

### Remaining Functions Of Converting Infix To Postfix

```

void push(struct Stack *p, char x)
{
    if(p->tos == MAX-1)
    {
        printf("Stack Overflow");
        return;
    }
    p->tos++;
    p->arr[p->tos] = x;
}

char pop(struct Stack *p)
{
    char x;
    if(p->tos == -1)
    {
        printf("Stack Underflow");
        return 0;
    }
    x = p->arr[p->tos];
    p->tos--;
    return x;
}

```

```

int isoperand(char ch)
{
    if((ch>='A' && ch<='Z') || (ch>='a' && ch<='z') || (ch>='0' && ch<='9'))
        return 1;
    return 0;
}

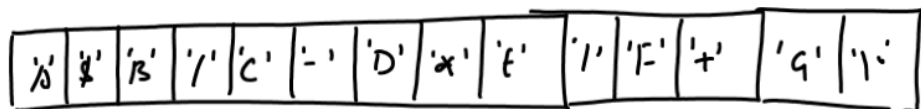
int isempty(struct Stack S)
{
    return (S.tos == -1);
}

int prcd(char op1, char op2)
{
    if(op2 == '$')
        return 0;
    else if(op1 == '$')
        return 1;
    else if(op2 == '/' || op2 == '*' || op2 == '%')
        return 0;
    else if(op1 == '/' || op1 == '*' || op1 == '%')
        return 1;
    else if(op2 == '+' || op2 == '-')
        return 0;
    else
        return 1;
}

```

+ - / \$ A B C / \* D E F G

### Converting Infix To Prefix



<u>Ele</u>	<u>Stack</u>	<u>Prefix</u>	<u>Ele</u>	<u>Stack</u>	<u>Prefix</u>
'G'		G	'-'	+', '-'	G F E D * /
'+'	+'	G	'C'	+', '-'	G F E D * / C
'F'	+', '+'	G F	'/'	+', '-', '/'	G F E D * / C
'/'	+', '/', '/'	G F	'B'	+', '-', '/', 'B'	G F E D * / C B
'E'	+', '/', 'E'	G F E	'\$'	+', '-', '/', '\$'	G F E D * / C B \$
'x'	+', '/', 'x'	G F E	'A'	+', '-', '/', '\$', 'A'	G F E D * / C B A
'D'	+', '/', 'x', 'D'	G F E D	end		G F E D * / C B A \$ /

### Algorithm For Converting Infix To Prefix

=====

1. Scan the given INFIX expression from RIGHT to LEFT one character at a time.
2. Check whether it is an OPERAND or OPERATOR.
3. If it is an OPERAND then copy it in the PREFIX array and goto step 5
4. If it is an operator then :
  - a. Check whether the Stack is empty or not.
  - b. If it is empty then PUSH the operator in the STACK and goto step 5.
  - c. If it is not empty then :
    - i. Check the precedence of OUTSIDE operator with STACKTOP
    - ii. If the precedence of OUTSIDE operator is higher or equal then PUSH it in the STACK and goto step 5
    - iii. If the precedence of STACKTOP is higher then POP it, copy it in the PREFIX array and goto step 4(a)
5. Repeat the above process until the INFIX expression ends.
6. POP the remaining operators from the STACK and copy them in PREFIX array
7. Reverse the PREFIX array
8. Finish and return