1.files.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

FILE \*fp;

char ch,s[100];char \*c;int sum=0;

int i=0,j,k=0,z=0,l,f=0,h,nl=0,nw=0;

float real[100],sumf=0;

char arr[100],farr[100],ex[10];int integer[100];

fp=fopen("one.txt","r");

strcpy(ex,"95.456r");

while((ch=getc(fp))!=EOF)

{

s[i]=ch;

i++;

}

for(j=0;j<=i;j++)

printf("%c",s[j]);

printf("no of charecters: %d\n",j);

for(j=0;j<=i;j++)

{

if(s[j]=='.')

{

for(l=0;l<=k;l++)

arr[l]=0;

k=0;

while(s[j+1]!='\n'&&s[j+1]!=' '&&s[j+1]!='\n')

j++;

continue;

}

if(isdigit(s[j]) && (!isalpha(s[j-1]))) // for integers

{

arr[k]=s[j];

k++;

if(s[j+1]=='\n' ||s[j+1]==' ' || s[j+1]=='\0')

{

integer[z]=atoi(arr);

z++;

for(l=0;l<=k;l++)

arr[l]=0;

k=0;

}

}

}

k=0;

for(j=0;j<=i;j++) //for float

{

if(isdigit(s[j]) || (s[j]=='.' &&(isdigit(s[j-1]) && isdigit(s[j+1]))))

{

farr[k]=s[j];

k++;

if(s[j+1]=='\n' ||s[j+1]==' ' || s[j+1]=='\0')

{

for(h=0;h<k;h++)

{

// printf(" inside %f",real[f]);

if(farr[h]=='.')

{

real[f]=atof(farr);

f++;

}

}

for(l=0;l<=k;l++)

farr[l]=0;

k=0;

}

}

}

for(j=0;j<i;j++) //no of lines and words

{

if(s[j]=='\n'){nl++;}

if(s[j]=='\n' || s[j]==' '){nw++; }

}

printf("\nNo of lines:%d",nl);

printf("\nNo of words:%d\n",nw-(nl+z+f));

for(j=0;j<z;j++) //to print integers

{

printf("integer[%d] :%d \n",j,integer[j]);

}

printf("Total no of integers: %d\n",z);

for(j=0;j<f;j++) //to print float

{

printf("float[%d] :%f \n",j,real[j]);

}

printf("\nTotal no of float values:%d\n",f);

for(i=0;i<z;i++) //sum of integers

sum=sum+integer[i];

printf("sum of integers:%d\n",sum);

for(i=0;i<f;i++) //sum of floats

sumf=sumf+real[i];

printf("sum of float:%.4f\n",sumf);

}

1. Test1.c

#include<stdio.h>

#include<stdlib.h>

#include<stdbool.h>

#include<math.h>

int l,w,i,f,ch;

bool isInt(char \*,int \*);

bool isFloat(char \*,float \*);

int main()

{

FILE \*fp;

l=w=i=f=ch=0;

char c;

int sumi=0;

float sumf=0.0;

fp=fopen("data.txt","r");char buf[100];int p=0;

if(fp==NULL)

exit(1);

else

{

while((c=fgetc(fp))!=EOF)

{

if(c!=' '&&c!='\n'&&c!='\r')

{

buf[p++]=c;

ch++;

}

else // if(c==' '||c=='\n'||c=='\r')

{

buf[p]='\0';

//printf("%s\n",buf);

if(c=='\n')//||c=='\r')

l++;

int getint;float getfloat;

if(isInt(buf,&getint))

{

printf("integer:%s\n",buf);

// i++;

sumi+=getint;

}

else if(isFloat(buf,&getfloat))

{

printf("float:%s\n",buf);

// f++;

sumf+=getfloat;

}

else

{

printf("word:%s\n",buf);

w++;

}

buf[p]='\0';

p=0;

}

// printf("%d %d %d %d %s",i,l,f,w,buf);

}

}

printf("sum if ints %d\n",sumi);

printf("sum of floats %f\n",sumf);

printf("no of words %d\n",w);

printf("no of lines %d\n",l);

printf("no of integers %d\n",i);

printf("no of floats %d\n",f);

printf("no of characters %d\n",ch);

}

bool isInt(char \*s,int \*int\_val)

{

// printf("hi %d\n",\*getint);

\*int\_val=0;

int sign=1;

int p;

if(s[0]=='-')

sign=-1;

if(s[0]=='+'||s[0]=='-'||(s[0]>=48&&s[0]<=57))

