Date
Page

LAB 2 : INFIX to POSTFIX

Q Write a program to convert a given infix arithematic expression to postfix expression.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX 100
char st[MAX];
int top = -1;
void push (char st[], char);
char pop (char st[]);
void InfixtoPostfix (char source[], char target[]);
int getPriority (char);
int main ()
{
    char infix[100], postfix[100];
    printf ("\n Enter any infix expression : ");
    gets (infix);
    strcpy (postfix, " ");
    InfixtoPostfix (infix, postfix);
    printf ("\n The corresponding postfix expression is : ");
    puts (postfix);
    getch ();
    return 0;
}
void InfixtoPostfix (char source[], char Target[])
{
    int i=0, j=0;
    char Temp;
```

Date __/__/__
Page _____

```
strcpy(target, "");
while(source[i] != '\0')
{
    if(source[i] == '(')
    {
        push(st, source[i]);
        i++;
    }
    else if(source[i] == ')')
    {
        while((top != -1) && (st[top] != '('))
        {
            target[j] = pop(st);
            j++;
        }
        if(top == -1)
        {
            printf("\n INCORRECT EXPRESSION");
            exit(1);
        }
        temp = pop(st); // remove left paranthesis
        i++;
    }
    else if(isdigit(source[i]) || isalpha(source[i]))
    {
        target[j] = source[i]
        j++;
        i++;
    }
    else if(source[i] == '+' || s[i] == '-' || s[i] == '*' || s[i] == '/' ||
            s[i] == '%')
    {
        while((top != -1) && (st[top] != '(') &&
              (getPriority(st[top])) > getPriority(source[i]))
```

Date / /
Page

```
        if
        while
        {
            target [j] = pop (st);
            j++;
        }
        push (st, source [i]);
        i++;
    }
    else
    {
        printf ("\n INCORRECT ELEMENT IN EXPRESSION");
        exit (1);
    }
}
while ((top!=-1) && (st [top]!='('))
{
    target [j] = pop (st);
    j++;
}
target [j] = '\0';
}
int getPriority (char op)
{
    if (op=='/' || op=='*' || op=='%')
        return 1;
    else if (op=='+' || op=='-')
        return 0;
}
void push (char st[], char val)
{
    if (top==MAX-1)
        printf ("\n STACK OVERFLOW");
    else
    {
        top++;
        st [top] = val;
    }
}
```

Date __/__/__
Page _____

```
char pop (char st[])
{
    char val = ' ';
    if (top == -1)
        printf ("\n STOCK UNDERFLOW");
    else
    {
        val = st[top];
        top--;
    }
    return val;
}
```

O/p

Enter any infix expression : A+B-C*D
The corresponding postfix expression is : AB+CD*-

LINEAR QUEUE

Date __/__/__
Page _____

LAB 3(a): LINEAR QUEUE

Q WAP to simulate the working of a queue of integers using Array. Provided operations
(a) Insert
(b) delete
(c) display
Also, for queue empty & overflow cond. message

```
# include <stdio.h>
# define MAX 10
int queue [MAX];
int front = -1, rear = -1;
void insert (void);
int delete (void);
void display (void);
int main ()
{
    int option, val;
    do
    {
        printf ("\n***** MAIN MENU *****");
        printf ("\n 1. Insert an element");
        printf ("\n 2. Delete an element");
        printf ("\n 3. Display the queue");
        printf ("\n 4. EXIT");
        printf ("\n Enter your option:");
        scanf ("%d", &option);

        switch (option)
        {
        case 1: insert ();
        break;
```

Date __/__/__
Page _____

do
{
switch(
case 2: val = delete();
if (val != -1)
printf("\n The number deleted is: %d", val);
break;) ×
case 2: delete()
break;
case 3: display();
break;
}
} while (option != 4);
getch();
return 0;
}
void insert()
{
int num;
printf("\n Enter the number to be inserted in the queue: ");
scanf("%d", &num);
if (rear == MAX-1)
printf("\n OVERFLOW");
else if (front == -1 && rear == -1)
front = rear = 0;
else
rear++;
queue[rear] = num;
}
int delete()
{
int val;
if (front == -1)
printf("\n UNDERFLOW");
return -1;
else printf("The number deleted is: %d", queue[front]);
front++;
}

Date __/__/__
Page _____

```
        if(front > rear)
        front = rear = -1;
        return val;
    }
}
void display()
{
    int i;
    printf("\n");
    if(front == -1 || front > rear)
    printf("\n Queue is EMPTY");
    else
    {
        for(i = front; i <= rear; i++)
        printf("\t %d", queue[i]);
    }
}
```

O/P

```
***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Display the queue
4. EXIT
Enter your option: 1
Enter the no. to be inserted in the queue: 90

***** MAIN MENU *****
1. Insert an element
2. Delete an element
3. Display an element
4. EXIT
enter your option: 2
The number deleted is: 90
```

Date __/__/__
Page _____

LAB(5b) CIRCULAR QUEUE

Q WAP to simulate the working of a queue of integer using Array. Provided operations:
(a) Insert
(b) delete
(c) display.
Also, for queue empty & overflow send message.

```
#include <stdio.h>
#define MAX 10
int queue[MAX];
int front=-1, rear=-1;
void insert (void);
int delete (void);
void display (void);
int main()
{
    int option, val;
    do
    {
        printf("MAIN MENU:\n 1. Insert\n 2. Delete\n
            3. Display\n 4. Exit\ Enter your option:");
        scanf("%d", &option);
        switch (option)
        {
            case 1: insert()
                    break;
            case 2: val= delete();
                    if (val!=-1)
                        printf("The number deleted is:%d", val);
                    break;
            case 3: display();
                    break;
        }
    } while (option!=4);
    getch();
    return 0;
}
```

Date __/__/__
Page _____

```
void insert()
{
    int num;
    printf("Enter the number to be inserted: ");
    scanf("%d", &num);
    if (front == 0 && rear == MAX-1)
        printf("Overflow");
    else if (front == -1 && rear == -1)
        front = rear = 0
        queue[rear] = num;
    else if (front != 0 && rear == MAX-1)
        rear = 0;
        queue[rear] = num;
    else
        rear++;
        queue[rear] = num;
}
int delete()
{
    int val;
    if (front == -1 && rear == -1)
    {
        printf("UNDERFLOW");
        return -1;
    }
    val = queue[front];
    if (front == rear)
        front = rear = -1;
    else
    {
        if (front == MAX-1)
            front = 0;
        else
            front++;
    }
    return val;
}
```

Date __/__/__
Page _____

```
void display()
{
    int i;
    printf("\n");
    if (front == -1 && rear == -1)
        printf("Queue is empty");
    else
    {
        if (front < rear)
        {
            for(i=front; i<=rear; i++)
                printf("%d", queue[i]);
        }
        else
        {
            for (i=front; i<MAX; i++)
                printf("%d", queue[i]);
            for (i=front; i<=rear; i++)
                printf("%d", queue[i]);
        }
    }
}
```

O/P

MAIN MENU
1. Insert
2. Delete
3. Display
4. EXIT

Enter your option : 1
Enter the element which is to be inserted : 90

MAIN MENU
1. Insert
2. Delete
3. Display
4. EXIT

Enter your option : 2
The number deleted is : 90

MAIN MENU
1. Insert
2. Delete
3. display
4. EXIT

Enter your option : 3
Queue is Empty.

01/01/2024