

Compiler Design Lab

(KCS 552)

Laboratory File For

(3rd Year/5th Semester Students)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY

Submitted By:-

Akansha Gupta

Submitted to:-

Mrs. Mohini Chakarverti

Class/Sec:- CSE A

Roll no. :- 1901330100026

INDEX

S.NO	EXPERIMENT	DATE	REMARKS	SIGNATURE
1.	AIM: WAP to check whether the string is a keyword.			
2.	AIM: To check whether string has constant or not.			
3.	AIM: WAP to check whether the string contains 'a,aa....." or not.			
4.	AIM: WAP to check whether the string starts with ab or not.			
5.	AIM: WAP to check whether the string starts with 'a' and ends with 'b'.			
6.	AIM: WAP to check whether the string contains 'ab' as substring.			
7.	AIM: WAP in C that string contains a identifier or not.			

INDEX

S.NO	EXPERIMENT	DATE	REMARKS	SIGNATURE
8.	AIM: WAP to remove left recursion.			
9.	AIM: WAP to remove left factoring.			
10.	AIM: WAP to check whether String contains comment or not.			
11.	AIM: Write a Program to print each identifier and operator from a given string without considering spaces.			
12.	AIM: WAP to implement operator precedence parsing.			

EXPERIMENT-1

AIM-WRITE A PROGRAM WHETHER THE STRING IS KEYWORD OR NOT.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

void main(){
    int i,flag=0,m;
    char
    s[18][10]={"if","else","goto","continue","return","for","while","break","switch","union","struct","default"
    , "int","static","char","double"},st[10];

    clrscr();
    printf("Akansha Gupta\n1901330100026");
    printf("\nEnter a string: ");
    gets(st);
    for(i=0;i<18;i++){
        m=strcmp(st,s[i]);
        if(m==0)
            flag=1;
    }
    if(flag==0)
        printf("It is not a keyword");
    else
        printf("It is a keyword");
    getch();
}
```

EXPERIMENT-1

```
Akansha Gupta  
1901330100026  
Enter a string: int  
It is a keyword
```

EXPERIMENT-2

AIM-WRITE A PROGRAM WHETHER THE STRING IS CONSTANT OR NOT.

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
void main(){
    char s[30];
    int len,a;
    clrscr();
    printf("Akansha Gupta\n1901330100026");

    printf("\n Input a string");
    gets(s);
    len=strlen(s);
    a=0;
    while(a<len){
        if(isdigit(s[a])){
            {
                a++;
            }
        }
        else {
            printf("string is not constant");
            break;
        }
    }
    if(a==len){
        printf("string is constant");
    }
    getch();
```

EXPERIMENT-2

}

Akansha Gupta

1901330100026

Input a string111

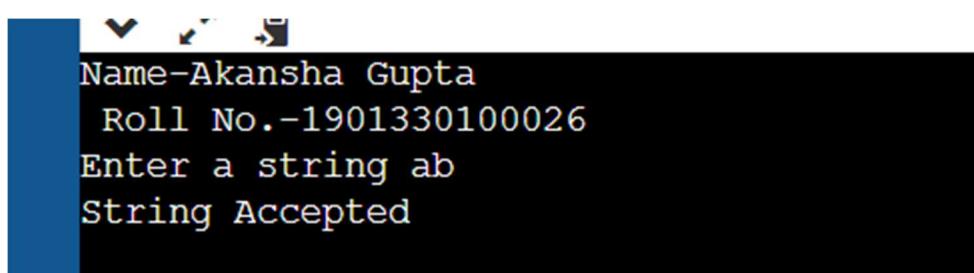
string is constant

EXPERIMENT-3

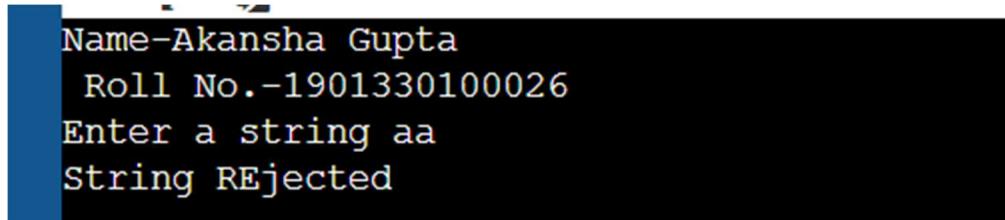
AIM- WRITE A PROGRAM TO CHECK WHETHER STRING STARTS WITH ab INPUT SYMBOL.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>

void main(){
    char str[40];
    printf("Name-Akansha Gupta\n Roll No.-1901330100026\n");
    printf("Enter a string");
    scanf("%s",str);
    if((str[0]=='a')&&(str[1]=='b')){
        printf("String Accepted");
    }
    else
    {
        printf("String REjected");
    }
    getch();
}
```



