CDSS LAB PROGRAMS

Pranav N Ghatigar
ENG19CS0227
6-D, D2 batch

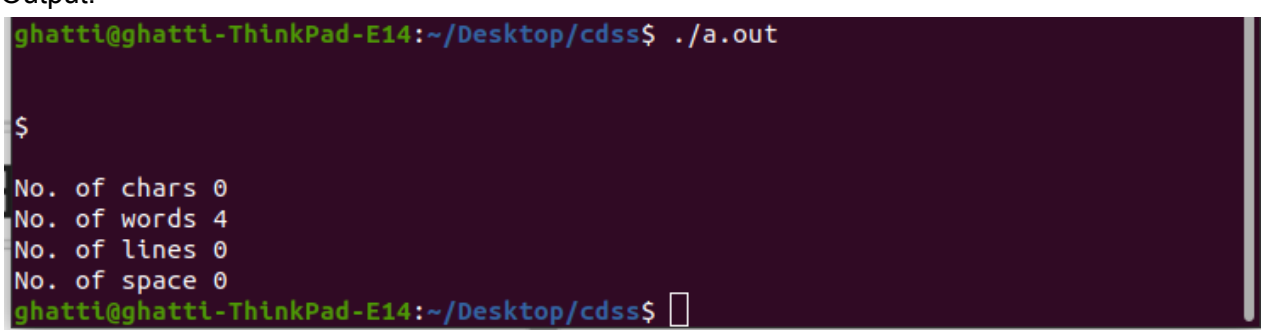
1a. Program to count the number, words, spaces and lines in a given input file

Code:

```
%{
        #include <stdio.h>
        int words = 0;
        int characters = 0;
        int lines = 0;
        int space = 0;
%}
%%
[a-zA-Z0-9]* {words++;}
%%
int main()
{
        yyin = fopen("Test","r");
        yylex();
        printf("No. of chars %d \n", characters);
        printf("No. of words %d \n", words);
        printf("No. of lines %d \n", lines);
        printf("No. of space %d \n", space);
}
```

Input File:

Output:



1b. Program to recognize and count the number of identifiers in a file

Code:

```
%{
#include<stdio.h>
int i=0;
%}
digit [0-9]
letter [a-z A-Z_]
%%
{letter}({letter}|{digit})* {i++;}
{digit}({letter}|{digit})* {i;}
%%
int main()
{
printf("Enter the values:\n");
yylex();
printf("Number of identifiers = %d\n", i);
return 0;
}
```

Output:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
Enter the values:
wfewe

1235434dwedw

vere23

Number of identifiers = 2
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

2a. Programs to count the numbers of comments lines in a given C program. Also eliminate them and copy the resulting program into a separate file.

Code:

```
%{
#include<stdio.h>
int s=0,m=0;
%}
%%
"/*"[a-zA-Z0-9' '\t\n]*"*/" m++;
"//".* s++;
%%
void main(){
yyin=fopen("f1.txt","r");
yyout=fopen("f2.txt","w");
yylex();
fclose(yyin);
fclose(yyout);
printf("no of single line comments=%d\n",s);
printf("no of multi line comments=%d\n",m);
}
int yywrap()
{return 1;}
```

Input file:

Open f1 ~/Desktop/cdss

```
icnwie
/* this is comment of multi
lines for firoaincw */
heiieonaa
//ncwoiew

wocniwn
//cwonicw
```

Output File:

Open f2 ~/Desktop/cdss

```
icnwie

heiieonaa


wocniwn

```

2b. Program to recognize whether a given sentence is simple or compound.

Code:

```
%{
#include<stdio.h>
int c=0;
%}
%%
[a-zA-Z]*[ ](and|or|but|yet|so)[ ][a-zA-Z]* {c=1;}
.|[\n];
%%
int yywrap()
{
return 1;
}
void main(){
printf("enter the text\n");
yylex();
if(c)
{
printf("The given statement is compound\n");
}
else
{
printf("The given statement is simple\n");
}
}
```

Output:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ cc lex.yy.c -ll
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
enter the text
It is raining outside.