{

if(s[0]>=48&&s[0]<=57)

\*int\_val=(int)(s[0]-48);

}

else

return false;

for(p=1;s[p]!='\0';p++)

{

if(s[p]>=48&&s[p]<=57)

{

\*int\_val=(\*int\_val)\*10+(int)(s[p]-48);

}

else

return false;

}

\*int\_val=(\*int\_val)\*sign;

i++;

return true;

}

bool isFloat(char \*s,float \*float\_val)

{

//printf("hi 3 : %s",s);

float sign=1.0;

\*float\_val=0.0;

int i;int count=0;

if(s[0]=='-')

sign=-1.0;

if(s[0]=='+'||s[0]=='-'||(s[0]>=48&&s[0]<=57))

{

if(s[0]>=48&&s[0]<=57)

\*float\_val=(\*float\_val)\*10.0+(float)(s[0]-48);

}

else

return false;

int flag=0;int power1=-1;

//printf("hi 4:%f",\*getfloat);

for(i=1;s[i]!='\0';i++)

{

if(s[i]=='.'&&count==0)

{

count++;flag=1;

continue;

}

else if(s[i]=='.'&&count>0)

return false;

if((s[i]>=48&&s[i]<=57)&&flag==0)

{

\*float\_val=(\*float\_val)\*10+(s[i]-48);

}

else if((s[i]>=48&&s[i]<=57)&&flag==1)

{

\*float\_val=(\*float\_val)+(s[i]-48)\*(pow(10,power1));

power1--;

}

else

return false;

}

\*float\_val=(\*float\_val)\*sign;

f++;

return true;

}

2.Scanner.c

#include<stdio.h>

#include<string.h>

int main()

{

FILE \*fp;

char ch,c;

int i=0,j,flag=0,to=0;

char s[100][100]={"do","double","else","enum","extern","float","for","goto","if", "int","long","register","return","short","signed","sizeof","static","struct","switch","typedef","union","unsigned","void","volatile","while","continue","break","case","char","const","default","auto"};

char st[10];

fp=fopen("f.txt","r");

printf("Token no\t\tToken\t\tLexame \n");

while((ch=getc(fp))!=EOF)

{

i=0;

flag=0;

if(ch>=42&&ch<=47)

{

to++;

printf("%d\t\toperator\t\t%c\n",to,ch);

}

if((ch>=32&&ch<=41)||(ch>=91&&ch<=96)||(ch>=58&&ch<=63)||(ch>=123&&ch<=126))

{

to++;

printf("%d\t\tspecial character\t\t%c\n",to,ch);

}

if(isdigit(ch))

{

to++;

printf("%d\t\tdigit\t\t\t%c\n",to,ch);

}

if(isalpha(ch))

{

st[i]=ch;

i++;

ch=getc(fp);

while(isalnum(ch)&&ch!=' ')

{

st[i]=ch;

i++;

ch=getc(fp);

}

st[i]='\0';

for(j=0;j<32;j++)

{

if(strcmp(st,s[j])==0)

{

flag=1;

break;

}

}

to++;

if(flag==1)

printf("%d\t\tkeyword\t\t\t%s \n",to,st);

else

printf("%d\t\tidentifier\t\t%s\n",to,st);

if(ch=='.')

{

i=0; int k=1;

if(isalpha(c=getc(fp)))

{

st[i]=c;

i++;

c=getc(fp);

while(isalnum(c)&&c!=' ')

{

st[i]=c;

i++;

k++;

c=getc(fp);

}

st[i]='\0';

to++;

printf("%d\t\tExtension\t\t.%s\n",to,st);

ch=c;

}

else

{

if(ch>=42&&ch<=47)

{

to++;

printf("%\t\toperator\t\t%c\n",to,ch);

}

if((c>=32&&c<=41)||(c>=91&&c<=96)||(c>=58&&c<=63)||(c>=123&&c<=126))

{

to++;

printf("%d\t\tspecial character\t\t%c\n",to,c);

}

}

}

if((ch>=32&&ch<=41)||(ch>=91&&ch<=96)||(ch>=58&&ch<=63)||(ch>=123&&ch<=126))

{

to++;

printf("%d\t\tspecial character\t\t%c\n",to,ch);

}

}

}

}

3 Lex scanner

%{

#include<stdio.h>

#include<string.h>

int t=1;

int l=1;

