1. Lexical Analyser of c  
%option noyywrap

%{

#include <stdio.h>

%}

identifier [a-zA-Z][a-zA-Z0-9\_]\*

%%

int|float|double|char printf("'%s' is a dtype.\n", yytext);

while|for|do|if|else|break|continue|return|goto|switch|case printf("'%s' is a keyword.\n", yytext);

"#".\*/\n printf("'%s' is a preprocessing directive\n", yytext);

"//".\*/\n printf("'%s' is a comment.\n", yytext);

"{" printf("'%s' indicates start of block.\n", yytext);

"}" printf("'%s' indicates end of block.\n", yytext);

\".\*\" printf("'%s' is a string.\n", yytext);

[0-9]+ printf("'%s' is a integer.\n", yytext);

"=" printf("'%s' indicates assignment\n", yytext);

"+"|"-"|"\*"|"/"|"&"|"|"|"^"|"%" printf("'%s' is a operator\n", yytext);

{identifier} printf("'%s' is a identifier\n", yytext);

"&"{identifier} printf("'%s' is a pointer.\n", yytext);

";" {}

"\n" {}

. {}

%%

int main() {

yylex();

return 0;

}

Input code:  
#include<stdio.h>

#include<conio.h>

void main()

{

int a,b,c;

a=1;

b=2;

c=a+b;

printf("Sum:%d",c);

}

**lex q1.L(small)**

**gcc lex.yy.c**

**cat input.c | ./a.out**

2.  
A) Left factor

#include <stdio.h>

int main(){

char b[50],r1[50],r2[50],p[50];int i=0,j=0,k=0;

scanf("%s",b);

while(b[i]!='|')r1[j++]=b[i++];r1[j]=0;i++;j=0;

while(b[i])r2[j++]=b[i++];r2[j]=0;

i=j=0;while(r1[i]&&r2[j]&&r1[i]==r2[j])p[k++]=r1[i++],j++;p[k]=0;

puts("---");

if(!r1[i]&&!r2[j])printf("A -> %s\n",p);

else{

printf("A -> %sA'\n",p);

if(!r1[i])printf("A' -> %s\n",r2+j);

else if(!r2[j])printf("A' -> %s\n",r1+i);

else printf("A' -> %s|%s\n",r1+i,r2+j);

}

}

gcc q.c

./a.out

**A->abcd|aBC**

B) Left Recurssion  
  
#include <stdio.h>

int main(){

char buf[50],r1[50],r2[50];int i=0,j=0;

scanf("%s",buf);

while(buf[i]!='|')r1[j++]=buf[i++];r1[j]=0;i++;j=0;

while(buf[i])r2[j++]=buf[i++];r2[j]=0;

if(r1[0]!='A'&&r2[0]!='A'){puts("No left recursion");return 0;}

char \*a,\*b;

if(r1[0]=='A'){b=r1+1;a=r2;}else{a=r1+1;b=r2;}

printf("A -> %sA'\nA' -> %sA'|epsilon\n",a,b);

}

gcc q.c

./a.out

A -> Abc|ax

3. LL Parsing

#include <stdio.h>

int g(char c){return c=='S'?0:c=='A'?1:c=='B'?2:c=='i'?0:c=='+'?1:c=='\*'?2:c=='$'?3:-1;}

char\* l(char t[3][4][5],char A,char a){int i=g(A),j=g(a);return i<0||j<0?0:t[i][j];}

int main(){

char t[3][4][5]={"A","","","", "Bi","","","", "","A+","A\*","e"};

char s[50]="$S",b[50];scanf("%s",b);

int i=1,j=0,k;char\* r;

while(b[j]){

printf("%s\t%s",s,b+j);

if(s[i]>='A'&&s[i]<='Z'){

r=l(t,s[i],b[j]);

if(!r||!\*r){puts("\nFAILED");break;}

if(\*r=='e')s[i--]=0;

else{for(k=0;r[k];k++)s[i++]=r[k];s[i--]=0;}

printf("\t(replace %s)\n",r);

}else{

if(s[i]==b[j]){

if(s[i]=='$'){puts("\nSUCCESS");break;}

s[i--]=0;j++;puts("\t(pop)");

}else{puts("\nFAILED");break;}

}

}

}

gcc q.c

./a.out

Enter: **i+i\*i$**

4. Simple Calculator

#include <stdio.h>

#include <string.h>

#define M 100

char s[M]; int p[M]; char o[M]; int n; char T='Z';

char get(int i,int d){

    while(i>=0 && s[i] && !strchr("+-\*/=:",s[i])){

        if(s[i]!='$'){ char c=s[i]; s[i]='$'; return c; }

        i+=d;

    } return '?';

}

int main(){

    printf("Enter the Expression: ");

    scanf("%s",s);

    char prec[]=":/\*+-";

    for(int k=0;k<5;k++) for(int i=0;s[i];i++) if(s[i]==prec[k]){p[n]=i;o[n++]=s[i];}

    for(int k=0;k<n;k++){

        char L=get(p[k]-1,-1),R=get(p[k]+1,1);

        printf("%c := %c %c %c\n",T,L,o[k],R);

        s[p[k]]=T--;

    }

    char r=get(0,1),l=get(strlen(s)-1,-1);

    printf("%c := %c\n",r,l);

}

Output:

yaac -d a.y

gcc y.tab.c -o a-lm

./a

2+3 simple calculation

5 . Symbol table

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

int main() {

char expr[100], c;

int i = 0;

printf("Input ending with $ sign: ");

while ((c = getchar()) != '$' && c != '\n') {

expr[i++] = c;

