

BHAGWAN PARSHURAM INSTITUTE OF TECHNOLOGY

INDEX

Subject PPL. Branch CSE Sem 8

| S. No. | Objective Of Experiment | Date | Faculty Signature |
|--------|--|------|-------------------|
| 1 | Implement all major function of String.h in single C program using switch case 0 to select specific function from user | | 1/2 |
| 2 | Write a program in C to reverse a linked list Iterative & recursive. | | 1/2 |
| 3 | Write a program in C to implement tower of Hanoi | | 1/2 |
| 4 | Write a program in C++ to count the no of object of class using static data member functions & constructor. | | 1/2 |
| 5 | WAP in C++ & java to deduce a clos time with data member millis & hh. | | 1/2 |
| 6 | WAP in C++ to define a class complex to represent set of all complex numbers | | 1/2 |
| 7 | WAP to prepare a list of 10 question and their answer | | 1/2 |
| 8 | WAP to display questions at random out of 10 of expt 7. | | 1/2 |
| 9 | WAP in C & C++ to convert postfix to infix expression & infix to postfix. | | 1/2 |

Experiment-1

Q : Implement all major functions of string.h in single C program using switch case to select specific function from user choice(strlen,strcpy,strcmp,strrev,substring)

```
#include<stdio.h>

int stringLen(char str1[20]){
    int count=0,i;
    for(i=0;str1[i]!='\0';i++){
        count++;
    }
    return count;
}

void stringCopy(char str1[20]){
    char str2[20];
    int len,i;
    len=stringLen(str1);
    for(i=0;i<len;i++){
        str2[i]=str1[i];
    }
    printf("\nString Copied:");
    for(i=0;i<len;i++){
        printf("%c",str2[i]);
    }
}

void stringRev(char str1[20]){
    char str2[20];
    int len,i,j;
    len=stringLen(str1);
    for(i=0,j=len-1;j>=0;i++,j--){
        str2[i]=str1[j];
    }
    printf("\nReverse String:");
    for(i=0;i<len;i++){
        printf("%c",str2[i]);
    }
}

void stringComp(char str1[20],char str2[20]){
    int i,temp=0;
    for(i=0; str1[i]!='\0'; i++)
    {
        if(str1[i] == str2[i])
            temp = 1;
```

```

        else
            temp = 0;
    }
    if(temp==1){
        printf("\nStrings are equal");
    }else{
        printf("\nStrings are not equal");
    }
}

void stringSub(int pos,int len,char str1[20]){
    char str[20];
    int i,j=0;
    for(i=pos-1;i<len;i++,j++){
        str[j]=str1[i];
    }
    printf("\nSubstring:");
    for(i=0;i<len;i++){
        printf("%c",str[i]);
    }
}
void main(){
    int n,i,pos,len;
    char str1[20],str2[20];
    printf("Enter string:");
    gets(str1);
    printf("\nEnter second string:");
    gets(str2);
    printf("\nFunctions:\n1.String length\n2.Copy\n3.Reverse\n4.Compare
    Strings\n5.Substring\n");
    for(i=0;i<5;i++){
        printf("\n\nSelect function:");
        scanf("%d",&n);

        switch(n){
            case 1: n=stringLen(str1);
                printf("\nLength of string is:%d",n);
                break;
            case 2: stringCopy(str1);
                break;
            case 3: stringRev(str1);
                break;
            case 4: stringComp(str1,str2);
                break;
            case 5: printf("Enter position and length of substring:");
                scanf("%d%d",&pos,&len);
        }
    }
}

```

```
    stringSub(pos,len,str1);
    break;
}
```

Output

```
C:\Users\admin\Desktop\stringfunc.exe
Enter string:teacher
Enter second string:ppl
Functions:
1.String length
2.Copy
3.Reverse
4.Compare Strings
5.Substring

Select function:1
Length of string is:7
Select function:2
String Copied:teacher
Select function:3
Reverse String:rehaeat
Select function:4
Strings are not equal
Select function:5
Enter position and length of substring:1 4
Substring:teac
Process returned 4 (0x4) execution time : 49.998 s
Press any key to continue.
```