%}

key auto|break|case|char|const|continue|default|do|double|else|enum|extern|float|for|goto|if|int|long|register|return|short|signed|sizeof|static|struct|switch|typedef|union|unsigned|void|volatile|while

file txt|h|c

preprocc include|define

%%

[#]{preprocc} {printf("\n%d\tpre process directive\t%s",t,yytext);t++;}

[#$&^{}()"'] {printf("\n%d\tspecial char\t%s",t,yytext);t++;}

{key} {printf("\n%d\tkeyword\t\t%s",t,yytext);t++;}

[a-z]+"."{file} {printf("\n%d\tfile\t\t%s",t,yytext);t++;}

[a-z]+"."[a-z]+ {printf("\n%d\tidentifier\t%s",t,yytext);t++;}

"++" {printf("\n%d\tincrementer\t%s",t,yytext);t++;}

"--" {printf("\n%d\tdecrementer\t%s",t,yytext);t++;}

"/\*"|"//" {printf("\n%d\tcomment begins\t%s",t,yytext);t++;}

"\*/" {printf("\n%d\tcomment ends\t%s",t,yytext);t++;}

[=+-/%<>] {printf("\n%d\toperator\t%s",t,yytext);t++;}

"\\"[btvrfn?0a] {printf("\n%d\tescape char\t%s",t,yytext);t++;}

[+-]?[0-9]+ {printf("\n%d\tdigit\t\t%s",t,yytext);t++;}

[a-zA-Z][a-zA-Z0-9\_]\* {printf("\n%d\tidentifier\t%s",t,yytext);t++;}

[;] {printf("\n%d\tterminator\t%s",t,yytext);t++;}

[\n] {l++;}

%%

int main()

{

yyin=fopen("f.txt","r");

yylex();

}

4.octal hex

%{

#include<stdio.h>

%}

Oct [+-]?[0][0-7]+

Hex [+-]?[0][xX][0-9a-fA-F]+

Dec [+-]?[0-9]+

invalid [+|-]?[0]?[0-9a-zA-Z]+

%%

[0] printf("decimal");

{Hex} printf("this is a hexadecimal number");

{Oct} printf("this is an octal number");

{Dec} printf("this is decimal number");

[0-9]+[\+\-\\*\/]\* printf("this invalid");

{invalid} printf("this is invalid");

%%

int main()

{

printf("ENTER A NUMBER\n");

yylex();

}

int yywrap()

{

return 1;

}

5.capital

%{

#include<stdio.h>

#include<ctype.h>

#include<stdlib.h>

void display(char \*);

%}

letter [a-z]

letter2 [A-Z]

%%

{letter} { display1(yytext);}

{letter2} {display2(yytext);}

[^/n] {printf("");}

%%

int main()

{

yylex();

}

void display1(char \*s)

{

int i;

for(i=0;s[i]!='\0';i++)

printf("%c", toupper(s[i]));

}

void display2(char \*s)

{

int i;

for(i=0;s[i]!='\0';i++)

printf("%c", tolower(s[i]));

}

int yywrap()

{

return 1;

}

6. Real precison

%{

#include<stdio.h>

#include<string.h>

int f,i,j;

%}

%%

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

[+-]?[0-9]\*[.][0-9]+ {f=0; for(i=0;i<yyleng;i++)

if(yytext[i]=='.')

{ j=i+1; break;}

for(;j<yyleng;j++)

f++;

printf("\n%s is a floating number of precision %d\n",yytext,f);}

[0-9a-zA-Z]+[.][0-9+-.a-zA-Z]+ {printf("\ninvalid!!!");}

[\n] {return 0;}

%%

int main()

{

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

yylex();

}

int yywrap()

{

return 1;

}

7. vowel

%{

#include<stdio.h>

int vowel=0;

int cons=0;

%}

%%

"a"|"e"|"i"|"o"|"u"|"A"|"E"|"I"|"O"|"U" {printf("%s vowel\n",yytext);vowel++;}

[a-zA-Z] {printf("%s consonant\n",yytext);cons++;}

[\n] { return 0;}

%%

int yywrap()

{

return 1;

}

int main()

{

printf("Enter a string\n");

yylex();

printf("\nvowel=%d and consonant=%d\n",vowel,cons);

return 0;

}

8.nposnneg

%{

#include<stdio.h>

int pos\_int\_count=0;

int neg\_int\_count=0;

int pos\_float\_count=0;

int neg\_float\_count=0;