The given statement is simple
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

3a. Program to count number of: i.+ve and -ve integers
ii. +ve and -ve fractions

Code:

```
%{
#include<stdio.h>
int pi=0,ni=0,pf=0,nf=0;
%}
%%
[-][0-9]+ {ni++;}
[+]?[0-9]+ {pi++;}
[-][0-9]*\.[0-9]+ {nf++;}
[+]?[0-9]*\.[0-9]+ {pf++;}
%%
void main(int argc,char *argv[])
{
if(argc!=2)
{
printf("usage :./a.out in.txt \n");
exit(0);
}
yyin=fopen(argv[1],"r");
yylex();
printf("Number of positive integer %d\n",pi);
printf("Number of negative integer %d\n",ni);
printf("Number of positive fraction %d\n",pf);
printf("Number of negative fraction %d\n",nf);
}
int yywrap(){
return 1;
}
```

Input File:

Open

in.txt
~/Desktop/cdss

```
5
-10
5.67
-78.1
```

Output:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out in.txt




Number of positive integer 1
Number of negative integer 1
Number of positive fraction 1
Number of negative fraction 1
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

3b. Program to count the number of “scanf” and “printf” statements in a C program.
Replace them with “readf” and “writef” statements respectively.

Code:

```
%{
#include<stdio.h>
int sf=0,pf=0;
%}
%%
"scanf" {sf++; fprintf(yyout,"readf");}
"printf" {pf++; fprintf(yyout,"writef");}
%%
int main()
{
yyin=fopen("f1.c","r");
yyout=fopen("f2.c","w");
yylex();
printf("no of scanf =%d\n no of printf =%d\n",sf,pf);
return 0;
}
```

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ cc lex.yy.c -ll
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
no of scanf =1
 no of printf =2
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

Input File:

Open

f1.c
~/Desktop/cdss

```c
#include<stdio.h>
int main()
{
int a,b,c;
printf("enter the values a and b \n");
scanf("%d%d",&a,&b);
c=a+b;
printf("Sum =%d",c);
return 0;
}
```

Output File:

Open

f2.c
~/Desktop/cdss

```c
#include<stdio.h>
int main()
{
int a,b,c;
writef("enter the values a and b \n");
readf("%d%d",&a,&b);
c=a+b;
writef("Sum =%d",c);
return 0;
}
```

4. Program to evaluate arithmetic expression involving operators +, -, *, /

Code:

//lex code

```
%{
#include "y.tab.h"
extern yylval;
%}
%%
[0-9]+ {yylval=atoi(yytext);return num;}
[\+\-\*\/] {return yytext[0];}
[)] {return yytext[0];}
[(] {return yytext[0];}
. {;}
\n {return 0;}
%%
```

//yacc code

```
%{#include<stdio.h>
#include<stdlib.h>
%}
%token num
%left '+"-'
%left '*"/'
%%
input:exp {printf("%d\n",$$);exit(0);}
exp:exp'+'exp {$$=$1+$3;}
|exp'-'exp {$$=$1-$3;}
|exp'*'exp {$$=$1*$3;}
|exp'/'exp {if($3==0){printf("Division by zero\n");exit(0);}
else
$$=$1/$3;}
|'('exp')' {$$=$2;}
|num {$$=$1;};
%%
int yyerror()
{
printf("error");
exit(0);
}
int main()
{
printf("Enter the expression:\n");
yyparse();
```

}

Output File:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
Enter the expression:
5*7
35
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
Enter the expression:
0/5
0
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

5. Program to recognize a valid variable which starts with a letter, followed by any number of letter or digits

Code:

//lex code

```
%{
#include "y.tab.h"
%}
%%
[a-zA-z] return L;
[0-9] return D;
%%
```