}

expr[i] = '\0';

printf("\nSymbol Table display\n");

printf("Symbol\tAddress\t\tType\n");

for (int j = 0; expr[j] != '\0'; j++) {

c = expr[j];

char \*ptr = (char \*)malloc(sizeof(char));

\*ptr = c;

if (isalpha(c)) {

printf("%c\t%p\tidentifier\n", c, (void \*)ptr);

} else if (c == '+' || c == '-' || c == '\*' || c == '=') {

printf("%c\t%p\toperator\n", c, (void \*)ptr);

}

}

return 0;

}

Run: Enter:

gcc .\code.c (file name) w=a+b\*c$

./a.exe

6. Intermediate code

#include <stdio.h>

#include <string.h>

#define M 100

char s[M]; int p[M]; char o[M]; int n; char T='Z';

char get(int i,int d){

    while(i>=0 && s[i] && !strchr("+-\*/=:",s[i])){

        if(s[i]!='$'){ char c=s[i]; s[i]='$'; return c; }

        i+=d;

    } return '?';

}

int main(){

    printf("Enter the Expression: ");

    scanf("%s",s);

    char prec[]=":/\*+-";

    for(int k=0;k<5;k++) for(int i=0;s[i];i++) if(s[i]==prec[k]){p[n]=i;o[n++]=s[i];}

    for(int k=0;k<n;k++){

        char L=get(p[k]-1,-1),R=get(p[k]+1,1);

        printf("%c := %c %c %c\n",T,L,o[k],R);

        s[p[k]]=T--;

    }

    char r=get(0,1),l=get(strlen(s)-1,-1);

    printf("%c := %c\n",r,l);

}

Run: Enter:

gcc .\code.c (file name) w=a+b\*c

./a.exe

7. Code optimisation

#include <stdio.h>

#include <string.h>

#define MAX 20

struct Exp { char l; char r[20]; } code[MAX], opt[MAX];

int main() {

int n,i,j,k=0;

printf("Enter no. of expr: ");

scanf("%d",&n);

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

printf("[%d] LHS: ",i+1);

scanf(" %c",&code[i].l);

printf(" RHS: ");

scanf("%s",code[i].r);

}

printf("\n-- Original Code --\n");

for(i=0;i<n;i++) printf("%c=%s\n",code[i].l,code[i].r);

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

int used=0;

for(j=i+1;j<n;j++) if(strchr(code[j].r,code[i].l)){ used=1; break; }

if(used||i==n-1) opt[k++]=code[i];

}

printf("\n After DCE \n");

for(i=0;i<k;i++) printf("%c=%s\n",opt[i].l,opt[i].r);

for(i=0;i<k;i++) for(j=i+1;j<k;j++)

if(strcmp(opt[i].r,opt[j].r)==0){

char x=opt[j].l; opt[j].l=0;

for(int m=j+1;m<k;m++){ char\*p=strchr(opt[m].r,x); if(p) \*p=opt[i].l; }

}

printf("\n After CSE \n");

for(i=0;i<k;i++) if(opt[i].l) printf("%c=%s\n",opt[i].l,opt[i].r);

return 0;

}

3 (Enter these values)

a

s

b

a+c

c

c\*5

gcc .\code.c (file name)

./a.exe

8.Target code

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int L[20],n;int chk(int k){for(int i=0;i<n;i++)if(k==L[i])return 1;return 0;}

void h(FILE\*f1,FILE\*f2,char o,char\*A,char\*B,char\*R){

 fscanf(f1,"%s%s%s",A,B,R);

 fprintf(f2,"\n\tLOAD %s,R0\n\tLOAD %s,R1",A,B);

 fprintf(f2,"\n\t%s R1,R0",o=='+'?"ADD":o=='-'?"SUB":o=='\*'?"MUL":"DIV");

 fprintf(f2,"\n\tSTORE R0,%s",R);

}

int main(){

 FILE\*f1,\*f2;char fn[20],op[10],A[8],B[8],R[8],c;int i=0;

 printf("File: ");scanf("%s",fn);

 if(!(f1=fopen(fn,"r"))||!(f2=fopen("target.txt","w")))return 0;

 while(fscanf(f1,"%s",op)!=EOF){

  if(chk(++i))fprintf(f2,"\nlabel#%d",i);

  if(!strcmp(op,"print")){fscanf(f1,"%s",R);fprintf(f2,"\n\tOUT %s",R);}

  else if(!strcmp(op,"goto")){fscanf(f1,"%s%s",A,B);fprintf(f2,"\n\tJMP %s,label#%s",A,B);L[n++]=atoi(B);}

  else if(!strcmp(op,"[]=")){fscanf(f1,"%s%s%s",A,B,R);fprintf(f2,"\n\tSTORE %s[%s],%s",A,B,R);}

  else if(!strcmp(op,"uminus")){fscanf(f1,"%s%s",A,R);fprintf(f2,"\n\tLOAD -%s,R1\n\tSTORE R1,%s",A,R);}

  else if(strchr("+-\*/%",op[0]))h(f1,f2,op[0],A,B,R);

  else if(op[0]=='='){fscanf(f1,"%s%s",A,R);fprintf(f2,"\n\tSTORE %s %s",A,R);}

  else if(op[0]=='>'||op[0]=='<'){fscanf(f1,"%s%s%s",A,B,R);fprintf(f2,"\n\tLOAD %s,R0\n\tJ%cT %s,label#%s",A,op[0]=='>'?'G':'L',B,R);L[n++]=atoi(R);}

 }

 fclose(f1);fclose(f2);

 f2=fopen("target.txt","r");while((c=fgetc(f2))!=EOF)putchar(c);fclose(f2);

}

Input.txt (create one)

+ t2 R0 t3

- t2 R1 t4

print t4

Run:

gcc .\a.c (file name)

./a.exe

Enter file : Input.txt