%}

%%

[+]?[0-9]+ {/\*printf("+ve Integer");\*/pos\_int\_count++;}

[-][0-9]+ {/\*printf("-ve Integer");\*/neg\_int\_count++;}

[+]?[0-9]+[.][0-9]+ {/\*printf("+ve Float");\*/pos\_float\_count++;}

[-][0-9]+[.][0-9]+ {/\*printf("-ve Float");\*/neg\_float\_count++;}

%%

void main(int argc,char \*argv[])

{

//printf("Enter any Number: ");

yyin=fopen(argv[1],"r");

yylex();

printf("No of +ve integers: %d\n",pos\_int\_count);

printf("No of -ve integers: %d\n",neg\_int\_count);

printf("No of +ve float: %d\n",pos\_float\_count);

printf("No of -ve float: %d\n",neg\_float\_count);

}

9.Comments

%{

#include<stdio.h>

int c=0,m=0;

%}

%%

[/][/]([a-zA-Z0-9]\*|[\t]?)+[^\n]+ {c++;fprintf(yyout,"\n");}

[/][\*]([a-zA-Z0-9 ]\*|[\n]?|[\t]?)+[\*][/] {m++;}

%%

int main()

{

yyin = fopen("in.txt","r");

yyout = fopen("commentoutput.c","w");

yylex();

printf("Number of single line comments %d\n",c);

printf("Number of multiline coments %d\n",m);

}

int yywrap()

{

return 1;

}

10.printfscanf

%{

#include<stdio.h>

int pfc=0, sfc=0;

%}

%%

"printf" {fprintf(yyout,"writef"); pfc++;}

"scanf" {fprintf(yyout,"readf"); sfc++;}

%%

main(int argc, char \*argv[])

{

if(argc!=3)

{

printf("Usage: ./a.out in.txt out.txt\n");

exit(0);

}

yyin=fopen(argv[1],"r");

yyout=fopen(argv[2],"w");

yylex();

printf("\n the number of printf lines = %d\n",pfc);

printf("\n the number of scanf lines = %d\n",sfc);

}

int yywrap()

{

return 1;

}

11.RDP

|  |
| --- |
| #include<stdio.h> |
|  |

|  |
| --- |
| int i=0; |
|  |

|  |
| --- |
| char a[10]; |
|  |

|  |
| --- |
| void e(); |
|  |

|  |
| --- |
| void e1(); |
|  |

|  |
| --- |
| void t(); |
|  |

|  |
| --- |
| void t1(); |
|  |

|  |
| --- |
| void f(); |
|  |

|  |
| --- |
| int main() |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| while(1) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| printf("\nEnter a string :\n"); |
|  |

|  |
| --- |
| scanf("%s",&a); |
|  |

|  |
| --- |
| if(strcmp(a,"bye")) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| e(); |
|  |

|  |
| --- |
| if(a[i]=='$') |
|  |

|  |
| --- |
| printf("Successful parse\n"); |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| printf("Unsucessful parse\n"); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else |
|  |

|  |
| --- |
| break; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| return 0; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| void e(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| t(); |
|  |

|  |
| --- |
| e1(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| void e1(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| if(a[i]=='+') |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| i++; |
|  |

|  |
| --- |
| t(); |
|  |

|  |
| --- |
| e1(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void t(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| f(); |
|  |

|  |
| --- |
| t1(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| void t1(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| if(a[i]=='\*') |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| i++; |
|  |

|  |
| --- |
| f(); |
|  |

|  |
| --- |
| t1(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| void f(void) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| if(a[i]=='(') |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| i++; |
|  |

|  |
| --- |
| e(); |
|  |

|  |
| --- |
| if(a[i]==')') |
|  |

|  |
| --- |
| i++; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| else if(a[i]=='i') |
|  |

|  |
| --- |
| i++; |
|  |

}`

12. First

#include<stdio.h>

#include<ctype.h>

void first(char[],char );

void final(char[],char);

int nop;

char productionSet[10][10];

main()

{

int i;

char choice;

char c;

char result[20];

printf("How many number of productions ? :");

scanf(" %d",&nop);

for(i=0;i<nop;i++)//read production string eg: E=E+T

{

printf("Enter productions Number %d : ",i+1);

scanf(" %s",productionSet[i]);

}

do

{

printf("\n Find the FIRST of :");

scanf(" %c",&c);

first(result,c); //Compute FIRST; Get Answer in 'result' array

printf("\n first(%c)= { ",c);

for(i=0;result[i]!='\0';i++)

printf(" %c ",result[i]); //Display result

printf("}\n");

printf("press 'y' to continue : ");

scanf(" %c",&choice);