Experiment-2

Q : Write a program in C to reverse a linked list iterative and recursive.

```
#include<stdio.h>

struct Node{
    int data;
    struct Node *next;
};

struct Node *head=NULL;

struct Node* newNode(int data){
    struct Node* temp=(struct Node*)malloc(sizeof(struct Node));
    temp->data=data;
    temp->next=NULL;
    return temp;
}

void insertData(int data){

    if(head==NULL){
        head=newNode(data);
    }else{
        struct Node *temp=head;
        while(head->next!=NULL){
            head=head->next;
        }
        head->next=newNode(data);
        head=temp;
    }
}

void printLL(){

    printf("\nLinked List:");
    struct Node *temp=head;
    while(temp->next!=NULL){
        printf("%d\n",temp->data);
        temp=temp->next;
    }
    printf("%d",temp->data);
}

void reverseLL_iter(){

    struct Node *p=NULL,*q;
}
```

```
struct Node *temp=head;
while(temp!=NULL){
    q=temp->next;
    temp->next=p;
    p=temp;
    temp=q;
}
head=p;
}

void reverseLL_recurse(struct Node *temp, struct Node *p) {

if(temp->next==NULL) {
    temp->next=p;
    head=temp;
    return;
}
struct Node *q=temp->next;
temp->next=p;

reverseLL_recurse(q,temp);
}

void main() {
    int n;
    char ch;

do{
    printf("Enter the data:");
    scanf("%d",&n);
    insertData(n);
    printf("\nDo you want to enter more data?");
    scanf(" %c",&ch);

} while(ch=='Y'||ch=='y');

printLL();
printf("\n\nIteratively Reverse:");
reverseLL_iter();
printLL();
printf("\n\nRecursively Reverse:");
reverseLL_recurse(head,NULL);
printLL();
}
```

Output

```
C:\Users\admin\Desktop\ll.exe
Enter the data:1
Do you want to enter more data?y
Enter the data:2
Do you want to enter more data?y
Enter the data:3
Do you want to enter more data?y
Enter the data:4
Do you want to enter more data?n
Linked List:1 2 3 4
Iteratively Reverse:
Linked List:4 3 2 1
Recursively Reverse:
Linked List:1 2 3 4
Process returned 1 <0x1> execution time : 16.645 s
Press any key to continue.
```

Experiment-3

Q : Write a program in C to implement Towers of Hanoi.

```
#include<stdio.h>

void towerOfHanoi(int n, char from_rod, char to_rod, char aux_rod)
{
    if(n == 1)
    {
        printf("\nMove disk 1 from rod %c to rod %c", from_rod, to_rod);
        return;
    }
    towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
    printf("\nMove disk %d from rod %c to rod %c", n, from_rod, to_rod);
    towerOfHanoi(n-1, aux_rod, to_rod, from_rod);
}

void main()
{
    int n;
    printf("Enter no. of disks:");
    scanf("%d",&n);
    printf("Rods are:A\tB\tC\n");
    towerOfHanoi(n, 'A', 'C', 'B');

}
```

Output

```
C:\Users\admin\Desktop\towerOfHanoi.exe
Enter no. of disks:3
Rods are:@      B      C
Move disk 1 from rod A to rod C
Move disk 2 from rod A to rod B
Move disk 1 from rod C to rod B
Move disk 3 from rod A to rod C
Move disk 1 from rod B to rod A
Move disk 2 from rod B to rod C
Move disk 1 from rod A to rod C
Process returned 32 <0x20>  execution time : 2.274 s
Press any key to continue.
```

Experiment-4

Q : Write a program in C++ to count the No. of Object of a class using static data member, function and constructor

```
#include<iostream>

using namespace std;

class Count{
private:
    static int counter;
public:
    Count(){
        cout<<"\n\nObject created";
        counter++;
    }
    void printObj(){
        cout<<"\nTotal No. of Objects:"<<counter;
    }
};

int Count :: counter = 0;

int main(){
    Count obj1;
    obj1.printObj();

    Count obj2;
    obj2.printObj();

    Count obj3;
    obj3.printObj();

    return 0;
}
```

Output

```
C:\Users\admin\Desktop\objectsCount.exe

Object created
Total No. of Objects:1

Object created
Total No. of Objects:2

Object created
Total No. of Objects:3

Process returned 0 <0x0> execution time : 0.045 s
Press any key to continue.
```

Experiment 5

Q: Write a program in C++ and Java to declare a class Time with data members mm,ss and hh. Define parametrize constructor to assign time to its objects. Add two time objects using member function and assign to third object.

C++ Program:

```
#include<iostream>
#include<conio.h>

using namespace std;

class Time{
public:
    int hh;
    int mm;
    int ss;
    Time(int hr,int minu,int sec){
        hh=hr;
        mm=minu;
        ss=sec;
    }
};

void add(Time t1,Time t2){
    Time t3(0,0,0);
    t3.hh=t1.hh+t2.hh;
    t3.mm=t1.mm+t2.mm;
    t3.ss=t1.ss+t2.ss;

    if(t3.ss >= 60)
    {
        t3.mm+=t3.ss / 60;
        t3.ss = t3.ss % 60;
    }
    if(t3.mm >= 60)
    {
        t3.hh+=t3.mm / 60;
        t3.mm = t3.mm % 60;
    }
    cout<<"\nAfter Addition of two time objects:\n"<<t3.hh<<":"<<t3.mm<<":"<<t3.ss;
}

int main(){
    int h1,m1,s1;
    int h2,m2,s2;
    char ch;
```

```
do{
    cout<<"\nEnter time1:";
    cin>>h1>>m1>>s1;
    cout<<"\nEnter time2:";
    cin>>h2>>m2>>s2;

    Time t1(h1,m1,s1);
    Time t2(h2,m2,s2);
    add(t1,t2);

    cout<<"\nDo you want to continue?";
    cin>>ch;
} while(ch=='y'||ch=='Y');

return 0;
}
```

