DATE 04-01-2022

EXPERIMENT 11

AIM

Construct Shift Reduce Parser for given language.

PROGRAM

```
#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;
void main() {
    printf("Grammar
                E → E+E   E → E*E   E → (E)   E → id \n");

    printf("Enter string");

    strcpy(act, "shift");
    printf("\nstack\tinput\taction");

    for(k=0, i=0; j<c; k++, i++, j++) {
        if(a[j]=='i' && a[j+1]=='d')
            stk[i]=a[i];
            stk[i+1]=a[j+1];
            stk[i+2]='\0';
            a[j]=' ';
            a[j+1]=' ';
            printf("\n%s\t%s\t%s", stk, a, act);
    }
    else {
        stk[i]=a[j];
        stk[i+1]='\0';
        a[j]=' ';
```

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

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 ("%s\t%s\t%s", stk, a, ac);
            j++;
        }
    // Replicate the same for E+E, E*E and (E)
```

OUTPUT

Grammar is $E \to E+E$

$E \to E * E$

$E \to (E)$

$E \to id$

Enter input string: $i+i*j+(10*k)$

Stack	input	action.
$	$i+i*j+(10+k)$	shift symbol
$i	$+i*j+(10+k)$	"
$i+	$i*j+(10+k)$	
⋮	⋮	
		Shift symbol
$i+i*j+(10+k)	$	

(Hence accepted).