}

while(choice=='y'||choice =='Y');

}

void first(char\* Result,char c)

{

int i,j,k;

char subResult[20];

int foundEpsilon;

subResult[0]='\0';

Result[0]='\0';

if(!(isupper(c)))

{

final(Result,c);

return ;

}

for(i=0;i<nop;i++)

{

if(productionSet[i][0]==c)

{

if(productionSet[i][2]=='$') final(Result,'$');

else

{

j=2;

while(productionSet[i][j]!='\0')

{

foundEpsilon=0;

first(subResult,productionSet[i][j]);

for(k=0;subResult[k]!='\0';k++)

final(Result,subResult[k]);

for(k=0;subResult[k]!='\0';k++)

if(subResult[k]=='$')

{

foundEpsilon=1;

break;

}

if(!foundEpsilon)

break;

j++;

}

}

}

}

return ;

}

void final(char Result[],char val)

{

int k;

for(k=0 ;Result[k]!='\0';k++)

if(Result[k]==val)

return;

Result[k]=val;

Result[k+1]='\0';

}

13.follows

#include<stdio.h>

#include<string.h>

int n,m=0,p,i=0,j=0;

char a[10][10],followResult[10];

void follow(char c);

void first(char c);

void addToResult(char);

int main()

{

int i;

int choice;

char c,ch;

printf("Enter the no.of productions: ");

scanf("%d", &n);

printf(" Enter %d productions\nProduction with multiple terms should be give as separate productions \n", n);

for(i=0;i<n;i++)

scanf("%s%c",a[i],&ch);

// gets(a[i]);

do

{

m=0;

printf("Find FOLLOW of -->");

scanf(" %c",&c);

follow(c);

printf("FOLLOW(%c) = { ",c);

for(i=0;i<m;i++)

printf("%c ",followResult[i]);

printf(" }\n");

printf("Do you want to continue(Press 1 to continue....)?");

scanf("%d%c",&choice,&ch);

}

while(choice==1);

}

void follow(char c)

{

if(a[0][0]==c)addToResult('$');

for(i=0;i<n;i++)

{

for(j=2;j<strlen(a[i]);j++)

{

if(a[i][j]==c)

{

if(a[i][j+1]!='\0')first(a[i][j+1]);

if(a[i][j+1]=='\0'&&c!=a[i][0])

follow(a[i][0]);

}

}

}

}

void first(char c)

{

int k;

if(!(isupper(c)))

//f[m++]=c;

addToResult(c);

for(k=0;k<n;k++)

{

if(a[k][0]==c)

{

if(a[k][2]=='$') follow(a[i][0]);

else if(islower(a[k][2]))

//f[m++]=a[k][2];

addToResult(a[k][2]);

else first(a[k][2]);

}

}

}

void addToResult(char c)

{

int i;

for( i=0;i<=m;i++)

if(followResult[i]==c)

return;

followResult[m++]=c;

}

14.LL(1)

#include<stdio.h>

//#include<conio.h>

#include<stdlib.h>

#include<string.h>

//#include<process.h>

char prod[10][20],start[2];

char nonterm[10],term[10];

char input[10],stack[50];

int table[10][10];

int te,nte;

int n;

void main()

{

init();

parse();

}

init()

{

int i,j;

printf("\nNOTE:\n");

printf("The terminals should be entered in single lower case letters,special symbol and\n");

printf("non-terminals should be entered in single upper case letters.\n");

printf("extends to symbol is '->' and epsilon symbol is '@' \n");

printf("\nEnter the no. of terminals:");

scanf("%d",&te);

for(i=0;i<te;i++)

{

printf("Enter the terminal %d:",i+1);

scanf("%c",&term[i]);

}

term[i]='$';

printf("\nEnter the no. of non terminals:");

scanf("%d",&nte);

for(i=0;i<nte;i++)

{

printf("Enter the non-terminal %d:",i+1);

scanf("%c",&nonterm[i]);

}

printf("\nEnter the no. of productions:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("Enter the production %d:",i+1);

scanf("%s",prod[i]);

}

printf("\nEnter the start symbol:");

scanf("%c",&start[0]);

printf("\nEnter the input string:");

scanf("%s",input);

input[strlen(input)]='$';

printf("\n\nThe productions are:");

printf("\nProductionNo. Production");

for(i=0;i<n;i++)

printf("\n %d %s",i+1,prod[i]);

printf("\n\nEnter the parsing table:");

printf("\n Enter the production number in the required entry as mentioned above.");

printf("\n Enter the undefined entry or error of table as '0'\n\n");

for(i=0;i<nte;i++)

