

```
//Created By Ritwik on 24th March
//183215
//Evaluation of Postfix using a single stack(arrays)
#include <stdio.h>
#include <ctype.h>

#define MAXSTACK 100 /* for max size of stack */
#define POSTFIXSIZE 100 /* define max number of charcters in postfix expression */

int stack[MAXSTACK];
int top = -1;

void push(int item)
{
    if (top >= MAXSTACK - 1) {
        printf("stack over flow");
        return;
    }
    else {
        top = top + 1;
        stack[top] = item;
    }
}

int pop()
{
    int item;
    if (top < 0) {
        printf("stack under flow");
    }
    else {
        item = stack[top];
        top = top - 1;
        return item;
    }
}
```

```

void EvalPostfix(char postfix[])
{
    int i;
    char ch;
    int val;
    int A, B;

    /* evaluate postfix expression */
    for (i = 0; postfix[i] != '\0'; i++) {
        ch = postfix[i];
        if (isdigit(ch)) {
            push(ch - '0');
        }
        else if (ch == '+' || ch == '-' || ch == '*' || ch == '/') {
            A = pop();
            B = pop();

            switch (ch) /* ch is an operator */
            {
                case '*':
                    val = B * A;
                    break;

                case '/':
                    val = B / A;
                    break;

                case '+':
                    val = B + A;
                    break;

                case '-':
                    val = B - A;
                    break;
            }

            push(val);
        }
    }
}

```

```

    }
}
if(top==0){
    printf("\n Result of expression evaluation : %d \n", pop());
}
else printf("Invalid postfix expression.\n");
}

int main()
{
    int i;

    char postfix[POSTFIXSIZE];
    printf("ASSUMPTION: There are only four operators(*, /, +, -) in an expression and operand is single digit only.\n");

    scanf("%s", postfix);

    EvalPostfix(postfix);

    return 0;
}

```