

## Evaluation of Postfix Expression

The Postfix notation is used to represent algebraic expressions. The expressions written in postfix form are evaluated faster compared to infix notation as parenthesis are not required in postfix.

Following is algorithm for evaluation postfix expressions.

- 1) Create a stack to store operands (or values).
- 2) Scan the given expression and do following for every scanned element.
  - .....a) If the element is a number, push it into the stack
  - .....b) If the element is a operator, pop operands for the operator from stack. Evaluate the operator and push the result back to the stack
- 3) When the expression is ended, the number in the stack is the final answer

**Example:**

Let the given expression be “2 3 1 \* + 9 -“. We scan all elements one by one.

```
#include<stdio.h>

int stack[20];
int top = -1;

void push(int x)
{
    stack[++top] = x;
}

int pop()
{
    return stack[top--];
}

int main()
{
    char exp[20];
    char *e;
    int n1,n2,n3,num;
    printf("Enter the expression :: ");
    scanf("%s",exp);
    e = exp;
    while(*e != '\0')
    {
        if(isdigit(*e))
        {
            num = *e - 48;
            push(num);
        }
        else
        {
            n1 = pop();
            n2 = pop();
            switch(*e)
```

```
{
    case '+':
    {
        n3 = n1 + n2;
        break;
    }
    case '-':
    {
        n3 = n2 - n1;
        break;
    }
    case '*':
    {
        n3 = n1 * n2;
        break;
    }
    case '/':
    {
        n3 = n2 / n1;
        break;
    }
}
push(n3);
e++;
}
printf("\nThe result of expression %s = %d\n\n",exp,pop());
return 0;
}
```

**Examples:**

1.  $(2-3+4)*(5+6*7)$

2.  $(2 + 3) - 7 / 9$

3.  $53+62/*35*+$