

50	
48	3
44	
20	
10	

Size = 5

Top = -1

Push(10) push(20) push(30) push(40)

POP() ---40 top =2 PEEK() --- 11 top = 2 POP() -- 11 top = 1 push(12) top = 2 Push(13) top = 3

POP()--13 top =2 POP() --- 12 top = 1 POP() -- 20 top = 0 pop() – 10 top = -1

Push(60)

POP() pop() pop() top = 1 20 push(44) push(48)

Int S1[5], top1, S2[5],top2, S3[5], top3;

Top = -1;

Struct stack

{

Int items[5];

Int top;

};

S1, S2, S3;

Applications of Stacks:

Conversions & Evaluation of expressions

Function call mechanism

Tree Traversals

Graph Traversals

Conversion & Evaluation of expressions

Infix Expression:

Opnd1 Opr Opnd2

A + B

Postfix Expression (Polished notation):

Opnd1 Opnd2 Opr

A B +

Prefix Expression (Reversed Polished notation) :

Opr Opnd1 Opnd2

+ A B

Conversions :

Infix to Postfix and Prefix

Postfix to Infix and Prefix

Prefix to Infix and Postfix

Infix to Postfix :

1. $A + B - C * D / E$

> $A + B - CD* / E$

> $A + B - CD*E/$

> $AB+ - CD*E/$

> $AB+CD*E/-$

2. $A + (B - C) * (D / E)$

> $A + BC- * DE/$

> $A + BC- DE/*$

> $A BC- DE/*+$

3. $(A - (B + C)) * D ^ (E + F)$

> $(A - BC+) * D ^ (E+F)$

> $ABC+- * D ^ EF+$

> $ABC+- * DEF+^$

> $ABC+- DEF+^*$

$$\begin{aligned}
4. \quad & (((A + B) * C) - (D - E)) ^ (F + G) ^ H \\
> & ((AB + * C) - (D - E)) ^ (F + G) ^ H \\
> & (AB + C * - DE -) ^ (F + G) ^ H \\
> & AB + C * DE - - ^ FG + ^ H \\
> & AB + C * DE - - ^ FG + H ^ \\
> & \mathbf{AB + C * DE - - FG + ^ ^}
\end{aligned}$$

Infix to Prefix :

$$\begin{aligned}
& 1. A + B - C * D / E \\
> & A + B - * CD / E \\
> & A + B - /* CDE \\
> & +AB - /* CDE \\
> & \mathbf{- +AB /* CDE}
\end{aligned}$$

$$\begin{aligned}
2. \quad & A + (B - C) * (D / E) \\
> & A + -BC * /DE \\
> & A + *-BC/DE \\
> & +A*-BC/DE \\
> &
\end{aligned}$$

$$\begin{aligned}
& (A - (B + C)) * D ^ (E + F) \\
> & (A - +BC) * D ^ +EF \\
> & -A + BC * D ^ +EF \\
> & -A + BC * ^D +EF \\
> & \mathbf{* -A + BC ^D +EF}
\end{aligned}$$

$$. \quad (((A + B) * C) - (D - E)) ^ { (F + G) ^ H}$$

Postfix to Infix:

$$\begin{aligned} & AB+CD*E/- \quad 23+14*5/- \\ & > (A+B)CD*E/- \\ & > (A+B) (C*D) E / - \\ & > (A+B) ((C*D) / E) - \\ & > ((A+B) - ((C*D)/E)) \end{aligned}$$

Prefix:

$$\begin{aligned} & A B + C D * E / - \\ & > +AB C D * E / - \\ & > +AB *CD E / - \\ & > +AB /*CDE - \\ & > -+AB /*CDE \\ & > \end{aligned}$$

Prefix to Infix and Postfix:

Infix	Postfix
> -+AB /*CDE	
> -+AB/ (C * D) E	-+AB / CD* E
> -+AB ((C * D) / E)	-+AB CD*E/
> - (A + B) ((C *D) /E)	- AB+ CD*E/
> ((A +B) - ((C *D) /E))	AB+ CD*E/-
>	

$$(A - (B + C)) * D ^ (E + F)$$

$$ABCDE - + \$ * EF * -$$

- > ABC (D-E) + \$ * EF * -
- > A B (C + (D-E)) \$* EF * -
- > A (B \$ (C + (D-E))) * EF* -
- > (A* (B \$ (C + (D-E)))) E F* -
- > (A* (B \$ (C + (D-E)))) (E *F) -
- > (A* (B \$ (C + (D-E)))) - (E *F)

$$(A * (B \$ (C + (D-E)))) - (E * F)$$

- > (A * (B \$ (C + DE-))) - (E * F)
- > (A * (B \$ CDE-+)) - (E * F)
- > (A * BCDE-+\$) - (E * F)
- > A BCDE-+\$* - (E*F)
- > A BCDE-+\$* - EF*
- > A BCDE-+\$* EF* -

$$^{++} A - * BCD / + EF + GHI$$

- > ^{++} A - * BCD / + EF (G + H) I
- > ^{++} A - * BCD/ (E + F) (G+H) I
- > ^{++} A - * BCD ((E+F)/ (G+H)) I
- > ^{++}A- (B*C) D ((E+F)/ (G+H)) I
- > ^{++} A ((B*C) - D) ((E+F)/ (G+H)) I
- > ^{+} (A + ((B*C) - D)) ((E+F)/ (G+H)) I+

- > $^ ((A + ((B * C) - D)) + ((E + F) / (G + H))) \mid$
- > $((A + ((B * C) - D)) + ((E + F) / (G + H))) ^ \mid$

Infix to Postfix:

$A + B - C * D / E : AB+CD * E / -$

I/P	Stack	Postfix Expression
A	-----	A
+	+	A
B	+	AB
-	-	AB+
C	-	AB+C
*	- *	AB+C
D	- *	AB+C D
/	- /	AB+C D *
E	- /	AB+C D * E
	-	AB+C D * E /
	-----	AB+C D * E / -

$A + (B - C) * (D / E)$: **A BC- DE/*+**

I/P	Stack	Postfix Expression
A	-----	A
+	+	A
(+ (A
B	+ (AB
-	+ (-	AB
C	+ (-	ABC
)	+	ABC-
*	+ *	ABC-
(+ * (ABC-
D	+ * (ABC-D
/	+ * (/	ABC-D
E	+ * (/	ABC-DE
)	+ *	ABC-DE/
	-----	ABC-DE/*+
Operator	Stkprec	Inputprec
+ -	2	1
*/	4	3
^ \$	7	8
(0	9
)	----	0

$(((A + B) * C) - (D - E)) ^ (F + G) ^ H$

Operator	Stkprec	Inputprec
+ -	2	1
*/	4	3
^ \$	7	8
(0	9
)	----	0

I/P	Stack	Postfix Expression
((
(((
((((
A	(((A
+	(((+	A
B	(((+	AB
)	((AB+
*	((*	AB+
C	((*	AB+C

)	(AB+C*
-	(-	AB+C*
((- (AB+C*
D	(- (AB+C*D
-	(- (-	AB+C*D
E	(- (-	AB+C*DE
)	(-	AB+C*DE-
)	-----	AB+C*DE- -
^	^	AB+C*DE- -
(^ (AB+C*DE- -
F	^ (AB+C*DE- -F
+	^ (+	AB+C*DE- -F
G	^ (+	AB+C*DE- -FG
)	^	AB+C*DE- -FG+
^	^ ^	AB+C*DE- -FG+
H	^ ^	AB+C*DE- -FG+H
	-----	AB+C*DE- -FG+H^^

$(((A + B) * C) - (D - E)) ^ (F + G) ^ H$

AB +C* DE-- FG+H^^

Infix to Prefix

A + B - C * D / E : -+AB/*CDE

strcpy(pre, opr) /

Strcat(pre, opnd1) :/*CD

Strcat(pre, opnd2) :/*CDE

I/P	Operator Stack	Operand Stack
A	----	A
+	+	A
B	+	A B
-	-	+AB
C	-	+AB C
*	- *	+AB C
D	- *	+AB C D

/	- /	+AB *CD
E	- /	+AB *CD E
	-	+AB /*CDE
		-+AB/*CDE
Operator	Stkprec	Inputprec
+ -	2	1
*/	4	3
^ \$	7	8
(0	9
)	----	0

