PRACTICAL-6

AIM:

Implementation of Context Free Grammar.

PROGRAM CODE:

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

int i,j,k,l,m,n=0,o,p,nv,z=0,t,x=0;
char str[10],temp[20],temp2[20],temp3[20];

struct prod
{
   char lhs[10],rhs[10][10];
   int n;
}pro[10];

void findter()
{
   for(k=0;k<n;k++)
   {
      if(temp[i]==pro[k].lhs[0])
      {
         for(t=0;t<pro[k].n;t++)
         {
            for(l=0;l<20;l++)
               temp2[l]='\0';
            for(l=i+1;l<strlen(temp);l++)
               temp2[l-i-1]=temp[l];
            for(l=i;l<20;l++)
               temp[l]='\0';
            for(l=0;l<strlen(pro[k].rhs[t]);l++)
               temp[i+l]=pro[k].rhs[t][l];
            strcat(temp,temp2);
```

```
            if(str[i]==temp[i])
               return;
            else if(str[i]!=temp[i] && temp[i]>=65 && temp[i]<=90)
               break;
         }
         break;
      }
   }
   if(temp[i]>=65 && temp[i]<=90)
      findter();
}

int main()
{
   FILE *f;
//    clrscr();

   for(i=0;i<10;i++)
      pro[i].n=0;

   f=fopen("input .txt","r");
   while(!feof(f))
   {
      fscanf(f,"%s",pro[n].lhs);
      if(n>0)
      {
         if( strcmp(pro[n].lhs,pro[n-1].lhs) == 0 )
         {
            pro[n].lhs[0]='\0';
            fscanf(f,"%s",pro[n-1].rhs[pro[n-1].n]);
            pro[n-1].n++;
            continue;
         }
      }
      fscanf(f,"%s",pro[n].rhs[pro[n].n]);
      pro[n].n++;
      n++;
   }
   n--;
   printf("\nPARTH PATEL\n19DCS098\n");
   printf("\n\nTHE GRAMMAR IS AS FOLLOWS\n\n");
   for(i=0;i<n;i++)
```

```
    for(j=0;j<pro[i].n;j++)
        printf("%s -> %s\n",pro[i].lhs,pro[i].rhs[j]);

  while(1)
  {
    for(l=0;l<10;l++)
        str[0]='\0';

    printf("\n\nENTER ANY STRING ( 0 for EXIT ) : ");
    scanf("%s",str);
    if(str[0]=='0')
        printf("Exit");
//      exit(1);

    for(j=0;j<pro[0].n;j++)
    {
        for(l=0;l<20;l++)
            temp[l]='\0';
        strcpy(temp,pro[0].rhs[j]);

        m=0;
        for(i=0;i<strlen(str);i++)
        {
            if(str[i]==temp[i])
                m++;
            else if(str[i]!=temp[i] && temp[i]>=65 && temp[i]<=90)
            {
                findter();
                if(str[i]==temp[i])
                    m++;
            }
            else if( str[i]!=temp[i] && (temp[i]<65 || temp[i]>90) )
                break;
        }

        if(m==strlen(str) && strlen(str)==strlen(temp))
    {
    printf("\n\nTHE STRING can be PARSED !!!");
    break;
    }
  }
```

```
    if(j==pro[0].n)
       printf("\n\nTHE STRING can NOT be PARSED !!!");
    }

    getch();

}
```

OUTPUT:

```
PARTH PATEL
19DCS098


THE GRAMMAR IS AS FOLLOWS

S -> aBaA
S -> AB
A -> Bc
B -> c


ENTER ANY STRING ( 0 for EXIT ) : ccccc


THE STRING can NOT be PARSED !!!

ENTER ANY STRING ( 0 for EXIT ) : ccc


THE STRING can be PARSED !!!

ENTER ANY STRING ( 0 for EXIT ) : abcde


THE STRING can NOT be PARSED !!!
```