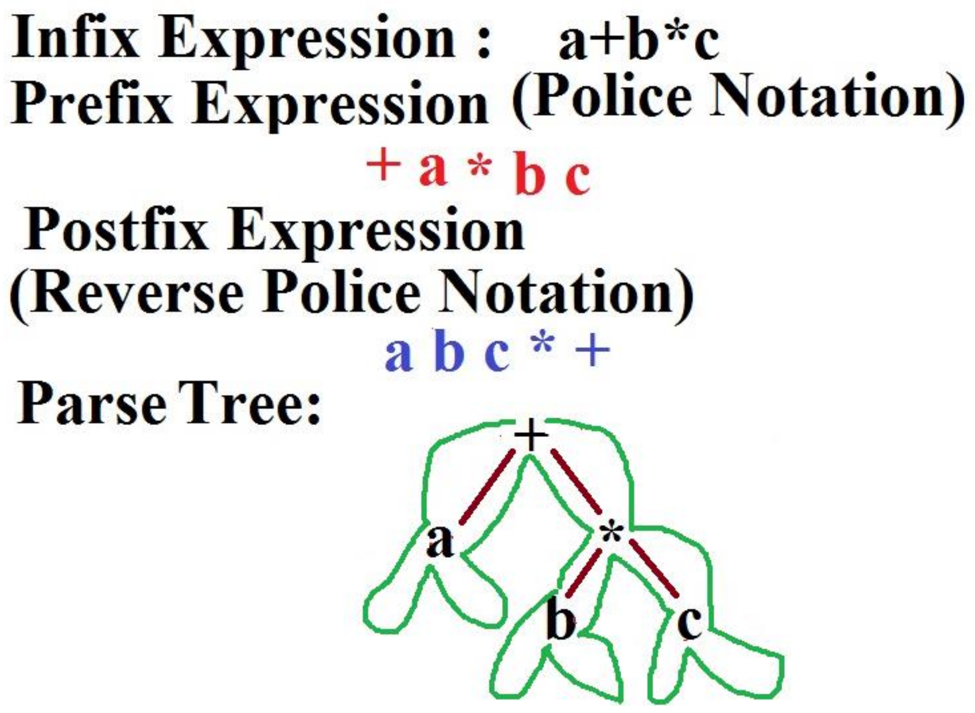
Reverse Polish Notation (RPN) or Postfix Notation Evaluation

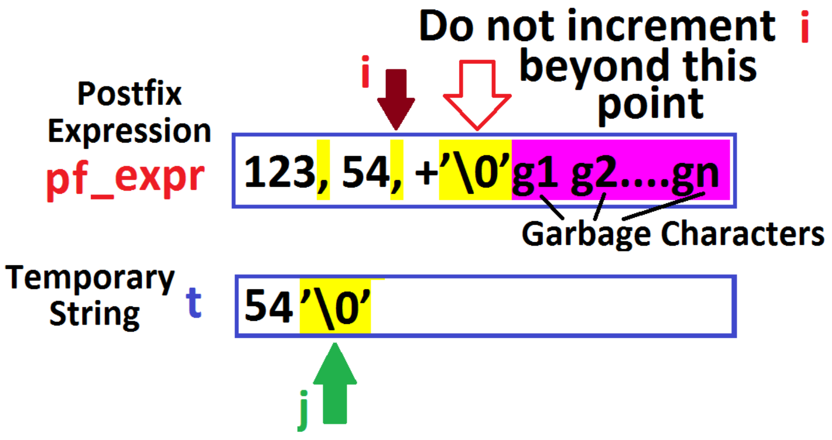
Dr. RAHUL DAS GUPTA



```
typedef  struct
{
     float   data[MAX];
     int top;
}STACK;




void initialisation(STACK *a_top)
{
    a_top->top=-1;
}

void  push(STACK *a_top, float n)
{
    if (a_top->top<=MAX-1)
            a_top->data[++(a_top->top)]=n;
    else
     {
            printf("\n Stack Overflow…");
            exit(1);
     }
}

float  pop (STACK *a_top)
{
   if(a_top->top>-1)
        return(a_top->data[(a_top->top)--]);
   else
     {
            printf("\n Empty Stack …");
            exit(1);
     }
}

int  isoperator(char c)
{
        return(c=='+'||c=='-'||c=='*'||c=='/'||c=='^');
}




/* Evaluation of a Single Operator Expression*/
float  evaluate (float  x, char operator, float  y)
{
      float v;
      switch(operator)
      {
         case  '+': v=x+y;
```

```
                  break;
         case   ‘-‘: v=x-y;
                   break;
         case  ‘*’: v=x*y;
                   break;
         case  ‘/’:  v=x/y;
                   break;
         case  ‘^’: v=pow(x,y);
                  break;
         default: printf(“\n ERROR: Operator undefined…”);
       }
   return(v);
}



/*A  string is represented by the starting address of array of character. Its data type is ‘char *’ */
float  postfix_evaluation ( char *pf_expr)
{
     STACK   S;
     int  i,j;
    /*Example of a Postfix Expression: 123, 54, +’\0’g1 g2....gn ‘\0’ */
     float  right_operand, left_operand;
     char   t[10];
     initialisation(&S);
     for(i=0; pf_expr[i]!=’\0’;) /* i: index for pf_expr */
      {
         for(j=0; pf_expr[i] !=’\0’ && pf_expr[i]!=’,’; j++,i++)
                   t[j]=pf_expr[i]; /* j: index for t (temporary string)*/
         t[j]=’\0’;  /* Terminate the temporary string t with a null character */
      /*Example of a Postfix Expression: 123, 54, +’\0’g1 g2....gn ‘\0’
        Temporary String : t=123’\0’     */

         if(!isoperator(t[0]))
                  push(&S,atof(t)); /* Pushing the numerical value of t on the top of the stack . */
         else
         {
             right_operand = pop(&S);
             left_operand = pop(&S);
             push(&S,evaluate(left_operand,t[0], right_operand));
         }
        if(pf_expr[i]!=’\0’)    /* Not to go beyond  the string end-mark ‘\0’ */
```

```
            i++;
   /* To skip the delimiter comas  ‘,’ , so that the ‘,’  will not be stored in the temporary string t.*/

     /*Example of a Postfix Expression: 123, 54, +’\0’g1 g2....gn ‘\0’
       Temporary String : t=123’\0’
    */

    }
  return(pop(&S));
}
```



A