a TEXT

%{

#include <stdio.h>

%}

%%

[ \t]+ ;

[a-zA-z]+ { printf("\n %s is not a verb",yytext);}

is |

was |

am |

are |

were |

being |

do |

did |

does { printf("\n %s is verb ",yytext);}

. | \n ;

%%

main()

{

printf("\n Enter Text : ");

yylex();

}

1. LEX program to count words, characters and lines in a TEXT

%{

#include <stdio.h>

int wc=0, lc=0,ch=0;

%}

%%

[^ \t\n]+ {wc++; ch+=yyleng;}

\n {lc++;}

. {ch++;}

%%

main()

{

printf("\n Enter a text : " );

yylex();

printf("\nNumber of words %d :",wc);

printf("\nNumber of lines %d :",lc);

printf("\nNumber of words %d :",ch);

}