Experiment Number 7

Aim: Shift Reduce Parsing.

Algorithm:

Step 1: Start

Step 2: Read productions and input string.

Step 3: Start with start symbol and end it with $ for both stacks.

Step 4: Parse string.

Step 5: Print respective output.

Step 6: Stop.

Code:

```
#include<iostream>
#include<string.h>
using namespace std;
struct prodn
{
char p1[10];
char p2[10];
};
int main()
{
char input[20],stack[50],temp[50],ch[2],*t1,*t2,*t;
int i,j,s1,s2,s,count=0;
struct prodn p[10];
int nop;
cout<<"Enter no of productions:";
cin>>nop;
cout<<"nop="<<nop;
cout<<"\nEnter the productions as E->E*E and at end input string:\n";
for(int i=0;i<nop;i++)
{
scanf("%s\n",temp);
t1=strtok(temp,"->");
t2=strtok(NULL,"->");
strcpy(p[count].p1,t1);
strcpy(p[count].p2,t2);
count++;
}
stack[0]='\0';
cout<<"Input String is :\n";
```

```
cin>>input;
cout<<input;
cout<<"\nStack\n";
i=0;
int runcount=0;
while(1)
{
if(i<strlen(input))
{
ch[0]=input[i];
ch[1]='\0';
i++;
strcat(stack,ch);
cout<<"\n"<<stack;
runcount++;
}
for(j=0;j<count;j++)
{
t=strstr(stack,p[j].p2);
if(t!=NULL)
{
s1=strlen(stack);
s2=strlen(t);
s=s1-s2;
stack[s]='\0';
strcat(stack,p[j].p1);
cout<<"\n$"<<stack;
j=-1;
runcount++;
if(runcount>20)
{cout<<"Conflict handled";
goto h;}
}
}
if(strcmp(stack,"E")==0&&i==strlen(input))
{
h:cout<<"\n\nAccepted";
break;
}
if(i==strlen(input))
{
cout<<"\n\nNot Accepted";
break;
}
}
```

```
return 0;
}
```

Output:

```
Enter no of productions:4
nop=4
Enter the productions as E->E*E and at end input string:
E->E+E
E->E*E
E->(E)
E->a
(a+a)*(a+a)
Input String is :
(a+a)*(a+a)
Stack

(
(a
$(E
(E+
(E+a
$(E+E
$(E
(E)
$E
E*
E*(
E*(a
$E*(E
E*(E+
E*(E+a
$E*(E+E
$E*(E
E*(E)
$E*E
$E

Accepted
```

Result: Thus, Shift Reduce Parsing implemented successfully.