//yacc code

```
%{
#include<stdio.h>
#include<stdlib.h>
%}
%token L D
%%
var:L E {printf("Valid Variable\n"); return 0;}
E:E L;
|E D;
| ;
%%
int main()
{
printf("Type the variable\n");
yyparse();
return 0;
}
int yyerror()
{
printf("Invalid Variable\n");
exit(0);
}
```

Output:

6. Program to recognize the strings using the grammar ($a^n b^n$; n>=0)

Code:

//lex code

```
%{
#include "y.tab.h"
%}
%%
a return A;
b return B;
. return yytext[0];
\n return yytext[0];
%%
```

//yacc code

```
%{
#include<stdio.h>
#include<stdlib.h>
%}
%token A B
%%
Str:S '\n' {return 0;}
S:A S B;
| ;
%%
int main()
{
printf("Type the string\n");
if (!yyparse())
printf("Valid String\n");
return 0;
}
int yyerror()
{
printf("Invalid String\n");
exit(0);
}
```

Output:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
Type the string
abb
Invalid String
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
Type the string
aaabbb
Valid String
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

7. C program to implement Pass1 of Assembler

Code:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
char opcode[10], operand[10], label[10], code[10], mnemonic[3];
int locctr, start, length;
FILE *fp1,*fp2,*fp3,*fp4;
fp1=fopen("Input.txt","r");
fp2=fopen("Optab.txt","r");
fp3=fopen("Symtab.txt","w");
fp4=fopen("Out.txt","w");
fscanf(fp1,"%s\t%s\t%s", label,opcode,operand);
if(strcmp(opcode,"START")==0)
{
start=atoi(operand);
locctr=start;
fprintf(fp4,"\t%s\t%s\t%s\n",label,opcode,operand);
fscanf(fp1,"%s\t%s\t%s",label,opcode,operand);
}
else
locctr=0;
while(strcmp(opcode,"END")!=0)
{
fprintf(fp4,"%d\t",locctr);
if(strcmp(label,"**")!=0)
fprintf(fp3,"%s\t%d\n",label,locctr);
fscanf(fp2,"%s\t%s",code,mnemonic);
while(strcmp(code,"END")!=0)
{
if(strcmp(opcode,code)==0)
{
locctr+=3;
break;
}
fscanf(fp2,"%s\t%s",code,mnemonic);
}
if(strcmp(opcode,"WORD")==0)
locctr+=3;
else if(strcmp(opcode,"RESW")==0)
locctr+=(3*(atoi(operand)));
else if(strcmp(opcode,"RESB")==0)
```

```
locctr+=atoi(operand);
else if(strcmp(opcode,"BYTE")==0)
++locctr;
fprintf(fp4,"%s\t%s\t%s\t\n",label,opcode,operand);
fscanf(fp1,"%s\t%s\t%s",label,opcode,operand);
}
fprintf(fp4,"%d\t%s\t%s\t%s\n",locctr,label,opcode,operand);
length=locctr-start;
printf("The length of the code:%d\n",length);
fclose(fp1);
fclose(fp2);
fclose(fp3);
fclose(fp4);
}
```

Input File:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
        COPY    START   1000

1000    -       LDA     ALPHA

1003    -       ADD     ONE

1006    -       SUB     TWO

1009    -       STA     BETA

1012    ALPHA   BYTE    C'KLNCE

1017    ONE     RESB    2

1019    TWO     WORD    5

1022    BETA    RESW    1

1025    -       END     -
Program length =25
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

Output:

```
Open                                    optab.txt
                                        ~/Desktop/cdss
1 LDA     00
2 STA     23
3 ADD     01
4 SUB     05
```

Open

symtab.txt
~/Desktop/cdss

