**12. Write a C program to implement the application of Stack (Notations)**

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#define MAX 100

char stack[MAX];

int top = -1;

void push(char item) {

if(top >= MAX-1) {

printf("Stack Overflow\n");

} else {

stack[++top] = item;

}

}

char pop() {

if(top < 0) {

printf("Stack Underflow\n");

return -1;

} else {

return stack[top--];

}

}

int precedence(char symbol) {

switch(symbol) {

case '+':

case '-': return 1;

case '\*':

case '/': return 2;

case '^': return 3;

default: return 0;

}

}

void infixToPostfix(char infix[], char postfix[]) {

int i = 0, j = 0;

char item, x;

push('(');

strcat(infix, ")");

for(i = 0; infix[i] != '\0'; i++) {

item = infix[i];

if(item == '(') {

push(item);

}

else if(isalnum(item)) {

postfix[j++] = item;

}

else if(item == ')') {

while((x = pop()) != '(') {

postfix[j++] = x;

}

}

else {

while(precedence(stack[top]) >= precedence(item)) {

postfix[j++] = pop();

}

push(item);

}

}

postfix[j] = '\0';

}

int evaluatePostfix(char postfix[]) {

int i, op1, op2, result;

char item;

for(i = 0; postfix[i] != '\0'; i++) {

item = postfix[i];

if(isdigit(item)) {

push(item - '0');

}

else {

op2 = pop();

op1 = pop();

switch(item) {

case '+': result = op1 + op2; break;

case '-': result = op1 - op2; break;

case '\*': result = op1 \* op2; break;

case '/': result = op1 / op2; break;

}

push(result);

}

}

return pop();

}

int main() {

char infix[MAX], postfix[MAX];

int choice, result;

while(1) {

printf("\nStack Applications - Expression Notations:\n");

printf("1. Infix to Postfix Conversion\n");

printf("2. Postfix Evaluation\n");

printf("3. Exit\n");

printf("Enter choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

printf("Enter infix expression: ");

scanf("%s", infix);

infixToPostfix(infix, postfix);

printf("Postfix expression: %s\n", postfix);

break;

case 2:

printf("Enter postfix expression (single digits only): ");

scanf("%s", postfix);

result = evaluatePostfix(postfix);

printf("Result: %d\n", result);

break;

case 3:

return 0;

default:

printf("Invalid choice!\n");

}

}

}