$A + (B - C) * (D / E)$: +A*BC/DE

I/P	Operator Stack	Operand Stack
A	-----	A
+	+	A
(+ (A
B	+ (A B
-	+ (-	A B
C	+ (-	A B C
)	+	A -BC
*	+ *	A -BC
(+ * (A -BC D
D	+ * (A -BC D
/	+ * (/	A -BC D
E	+ * (/	A -BC D E
)	+ *	A -BC /DE
	+	A *-BC/DE
		+ A *BC/DE
Operator	Stkprecd	Inputprecd
+ -	2	1
*/	4	3
^ \$	7	8
(0	9
)	----	0

$(((A + B) * C) - (D - E)) ^ (F + G) ^ H$

I/P	Operator Stack	Operand Stack
((
((

Operator	Stkprec	Inputprec
+ -	2	1
* /	4	3
^ \$	7	8
(0	9
)	----	0

Postfix to Infix AND PREFIX

AB+CD*E/- : A + B - C * D / E :

I/P	Operand Stack(Infix)	Operand Stack(prefix)
A	A	A
B	A B	A B
+	(A+B)	+AB
C	(A+B) C	+AB C
D	(A+B) C D	+AB C D
*	(A+B) (C*D)	+AB *CD
E	(A+B) (C*D) E	+AB *CD E
/	(A+B) ((C*D) /E)	+AB /*CDE
-	((A+B) - ((C*D) /E))	-+AB/*CDE

Opr : - op1: (A+B) op2: ((C*D) /E)

Infix : strcpy(inf, "(") strcat(inf, op1) strcat(inf, opr) strcat(inf, op2)
Strcat(inf, ")")

Inf: (A+B) : (C*D) : ((A+B) - ((C*D) /E))

Prefix: strcpy(pre, opr) strcat(pre, op1) strcat(pre, op2)

Prefix to Infix AND POSTFIX

$-+A*BC/DE$:

I/P	Operand Stack Infix)	Operand Stack(Postfix)
E	E	E
D	E D	E D
/	(D/E)	DE/
C	(D/E) C	DE/ C
B	(D/E) C B	DE/ C B
*	(D/E) (B*C)	DE/ BC*
A	(D/E) (B*C) A	DE/ BC* A
+	(D/E) (A+(B*C))	DE/ ABC*+
-	((A+(B*C)- (D/E))	ABC*+DE/-

Opr: + op1: A op2: (B*C)

Infix : strcpy(inf , "(") strcat(inf, op1) strcat(inf, opr) strcat(inf, op2)
 Strcat(inf, ")")

Inf: (D/E)

Postix: strcpy(po, op1) strcat(po, op2) strcat(po,opr)

Pos: DE/

Evaluation of Postfix and prefix expression: 123*+45\$- : -+1*23\$45

[illegible]

Val = c- '0' opr= - op2 = 1024 op1 =7 res =

Stkprecd(S.items[S.top]) > inputprecd (c)

```
Int stkprecd(char op) //      int Inpprecd(char op)
{ Switch( op)
{
    Case '+' :
    Case '-' : return 2 ;
    Case '*':
    Case '/': return 4 ; break;
    Case '^' :
    Case '$': return 6 ; break;
    Case '(': return 0; break;
}
}

{
    Switch( op)
    {
        Case '+' :
        case '-' : return 1 ;
        case '*':
        case '/': return 3 ;
        case '^':
        case '$': return 7 ;
        Case '(': return 9 ;
        case ')' : return 0;
```

```
Int Inprecd(char op)
{
    Case '+' :
    Case '-' : return 1 ; break;
    Case '*':
    Case '/': return 3 ; break;
    Case '^' :
    Case '$': return 8 ; break;
    Case '(': return 9 ; break;
```

```
    Case: ')': return 0;
}
}
```

```
Int Isoperand(char c)
{
    Switch( c)
    {
        Case '+':
        Case '-':
        Case '*':
        Case '/':
        Case '^':
        Case '$':
        Case '(':
        Case ')': return 0;
        Default: return 1;
    }
}
```

ABCDE - + \$ * EF * - ii) + + A - * BCD / + EF * GHI

i) ++A - * + BCD / + EF * GHI ii) AB + C – BA + C \$

i) ++A - * + BCD / + EF * GHI ii) AB + C – BA + C \$ -

<https://meet.google.com/qmh-umsn-fet> A

<https://meet.google.com/xjv-esdd-mzn> C