```
1
2 1012    ALPHA
3
4 1017    ONE
5
6 1019    TWO
7
8 1022    BETA
```

8. C program to implement Absolute loader.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
        FILE *fp;
 int i,addr1,l,j,staddr1;
 char name[10],line[50],name1[10],addr[10],rec[10],ch,staddr[10];
 //clrscr();
 printf("enter program name:" );
 scanf("%s",name);
 fp=fopen("8_src.txt","r");
 fscanf(fp,"%s",line);
 for(i=2,j=0;i<8,j<6;i++,j++)
  name1[j]=line[i];
  name1[j]='\0';
 printf("name from obj. %s\n",name1);
 if(strcmp(name,name1)==0)
  {
  do
  {
  fscanf(fp,"%s",line);
  if(line[0]=='T')
  {
  for(i=2,j=0;i<8,j<6;i++,j++)
  staddr[j]=line[i];
  staddr[j]='\0';
  staddr1=atoi(staddr);
  i=12;
  while(line[i]!='$')
  {
```

```
      if(line[i]!='^')
      {
        printf("00%d \t %c%c\n", staddr1,line[i],line[i+1]);
        staddr1++;
        i=i+2;
      }
      else i++;
    }
    }
    else if(line[0]='E')
    fclose(fp);
    }while(!feof(fp));
    }
//getch();
}
```

Output:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ gcc absload.c
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
enter program name:sample
name from obj. sample
001000   00
001001   10
001002   03
001003   07
001004   10
001005   09
002000   11
002001   11
002002   11
free(): invalid pointer
Aborted (core dumped)
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

9. C program to implement the FIRST in context free grammar

```
#include<stdio.h>
#include<ctype.h>
void FIRST(char[],char );
void addToResultSet(char[],char);
int numOfProductions;
char productionSet[10][10];
void main()
{
   int i;
   char choice;
   char c;
   char result[20];
   printf("How many number of productions ? :");
   scanf(" %d",&numOfProductions);
   for(i=0;i<numOfProductions;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 X is terminal, FIRST(X) = {X}.
   if(!(isupper(c)))
   {
```

```
        addToResultSet(Result,c);
              return ;
    }
    for(i=0;i<numOfProductions;i++)
    {
        if(productionSet[i][0]==c)
        {
 if(productionSet[i][2]=='$') addToResultSet(Result,'$');
      else
           {
               j=2;
               while(productionSet[i][j]!='\0')
               {
               foundEpsilon=0;
               FIRST(subResult,productionSet[i][j]);
               for(k=0;subResult[k]!='\0';k++)
                   addToResultSet(Result,subResult[k]);
                for(k=0;subResult[k]!='\0';k++)
                    if(subResult[k]=='$')
                    {
                        foundEpsilon=1;
                        break;
                    }
                if(!foundEpsilon)
                    break;
                j++;
               }
           }
    }
}
    return ;
}
void addToResultSet(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';
}
```

Output.txt

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
How many number of productions ? :8
Enter productions Number 1 : Epsilon=$
Enter productions Number 2 : E=TD
Enter productions Number 3 : D=+TD
Enter productions Number 4 : D=$
Enter productions Number 5 : T=FS
Enter productions Number 6 : S=*FS
Enter productions Number 7 : S=$
Enter productions Number 8 : F=(E)

 Find the FIRST of  :E

 FIRST(E)= {  s  ( }
press 'y' to continue : Y

 Find the FIRST of  :D

 FIRST(D)= {  +  $ }
press 'y' to continue : n
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