```
Name-Akansha Gupta
Roll No.-1901330100026
Enter a string ab
String Accepted
```



```
Name-Akansha Gupta
Roll No.-1901330100026
Enter a string aa
String REjected
```

EXPERIMENT-4

AIM-WRITE A PROGRAM WHETHER THE STRING CONTAINS “aaa” OR NOT.

```
#include<string.h>
#include<stdlib.h>
void main(){
    char str[40];
    int n,i,c=0;

    printf("Name-Akansha Gupta\n Roll No.-1901330100026\n");
    printf("enter the string : ");
    scanf("%s",str);
    n=strlen(str);

    for(i=0;i<n;i++){
        if(str[i]!='a'){
            printf("string rejected");
            c=1;
            break;
        }
    }
    if(c==0)
    {
        printf("String Accepted");
    }

    getch();
}
```

EXPERIMENT-4

```
Name-Akansha Gupta  
Roll No.-1901330100026  
enter the string : aaa  
String Accepted
```

```
Name-Akansha Gupta  
Roll No.-1901330100026  
enter the string : abbab  
string rejected
```

EXPERIMENT-5

AIM-WAP TO CHECK WHETHER THE STRING STARTS WITH 'a' AND ENDS WITH 'b'

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char str[40];
    int n;
    printf("Name: AKANSHA GUPTA");
    printf("\n Roll No.: 1901330100026");
    printf("\n Input a string: ");
    scanf("%s",str);
    n=strlen(str);
    if((str[0]=='a')&&(str[n-1]=='b'))
    {
        printf("String starts with a and ends with b.");
    }
    else
    {
        printf("String does not starts with a and ends with b.");
    }
    getch();
}
```

```
Name: AKANSHA GUPTA
Roll No.: 1901330100026
Input a string: aughjfbb
String starts with a and ends with b.
```

EXPERIMENT-6

AIM-WAP TO CHECK WHETHER THE STRING CONTAINS 'ab' AS SUBSTRING

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

void main()
{
    char str[20];
    int i,n,c=0;
    printf("Name: Akansha Gupta");
    printf("\n Roll No.: 1901330100026");
    printf("\n Input a string: ");
    scanf("%s",str);
    n=strlen(str);
    for(i=0;i<n;i++)
    {
        if((str[i]=='a') && (str[i+1]=='b'))
            c++;
    }
    if (c>0)
    {
        printf("String has substring ab.");
    }
    else
    {
```

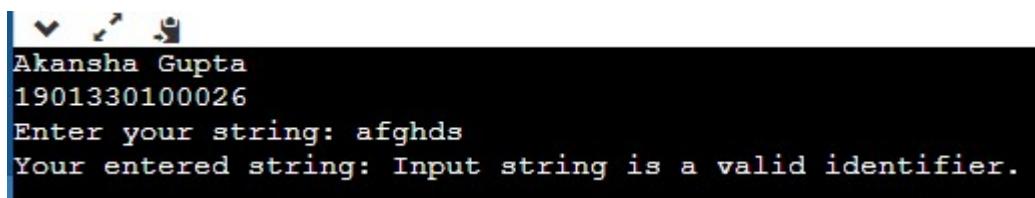
```
    printf("String does not have substring ab.");  
}  
  
getch();  
}
```

```
Name: Akansha Gupta  
Roll No.: 1901330100026  
Input a string: abgaghjg  
String has substring ab.
```

EXPERIMENT-7

AIM- WAP TO CHECK WHETHER STRING IS IDENTIFIER OR NOT

```
#include<conio.h>
#include<string.h>
#include<stdio.h>
int main()
{
char input[50];
int i;
printf("\nAkansha Gupta\n1901330100026 \n");
printf("Enter your string: ");
scanf("%c",input);
printf("Your entered string: ");
scanf("%c",input);
i = input[0];
if(i>=65 && i<=90 || i>=97 && i<=122 || i==95)
printf("Input string is a valid identifier.");
else
printf("Input string is not a valid identifier.");
return 0;
}
```