{

for(j=0;j<=te;j++)

{

printf("Entry of table[%c,%c]:",nonterm[i],term[j]);

scanf("%d",&table[i][j]);

}

}

}

parse()

{

int i,j,prodno;

int top=-1,current=0;

stack[++top]='$';

stack[++top]=start[0];

do

{

if((stack[top]==input[current])&&(input[current]=='$'))

{

printf("\nThe given input string is parsed");

exit(0);

}

else if(stack[top]==input[current])

{

top--;

current++;

}

else if(stack[top]>='A'&&stack[top]<='Z')

{

for(i=0;i<nte;i++)

if(nonterm[i]==stack[top]) break;

for(j=0;j<=te;j++)

if(term[j]==input[current]) break;

prodno=table[i][j];

if(prodno==0)

{

printf("\nThe given input string is not parsed");

exit(0);

}

else

{

for(i=strlen(prod[prodno-1])-1;i>=3;i--)

{

if(prod[prodno-1][i]!='@')

stack[top++]=prod[prodno-1][i];

}

top--;

}

}

else

{

printf("\nThe given input string is not parsed");

exit(0);

}

}while(1);

}

17.Nestedif

.l

%{

#include "y.tab.h"

%}

%%

"if" {return IF;}

[sS][0-9]\* {return S;}

"<"|">"|"=="|"<="|">="|"!=" {return RELOP;}

[0-9]+ {return NUMBER;}

[a-z][a-zA-Z0-9\_]\* {return ID;}

\n {return NL;}

. {return yytext[0];}

%%

.y

%{

#include<stdio.h>

#include<stdlib.h>

int count=0;

%}

%token IF RELOP S NUMBER ID NL

%%

stmt: if\_stmt NL {printf("No. of nested if statements=%d\n",count);exit(0);}

;

if\_stmt : IF'('cond')''{'if\_stmt'}' {count++;}

|S

;

cond: x RELOP x

;

x:ID | NUMBER

;

%%

int yyerror(char \*msg)

{

printf("the statement is invalid\n");

exit(0);

}

main()

{

printf("enter the statement\n");

yyparse();

}

18.anbn

.l

%{

#include<stdio.h>

#include"y.tab.h"

%}

%%

a return A;

b return B;

\n|. return yytext[0];

%%

.y

%{

#include<stdio.h>

int vd;

%}

%union

{

char dval;

}

%token <dval> A

%token <dval> B

%%

str: s '\n' { vd=1; return 0;}

s : A s B ;

| ;

%%

int main()

{

printf("enter the string");

yyparse();

if(vd==1)

printf(" valid");

else

printf(" not valid");

}

yyerror(char \*s)

{

printf("%s",s);

}

16.CALC

.l

|  |
| --- |
| %{ |
|  |

|  |
| --- |
| #include"y.tab.h" |
|  |

|  |
| --- |
| %} |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| %% |
|  |

|  |
| --- |
| [0-9]+ {yylval.dval=atoi(yytext);return digit;} |
|  |

|  |
| --- |
| \n|. return yytext[0]; |
|  |

%%

.y

|  |
| --- |
| %{ |
|  |

|  |
| --- |
| #include<stdio.h> |
|  |

|  |
| --- |
| %} |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| %union |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| int dval; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| %token <dval> digit |
|  |

|  |
| --- |
| %type <dval> expr |
|  |

|  |
| --- |
| %type <dval> expr1 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| %% |
|  |

|  |
| --- |
| line:expr '\n' {printf("%d\n",$1);} |
|  |

|  |
| --- |
| ; |
|  |

|  |
| --- |
| expr:expr '+' expr1 {$$=$1+$3;} |
|  |

|  |
| --- |
| |expr '-' expr1 {$$=$1-$3;} |
|  |

|  |
| --- |
| |expr '\*' expr1 {$$=$1\*$3;} |
|  |

|  |
| --- |
| |expr '/' expr1 {$$=$1/$3;} |
|  |

|  |
| --- |
| |expr1 |
|  |

|  |
| --- |
| ; |
|  |

|  |
| --- |
| expr1: '('expr')' {$$=$2;} |
|  |

|  |
| --- |
| | digit |
|  |

|  |
| --- |
| ; |
|  |

|  |
| --- |
| %% |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int main() |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| yyparse(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| yyerror(char \*s) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| printf("%s",s); |
|  |

}