Output

```
C:\Users\admin\Downloads\TIME.exe

Enter time1:1 20 20
Enter time2:1 20 20
After Addition of two time objects:
2:40:40
Do you want to continue?y

Enter time1:1 40 20
Enter time2:1 20 20
After Addition of two time objects:
3:0:40
Do you want to continue?y

Enter time1:1 40 40
Enter time2:1 10 20
After Addition of two time objects:
2:51:0
Do you want to continue?n

Process returned 0 <0x0> execution time : 22.742 s
Press any key to continue.
```

Java Program:

```
import java.util.Scanner;

public class Time {
    int hh;
    int mm;
    int ss;
    Time(int hr,int minu,int sec){
        hh=hr;
        mm=minu;
        ss=sec;
    }
    void add(Time t2){
        Time t3= new Time(0,0,0);
        t3.hh=hh+t2.hh;
        t3.mm=mm+t2.mm;
        t3.ss=ss+t2.ss;

        if(t3.ss >= 60)
        {
            t3.mm+= t3.ss / 60;
            t3.ss = t3.ss % 60;
        }
        if(t3.mm >= 60)
        {
            t3.hh+= t3.mm / 60;
            t3.mm = t3.mm % 60;
        }
        System.out.println("After Addition of two time objects:\n"+t3.hh+":"+t3.mm+":"+t3.ss);
    }

    public static void main(String[] args){
        int h1,m1,s1;
        int h2,m2,s2;
        String ch;
        Scanner in=new Scanner(System.in);

        do{
            System.out.println("Enter time1:");
            h1=in.nextInt();
            m1=in.nextInt();
            s1=in.nextInt();

            System.out.println("Enter time2:");
            h2=in.nextInt();
            m2=in.nextInt();
            s2=in.nextInt();

            Time t1=new Time(h1,m1,s1);
            Time t2=new Time(h2,m2,s2);
            t1.add(t2);
        }while(ch.equalsIgnoreCase("y"));
    }
}
```

```
    h2=in.nextInt();
    m2=in.nextInt();
    s2=in.nextInt();
    Time t1=new Time(h1,m1,s1);
    Time t2=new Time(h2,m2,s2);
    t1.add(t2);
    System.out.println("Do you want to continue?");
    ch=in.next();
}while(ch.equals("y")||ch.equals("Y"));
}
}
```

Output

```
run:  
Enter time1:  
1 20 20  
Enter time2:  
1 20 20  
After Addition of two time objects:  
2:40:40  
Do you want to continue?  
y  
Enter time1:  
1 40 20  
Enter time2:  
1 20 20  
After Addition of two time objects:  
3:0:40  
Do you want to continue?  
y  
Enter time1:  
2 40 40  
Enter time2:  
1 20 20  
After Addition of two time objects:  
4:1:0  
Do you want to continue?  
n  
BUILD SUCCESSFUL (total time: 32 seconds)
```

Experiment-6

Q : Write a program in C++ to define a class Complex to represent set of all complex numbers. Overload ‘+’ operator to add complex numbers using member function and overload ‘*’ operator to multiply two complex numbers using friend function of class Complex.

```
#include<iostream>
#include<conio.h>

using namespace std;

class Complex {
public:
    int real;
    int imag;

    Complex(int r,int i){
        real=r;
        imag=i;
    }

    Complex operator+(Complex c2){
        Complex c3(0,0);
        c3.real=real+c2.real;
        c3.imag=imag+c2.imag;
        return c3;
    }

    friend Complex operator*(Complex c2);
};

Complex operator*(Complex c1,Complex c2){
    Complex c3(0,0);
    c3.real=(c1.real*c2.real)-(c1.imag*c2.imag);
    c3.imag=(c1.real*c2.imag)+(c1.imag*c2.real);
    return c3;
}

int main(){
    int real1,real2;
    int imag1,imag2;
    char ch;
    do{
        cout<<"\nEnter real and imaginary part(1):";
        cin>>real1>>imag1;
        cout<<"\nEnter real and imaginary part(2):";
        cin>>real2>>imag2;
        c1=Complex(real1,imag1);
        c2=Complex(real2,imag2);
        c3=c1+c2;
        cout<<"\nSum = ";
        cout<<c3.real<<","<<c3.imag;
    }while(ch=='y');
}
```

```
cin>>real1>>imag1;
cout<<"\nEnter real and imaginary part(2):";
cin>>real2>>imag2;

Complex c1(real1,imag1);
Complex c2(real2,imag2);
Complex c3=c1+c2;
cout<<"\n\nAfter adding two complex objects: "<<c3.real<<"+"<<c3.imag;

c3=c1*c2;
cout<<"\nAfter multiplying two complex objects: "<<c3.real<<"+"<<c3.imag;

cout<<"\n\nDo you want to continue? ";
cin>>ch;
}while(ch=='Y'||ch=='y');

return 0;
}
```

