```c
// OPERATOR PRECEDENCE PARSER

#include<stdio.h>
#include<string.h>
void main()
{
        char stack[20],ip[20],opt[10][10][1],ter[10];
        int i,j,k,n,top=0,row,col;
        for(i=0;i<10;i++)
        {
                stack[i]=0;
                ip[i]=0;
                for(j=0;j<10;j++)
                {
                        opt[i][j][1]=0;
                }
        }
        printf("Enter the no.of terminals:");
        scanf("%d",&n);
        printf("\nEnter the terminals:");
        scanf("%s",ter);
        printf("\nEnter the table values:\n");
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        printf("Enter the value for %c %c:",ter[i],ter[j]);
                        scanf("%s",opt[i][j]);
                }
        }
        printf("\nOPERATOR PRECEDENCE TABLE:\n");
        for(i=0;i<n;i++)
        {
                printf("\t%c",ter[i]);
        }
        printf("\n ");
        printf("\n");
        for(i=0;i<n;i++)
        {
                printf("\n%c |",ter[i]);
                for(j=0;j<n;j++)
                {
                        printf("\t%c",opt[i][j][0]);
                }
        }
}
```

```c
stack[top]='$';
printf("\n\nEnter the input string(append with $):");
scanf("%s",ip);
i=0;
printf("\nSTACK\t\t\tINPUT STRING\t\t\tACTION\n");
printf("\n%s\t\t\t%s\t\t\t",stack,ip);
while(i<=strlen(ip))
{
        for(k=0;k<n;k++)
        {
                if(stack[top]==ter[k]) row=k;
                if(ip[i]==ter[k]) col=k;
        }
        if((stack[top]=='$')&&(ip[i]=='$'))
        {
                printf("String is ACCEPTED"); break;
        }
        else if((opt[row][col][0]=='<') ||(opt[row][col][0]=='='))
        {
                stack[++top]=opt[row][col][0];
                stack[++top]=ip[i];
                ip[i]=' ';
                printf("Shift %c",ip[i]);
                i++;
        }
        else
        {
                if(opt[row][col][0]=='>')
                {
                        while(stack[top]!='<')
                        {
                                --top;
                        }
                        top=top-1; printf("Reduce");
                }
                else
                {
                        printf("\nString is not accepted"); break;
                }
        }
        printf("\n"); printf("%s\t\t\t%s\t\t\t",stack,ip);
}
getchar();
}
```