Data Structures Lab

Experiment 9: Prefix and Postfix Evaluation

Project By:

Mohammed Rabeeh
Roll No: 35
TVE18CS036

Contents

1 Aim

Write a menu-driven program to evaluate prefix and postfix expression.

2 Problem Description

A postfix expression is one in which the operators are placed after the operands. For example,

$$A + B = AB+$$

A prefix expression is one in which the operators are placed before the operands. For example,

$$A + B = +AB$$

3 Algorithm

3.1 Postfix Evaluation

1. Reach each character from left to right until end of expression.
2. If character is operand, push to stack
3. if character is operator, pop operand 1 and operand 2 from stack and perform operand 2 operator operand 1
4. pop final result from stack.

3.2 Prefix Evaluation

Reverse the expression and do the postfix evaluation.

4 Program Code

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>
#include<stdlib.h>
int stack[20];
int top = -1;
 char *strrev( char str[])
 {
     char temp;
     int i,j;
   i = 0;
   j = strlen(str) - 1;

   while (i < j) {
      temp = str[i];
      str[i] = str[j];
      str[j] = temp;
      i++;
      j--;
   }
   return str;
 }
void push(int x)
{
       stack[++top] = x;
}

int pop()
{
       return stack[top--];
}
 int PreEval(char s[25])
{
  char temp[25];
  int i,val=0,ch1,ch2,j=0;
  i=0; top=-1;
  while (s[i]!='\0')
```

```
  {
    /*if operand is countered print it*/
    if ( (s[i]>=48 && s[i]<=57) )
    {
      j=0;
      temp[j]=s[i];
      j++;
      temp[j]='\0';
      push(atoi(temp));
    }
    else
    {
      ch2=pop();
      ch1=pop();
      switch(s[i])
      {
     case '+' :{
        val=ch2+ch1;
        break;
     }
     case '-' :{
        val=ch2-ch1;
        break;
     }
     case '*' :{
        val=ch2*ch1;
        break;
     }
     case '/' :{
        val=ch2/ch1;
        break;
     }
       }
     push(val);
    }
   i++;
  }
  val=pop();
  return val;
}
```

```
int main()
{
        int choice;
        printf("\nEnter choice:\n");
        printf("1.Evaluation of a postfix expression\n2.Evaluation
            of a prefix expression\n");
        scanf("%d",&choice);
        if(choice==1)
        {
        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());
}
else if(choice==2)
{
     char s[25],s1[25];
    int val;
    printf("enter a Prefix expression for evaluation\n");
    scanf("%s",s);
    strcpy(s1,strrev(s));
     val= PreEval(s1);
    printf("Value of Prefix Expression=%d\n", val);

}
return 0;

}
```

5 Output

```
Enter choice:
1.Evaluation of a postfix expression
2.Evaluation of a prefix expression
1
Enter the expression :: 35+

The result of expression 35+  =  8

rabeehrz@BatPC:~/Downloads$ ./a.out

Enter choice:
1.Evaluation of a postfix expression
2.Evaluation of a prefix expression
2
enter a Prefix expression for evaluation
+35
Value of Prefix Expression=8
```

6 Result

A C program to evaluate prefix and postfix expressions and the output was verified.