10. C program to implement Shift Reduce Parser for the given grammar:

E → E+E
E → E*E
E →(E)
E→ id

```
#include<stdio.h>
#include<string.h>
int k=0,z=0,i=0,j=0,c=0;
char a[16],ac[20],stk[15],act[10];
void check();
int main()
  {

     puts("GRAMMAR is E->E+E \n E->E*E \n E->(E) \n E->id");
     puts("enter input string ");
     gets(a);
     c=strlen(a);
     strcpy(act,"SHIFT->");
     puts("stack \t     input \t action");
     for(k=0,i=0; j<c; k++,i++,j++)
      {
        if(a[j]=='i' && a[j+1]=='d')
         {
            stk[i]=a[j];
            stk[i+1]=a[j+1];
            stk[i+2]='\0';
            a[j]=' ';
            a[j+1]=' ';
            printf("\n$%s\t%s$\t%sid",stk,a,act);
            check();
         }
        else
         {
            stk[i]=a[j];
            stk[i+1]='\0';
            a[j]=' ';
            printf("\n$%s\t%s$\t%ssymbols",stk,a,act);
            check();
         }
      }

  }
void check()
```

```
{
  strcpy(ac,"REDUCE TO E");
  for(z=0; z<c; z++)
    if(stk[z]=='i' && stk[z+1]=='d')
      {
        stk[z]='E';
        stk[z+1]='\0';
        printf("\n$%s\t%s$\t%s",stk,a,ac);
        j++;
      }
  for(z=0; z<c; z++)
    if(stk[z]=='E' && stk[z+1]=='+' && stk[z+2]=='E')
      {
        stk[z]='E';
        stk[z+1]='\0';
        stk[z+2]='\0';
        printf("\n$%s\t%s$\t%s",stk,a,ac);
        i=i-2;
      }
  for(z=0; z<c; z++)
    if(stk[z]=='E' && stk[z+1]=='*' && stk[z+2]=='E')
      {
        stk[z]='E';
        stk[z+1]='\0';
        stk[z+1]='\0';
        printf("\n$%s\t%s$\t%s",stk,a,ac);
        i=i-2;
      }
  for(z=0; z<c; z++)
    if(stk[z]=='(' && stk[z+1]=='E' && stk[z+2]==')')
      {
        stk[z]='E';
        stk[z+1]='\0';
        stk[z+1]='\0';
        printf("\n$%s\t%s$\t%s",stk,a,ac);
        i=i-2;
      }
}
```

Output:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
GRAMMAR is E->E+E
 E->E*E
 E->(E)
 E->id
enter input string
id+id*id
stack    input   action

$id       +id*id$       SHIFT->id
$E        +id*id$       REDUCE TO E
$E+        id*id$       SHIFT->symbols
$E+id        *id$       SHIFT->id
$E+E         *id$       REDUCE TO E
$E           *id$       REDUCE TO E
$E*           id$       SHIFT->symbols
$E*id           $       SHIFT->id
$E*E            $       REDUCE TO E
$E              $       REDUCE TO E
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

11. C program to implement code optimization techniques.

Code:

FOR LOOP:

```
#include<stdio.h> //using for loop
int main()
{int i,fact=1,number;
printf("Enter a number: ");
scanf("%d",&number);
for(i=1;i<=number;i++){
fact=fact*i;
}printf("Factorial of %d is: %d",number,fact);
return 0;}
```

RECURSION:

```
#include<stdio.h> // using Recursion
long factorial(int n)
{if (n == 0)
return 1;
else
return(n * factorial(n-1));
}
void main()
{int number;
long fact;
printf("Enter a number: ");
scanf("%d", &number);
fact = factorial(number);
printf("Factorial of %d is %ld\n", number, fact);
return 0;
}
```

DO WHILE:

```
#include<stdio.h> // using do-while loop
void main()
{int n,i=1,f=1;
printf("\n Enter The Number:");
scanf("%d",&n);
do
{f=f*i;
i++;
}while(i<=n);
printf("\n The Factorial of %d is %d",n,f);
}
```

Output:

Do While:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ gcc 11do.c
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out

 Enter The Number:10

 The Factorial of 10 is 3628800
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

For Loop:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ gcc 11for.c
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
Enter a number: 10
Factorial of 10 is: 3628800
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```

Recursion:

```
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ gcc 11rec.c
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$ ./a.out
Enter a number: 10
Factorial of 10 is 3628800
ghatti@ghatti-ThinkPad-E14:~/Desktop/cdss$
```