```
Akansha Gupta
1901330100026
Enter your string: afghds
Your entered string: Input string is a valid identifier.
```

EXPERIMENT – 8

AIM - WAP to remove left Recursion

CODE-

```
#include<stdio.h>
#include<string.h>
void main() {
    char input[100],l[50],r[50],temp[10],tempprod[20],productions[25][50];
    int i=0,j=0,flag=0,consumed=0;
    printf("\n\nAkansha Gupta\n1901330100026\n");
    printf("Enter the productions: ");
    scanf("%1s->%s",l,r);  printf("%s",r);
    while(sscanf(r+consumed,"%[^]s",temp) == 1 && consumed <= strlen(r)) {
        if(temp[0] == l[0]) {
            flag = 1;
            sprintf(productions[i++],"%s->%s%s\0",l,temp+1,l);
        }
        else
            sprintf(productions[i++],"%s->%s%s\0",l,temp,l);
        consumed += strlen(temp)+1;
    }
    if(flag == 1) {
        sprintf(productions[i++],"%s->\0",l);
        printf("The productions after eliminating Left Recursion are:\n");
        for(j=0;j<i;j++)
            printf("%s\n",productions[j]);
    }
    else
        printf("The Given Grammar has no Left Recursion");
}
```

OUTPUT-

```
Akansha Gupta
1901330100026
Enter the productions: c->cdd/cd
cdd/cdThe productions after eliminating Left Recursion are:
c->dd/cdc'
c-> $\epsilon$ 
```

EXPERIMENT – 9

AIM - Write a program to remove left factoring.

CODE-.

```
#include<stdio.h>
#include<string.h>
void main()
{
char a[100],b[100],c[100],f[100],g[100],h[100];
int i,n=0,j;
printf("\n\nAkansha Gupta\n1901330100026\n");
printf("Enter the grammar\n");
gets(a);
i=2;
j=0;
while(a[i]!='/'){
    b[j]=a[i];
    i++;
    j++;
}
b[j]='\0';
i++;
while(a[i]!='\0'){
    c[n]=a[i];
    n++;
    i++;
}
}
```

```

c[n]='\0';
i=0;
while(b[i]==c[i]){
f[i]=b[i];
i++;
}
f[i]='\0';
n=0;
j=i;
while(b[i]!='\0'){
g[n]=b[i];
n++;
i++;
}
g[n]='\0';
n=0;
while(c[j]!='\0'){
h[n]=c[j];
n++;
j++;
}
h[n]='\0';
printf("S-%sD\n",f);
printf("D-%s/%s/e\n",g,h);
}

```

OUTPUT-

```

Akansha Gupta
1901330100026
Enter the grammar
s->abc/aBc/e
S-D
D->abc/aBc/e/e

```

EXPERIMENT – 10

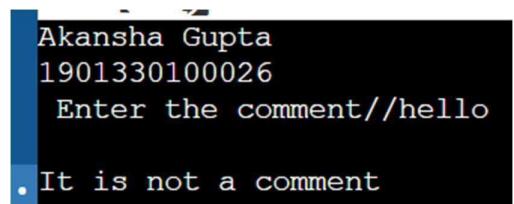
AIM: WAP to check whether String contains comment or not.

CODE-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char com[30];
    int i=2,a=0;
    clrscr();
    printf("\nAkansha Gupta\n1901330100026\n ");
    printf("Enter the comment");
    gets(com);
    if(com[0]=='/')
    {
        if(com[1]== '/')
            printf("\n It is a comment");
        else if(com[1]=='*')
        {
            for(i=2;i<=30;i++)
            {
                if(com[i]=='*'&&com[i+1]=='/')
                {
                    printf("\nIt is a comment");
                    a=1;
                    break;
                }
            }
        }
        else
            continue;
    }
}
```

```
    }
    if(a==0)
        printf("\nIt is a not comment");
    }
    else
        printf("\nIt is not a comment");
}
else
    printf("\nIt is not a comment");
getch();
}
```

OUTPUT-



```
Akansha Gupta
1901330100026
Enter the comment//hello
It is not a comment
```

EXPERIMENT – 11

AIM: Write a Program to print each identifier and operator from a given string without considering spaces.

