

# SHIFT REDUCE

```
#include<stdio.h>

#include<stdlib.h>

#include<string.h>

char ip[15],stack[30];

int iptr=0,sptr=0,len,i;

char temp[5];

char act[15];


void check()

{

int flag=0;

while(1)

{

if(stack[sptr]=='d' && stack[sptr-1]=='i')

{

stack[sptr-1]='F';

stack[sptr]='\0';

sptr--;

flag=1;

printf("\n%s\t\t%s\t\tF->id",stack,ip);

}

if(stack[sptr]=='') && stack[sptr-1]=='E' && stack[sptr-2]=='(')

{

stack[sptr-2]='F';

stack[sptr-1]='\0';

flag=1;

sptr=sptr-2;

printf("\n%s\t\t%s\t\tF->id",stack,ip);

}
```

```

if(stack[sptr]=='F' && stack[sptr-1]=='*' && stack[sptr-2]=='T')
{
stack[sptr-1]='\0';
sptr=sptr-2;
flag=1;
printf("\n${s}\t\t${s}\t\tT->T*F",stack,ip);
}
else
if(stack[sptr]=='F')
{
stack[sptr]='T';
flag=1;
printf("\n${s}\t\t${s}\t\tT->F",stack,ip);
}
if(stack[sptr]=='T' && stack[sptr-1]=='+' && stack[sptr-2]=='E' && ip[iptr]!='*')
{
stack[sptr-1]='\0';
sptr=sptr-2;
flag=1;
printf("\n${s}\t\t${s}\t\tE->E+T",stack,ip);
}
else
if((stack[sptr]=='T' && ip[iptr]=='+') || (stack[0]=='T' && ip[iptr]=='\0') || (stack[sptr]=='T' &&
ip[iptr]==''))
{
stack[sptr]='E';
flag=1;
printf("\n${s}\t\t${s}\t\tE->T",stack,ip);
}
if((stack[sptr]=='T' && ip[iptr]=='*') ||
(stack[sptr]=='E' && ip[iptr]=='')) ||

```

```

(stack[sptr]=='E' && ip[iptr]=='+') ||
(stack[sptr]=='+' && ip[iptr]=='i' && ip[iptr+1]=='d') ||
(stack[sptr]=='(' && ip[iptr]=='i' && ip[iptr+1]=='d') ||
(stack[sptr]=='(' && ip[iptr]=='(') ||
(stack[sptr]=='*' && ip[iptr]=='i' && ip[iptr+1]=='d') ||
(stack[sptr]=='*' && ip[iptr]=='(') ||
(stack[sptr]=='+' && ip[iptr]=='(')
)
{
flag=2;
}
if(!strcmp(stack,"E") && ip[iptr]=='\0'){
printf("\n%s\t\t%s\t\tACCEPT",stack,ip);
exit(0);
}
if(flag==0){
printf("\n%s\t\t%s\t\tREJECT",stack,ip);
exit(0);
}
if(flag==2)
return;
flag=0;
}
}

```

```

void main(){
printf("\n\t\tSHIFT REDUCE PARSER\n");
printf("\nGRAMMER\n");
printf("\nE->E+T | T\nT->T*F | F");
printf("\nF->(E) | id\n");
printf("Enter the input string:");

```

```

scanf("%s",ip);

printf("\n\tStack Implementation Table\n");

printf("\nStack\t\tInput\t\tAction");

printf("\n_____ \t\t_____ \t\t_____ \n");

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

strcpy(act,"Shift");

if(ip[iptr]==''){

temp[0]=ip[iptr];

temp[1]='\0';

}

else{

temp[0]=ip[iptr];

temp[1]=ip[iptr+1];

temp[2]='\0';

}

strcat(act,temp);

len=strlen(ip);

for(i=0;i<=len-1;i++){

if(ip[iptr]=='i' && ip[iptr+1]=='d'){

stack[sptr]=ip[iptr];

sptr++;

ip[iptr]=' ';

iptr++;

stack[sptr]=ip[iptr];

stack[sptr+1]='\0';

ip[iptr]=' ';

iptr++;

}

else{

stack[sptr]=ip[iptr];

stack[sptr+1]='\0';

```

```
ip[iptr]=' ';
iptr++;
}
printf("\n$s%s\t\t%s$\t\t%s",stack,ip,act);
strcpy(act,"Shift");
if(ip[iptr]=='(' || ip[iptr]=='*' || ip[iptr]=='+' || ip[iptr]==''){
temp[0]=ip[iptr];
temp[1]='\0';
}
else{
temp[0]=ip[iptr];
temp[1]=ip[iptr+1];
temp[2]='\0';
}
strcat(act,temp);
check();
sptr++;
}
sptr++;
check();
}
```