

19. SLR parser program

```
#include<stdio.h>

#include<string.h>

int axn[][6][2]={

    {{100,5},{-1,-1},{-1,-1},{100,4},{-1,-1},{-1,-1}},

    {{-1,-1},{100,6},{-1,-1},{-1,-1},{-1,-1},{102,102}},

    {{-1,-1},{101,2},{100,7},{-1,-1},{101,2},{101,2}},

    {{-1,-1},{101,4},{101,4},{-1,-1},{101,4},{101,4}},

    {{100,5},{-1,-1},{-1,-1},{100,4},{-1,-1},{-1,-1}},

    {{-1,-1},{101,6},{101,6},{-1,-1},{101,6},{101,6}},

    {{100,5},{-1,-1},{-1,-1},{100,4},{-1,-1},{-1,-1}},

    {{100,5},{-1,-1},{-1,-1},{100,4},{-1,-1},{-1,-1}},

    {{-1,-1},{100,6},{-1,-1},{-1,-1},{100,1},{-1,-1}},

    {{-1,-1},{101,1},{100,7},{-1,-1},{101,1},{101,1}},

    {{-1,-1},{101,3},{101,3},{-1,-1},{101,3},{101,3}},

    {{-1,-1},{101,5},{101,5},{-1,-1},{101,5},{101,5}}

};//Axn Table

int gotot[12][3]={1,2,3,-1,-1,-1,-1,-1,-1,-1,-1,8,2,3,-1,-1,-1,

    -1,9,3,-1,-1,10,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1}; //GoTo table

int a[10];

char b[10];

int top=-1,btop=-1,i;
```

```
void push(int k)
```

```
{
```

```
    if(top<9)
```

```
        a[++top]=k;
```

```
}
```

```
void pushb(char k)
```

```
{
```

```
    if(btop<9)
```

```
        b[++btop]=k;
```

```
}
```

```
char TOS()
```

```
{
```

```
    return a[top];
```

```
}
```

```
void pop()
```

```
{
```

```
    if(top>=0)
```

```
        top--;
```

```
}
```

```
void popb()
```

```
{
```

```
    if(btop>=0)
```

```
        b[btop--]='\0';
```

```
}
```

```
void display()
```

```
{
```

```
    for(i=0;i<=top;i++)
```

```
        printf("%d%c",a[i],b[i]);
```

```
}
```

```
void display1(char p[],int m) //Displays The Present Input String
```

```
{
```

```
    int l;
```

```
    printf("\t\t");
```

```
    for(l=m;p[l]!='\0';l++)
```

```
        printf("%c",p[l]);
```

```
    printf("\n");
```

```
}
```

```
void error()
```

```
{  
  
    printf("Syntax Error");  
  
}
```

```
void reduce(int p)
```

```
{  
  
    int len,k,ad;  
  
    char src,*dest;  
  
    switch(p)  
  
    {  
  
case 1:dest="E+T";  
  
        src='E';  
  
        break;  
  
case 2:dest="T";  
  
        src='E';  
  
        break;  
  
case 3:dest="T*F";  
  
        src='T';  
  
        break;  
  
case 4:dest="F";  
  
        src='T';  
  
        break;
```

```
case 5:dest="(E)";
```

```
    src='F';
```

```
    break;
```

```
case 6:dest="i";
```

```
    src='F';
```

```
    break;
```

```
default:dest="\0";
```

```
src='\0';
```

```
break;
```

```
}
```

```
for(k=0;k<strlen(dest);k++)
```

```
{
```

```
    pop();
```

```
    popb();
```

```
}
```

```
pushb(src);
```

```
switch(src)
```

```
{
```

```
case 'E':ad=0;
```

```
    break;
```

```
case 'T':ad=1;
```

```
    break;
```

```
case 'F':ad=2;
```

```

        break;

default: ad=-1;

        break;

    }

    push(gotot[TOS()][ad]);

}

int main()

{

    int j,st,ic;

    char ip[20]="\0",an;

    // clrscr();

    printf("Enter any String\n");

+

    scanf("%s",ip);

    push(0);

    display();

    printf("\t%s\n",ip);

    for(j=0;ip[j]!='\0';)

    {

        st=TOS();

        an=ip[j];

        if(an>='a'&&an<='z') ic=0;

```

```
else if(an=='+') ic=1;

else if(an=='*') ic=2;

else if(an=='()') ic=3;

else if(an=='') ic=4;

else if(an=='$') ic=5;

else {

    error();

    break;

}

if(axn[st][ic][0]==100)

{

    pushb(an);

    push(axn[st][ic][1]);

    display();

    j++;

    display1(ip,j);

}

if(axn[st][ic][0]==101)

{

    reduce(axn[st][ic][1]);

    display();

    display1(ip,j);

}
```

```

if(axn[st][ic][1]==102)

{

printf("Given String is accepted \n");

// getch();

break;

}

/* else

{

printf("Given String is rejected \n");

break;

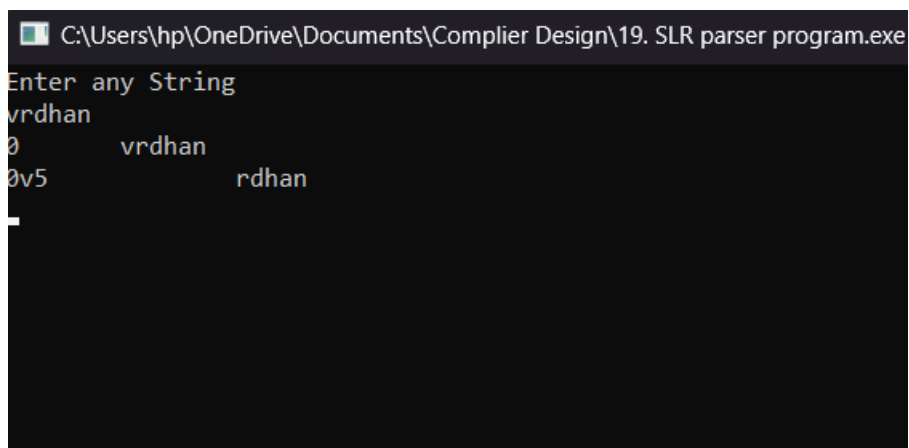
}*/

}

return 0;

}

```



```

C:\Users\hp\OneDrive\Documents\Compiler Design\19. SLR parser program.exe
Enter any String
vrdhan
0      vrdhan
0v5      rdhan

```