CODE-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

int main()
{
    int i,j,len;
    char str[20];
    clrscr();

    printf("Akansha Gupta\n1901330100026\n");
    printf("Enter Your String: ");
    gets(str);

    len = sizeof(str)/sizeof(str[0]);

    for(i = 0; i < len; i++)
    {
        if(str[i] == ' ')
        {
            for(j=i;j<len;j++)
            {
                str[j]=str[j+1];
            }
            len--;
        }
    }

    printf("String after removing all the white spaces : %s", str);
```

```
getch();  
}
```

OUTPUT-

```
Akansha Gupta  
1901330100026  
Enter Your String: Akansha Gupta  
String after removing all the white spaces : Akansha
```

EXPERIMENT -12

CODE-

```
#include<stdio.h>
#include<string.h>
char *input;
int i=0;
char lasthandle[6],stack[50],handles[][][5]={"")E(),"E*E","E+E","i","E^E"};
//(E) becomes )E( when pushed to stack
int top=0,l;
char prec[9][9]={
    /input/
    /*stack + - * / ^ i ( ) $ */
    /* + */ '>', '>', '<', '<', '<', '<', '<', '>', '>',
    /* - */ '>', '>', '<', '<', '<', '<', '<', '>', '>',
    /* * */ '>', '>', '>', '>', '<', '<', '<', '>', '>',
    /* / */ '>', '>', '>', '>', '<', '<', '<', '>', '>',
    /* ^ */ '>', '>', '>', '>', '<', '<', '<', '>', '>',
    /* i */ '>', '>', '>', '>', 'e', 'e', '>', '>',
    /* ( */ '<', '<', '<', '<', '<', '<', '<', '>', 'e',
    /* ) */ '>', '>', '>', '>', 'e', 'e', '>', '>',
    /* $ */ '<', '<', '<', '<', '<', '<', '<', '<', '>',
};int getindex(char c)
{switch(c){
```

```
case '+':return 0;
case '-':return 1;
case '*':return 2;
case '/':return 3;
case '^':return 4;
case 'l':return 5;
case '(':return 6;
case ')':return 7;
case '$':return 8;
}
}int shift()
{stack[++top]=*(input+i++);
stack[top+1]='\0';
}int reduce()
{
int i,len,found,t;
for(i=0;i<5;i++)//selecting handles
{
len=strlen(handles[i]);
if(stack[top]==handles[i][0]&&top+1>=len)
{
found=1;
for(t=0;t<len;t++)
{
```

```
if(stack[top-t]!=handles[i][t])
{
    found=0;
    break;
}
if(found==1)
{
    stack[top-t+1]='E';
    top=top-t+1;
    strcpy(lasthandle,handles[i]);
    stack[top+1]='\0';
    return 1;//successful reduction
}
}
}
return 0;
}void dispstack()
{
int j;
for(j=0;j<=top;j++)
printf("%c",stack[j]);
}void dispinput(){
int j;
for(j=i;j<l;j++)
```

```
printf("%c",*(input+j));

}void main(){

int j;

input=(char*)malloc(50*sizeof(char));

printf("Akansha Gupta\n 1901330100026\n");

printf("\nEnter the string\n");

scanf("%s",input);

input=strcat(input,"$");

l=strlen(input);

strcpy(stack,"$");

printf("\nSTACK\tINPUT\tACTION");

while(i<=l){

shift();

printf("\n");

dispstack();

printf("\t");

dispinput();

printf("\tShift");

if(prec[getindex(stack[top])][getindex(input[i])]=='>')

{

while(reduce()){

printf("\n");

dispstack();

printf("\t");
```

```

dispinput();
printf("\tReduced: E->%s",lasthandle);
}
}
}

if(strcmp(stack,"$E$")==0)
printf("\nAccepted;");
else
printf("\nNot Accepted;")}

```

OUTPUT-

```

Enter the string
i*(i+i)*i
STACK   INPUT   ACTION
$i  *(i+i)*i$  Shift
$E  *(i+i)*i$  Reduced: E->i
$E* (i+i)*i$  Shift
$E*( i +i)*i$ Shift
$E*(i  +i)*i$ Shift
$E*(E  +i)*i$ Reduced: E->i
$E*(E+ i)*i$  Shift
$E*(E+i )*i$  Shift
$E*(E+E )*i$  Reduced: E->i
$E*(E  )*i$  Reduced: E->E+E
$E*(E) *i$ Shift
$E*E  *i$ Reduced: E->)E(
$E  *i$ Reduced: E->E*E
$E* i$ Shift
$E*i    $ Shift
$E*E    $ Reduced: E->i
$E  $  Reduced: E->E*E
$E$    Shift
$E$... Shift
Accepted;

```