nterpreter and REPL
and deploy C online from your browser

Save

Run

main.c

```c
#include <stdio.h>

# define MAX 100
char stack[MAX];
int top=-1;

void push(char ch)
{
    if (top==MAX-1)
        printf("Stack is full\n");
    else
    {
        top++;
        stack[top]=ch;
    }
}
char pop()
{
    char item;
    if (top==-1)
        printf("\n stack is empty !");
    else
    {
        item=stack[top];
        top--;
        return item;
    }

```

main.c

```
24          item=stack[top];
25          top--;
26          return item;
27      }
28
29  }
30
31  int stackempty()
32  {
33      if(top==-1)
34       return 1;
35      else
36      return 0;
37  }
38
39  char stacktop()
40  {
41      if( top==-1)
42          printf("\n stack is empty!");
43      else
44          return stack[top];
45  }
46  int priority(char ch)
47  {
48      switch(ch)
49      {
50          case '+':
51          case '-':return (1);
```

```
in.c
        case '+':
        case '-':return (1);
        case '*':
        case '/':return (2);
        case '^': return (3);
        default : return (0);
    }
}

int main(int argc, char **argv)
{
  char infix[100];
    int i, item;
    printf("Enter the infix expression :");
    scanf("%s",infix);
    printf("Expression : %s",infix);
    printf("\n Postfix: ");
    i=0;
     while (infix[i]!='\0')
    {

        switch (infix[i])
        {
            case '(' : push(infix[i]);
                       break;
```

```
switch (infix[i])
{
    case '(': push(infix[i]);
              break;
    case ')':while(( item=pop())!='(')
               printf("%c",item);
             break;
    case '+':
    case '-':
    case '*':
    case '/':
    case '^':
             while(!stackempty() && priority(infix
             [i])<=priority(stacktop()))
             {
                  item=pop();

                  printf("%c", item);
             }

             push(infix[i]);
             break;
    default : printf("%c", infix[i]);
             break;
}
```

main.c

```
94                    break;
95              default : printf("%c", infix[i]);
96                    break;
97
98
99          }
00           i++;
01      }
02
03      while(!stackempty())
04      {
05          char item;
06          item=pop();
07          printf("%c", item);
08
09      }
10      printf("\n");
11      return 0;
12
13  }
```