Output

```
C:\Users\admin\Desktop\Complex.exe

Enter real and imaginary part(1):1 2
Enter real and imaginary part(2):3 4

After adding two complex objects: 4+ i 6
After multiplying two complex objects: -5+ i 10

Do you want to continue?n

Process returned 0 (0x0) execution time : 6.457 s
Press any key to continue.
```

Experiment-7

Q : Write a program to prepare a list of 10 questions and their answers.

```
#include <iostream>
#include <fstream>
using namespace std;

int main () {
    fstream file;
    int c=0;
    char ch;
    char str[100];
    do{
        file.open("Q.txt", ios::out|ios::app);
        c++;
        cout<<"\nEnter Qtion"<<c<<";";
        cin.getline(str,1000);
        file <<c<<". "<<str<<"\n";

        cout<<"\nIts answer:";
        cin.getline(str,1000);
        file <<str<<"\n";

        file.close();

        cout<<"Do you want to continue?";
        cin>>ch;
        cin.ignore();
    }while(ch=='y'||ch=='Y');

    cout<<"\n\nQtions list is successfully prepared";
    return 0;
}
```

Output

C:\Users\admin\Desktop\Untitled2.exe

Enter question1:In C++ which operator is used for dynamic memory allocation?

Its answer:New

Do you want to continue?y

Enter question2:What is the only function all C++ programs must contain?

Its answer:main()

Do you want to continue?y

Enter question3:The operator used for indirection is?

Its answer:*

Do you want to continue?y

Enter question4:The size of an object or a type can be determined using which operator?

Its answer:sizeof

Do you want to continue?y

Enter question5:How many sequence of statements are present in C++?

Its answer:5

Do you want to continue?y

Enter question6:The switch statement is also called as?

Its answer:selective structure

Do you want to continue?y

Enter question7:Which keyword is used to check exception in block of code?

Its answer:try

Do you want to continue?y

Enter question8:Where does the object is created?

Its answer:class

Do you want to continue?y

Enter question9:What is the default calling convention for a compiler in C++?

Its answer:_cdecl

Do you want to continue?y

Enter question10:The constants are also called as?

Its answer:literals

Do you want to continue?n

Questions list is successfully prepared

Process returned 0 (0x0) execution time : 1076.898 s

Press any key to continue.

Experiment-8

Q : Write a program to display questions at random out of experiment 7.

```
#include<iostream>
#include<stdlib.h>
#include<conio.h>
#include<fstream>
#include<ctime>
using namespace std;

void define()
{
    ifstream in;
    char s[100],a[50],ans[50];
    int i=0,r=0,j=0;
    int temp=0;

    srand(time(0));
    for(int k=0;k<5;k++){
        j=0;
        r=rand()%10+1;

        in.open("Q.txt");

        while(j<(r*2-2)){
            in.getline(s,100);
            j++;
        }
        in.getline(s,100);
        cout<<endl<<endl<<s;

        in.getline(a,50);

        cout<<"\nYour answer:";
        cin.getline(ans,50);

        for(int i=0;a[i]!='\0';i++){
            if(tolower(ans[i])==tolower(a[i])){
                temp+=1;
            }
            else
            {
                temp=0;
                break;
            }
        }
    }
}
```

```
    }
}

if(temp==1){
    cout<<"\nCorrect answer";
}else{
    cout<<"\nWrong answer";
    cout<<"\tCorrect answer is:"<<a;
}

in.close();
}

int main()
{
    cout<<"Qtion Answer\n";
    define();

    return 0;
}
```

Output

C:\Users\admin\Desktop\Untitled1.exe

Question Answer

4.The size of an object or a type can be determined using which operator?

Your answer:sizeof

Correct answer

3.The operator used for indirection is?

Your answer:*

Correct answer

8.Where does the object is created?

Your answer:class

Correct answer

9.What is the default calling convention for a compiler in C++?

Your answer:_cdecl

Correct answer

7.Which keyword is used to check exception in block of code?

Your answer:catch

Wrong answer Correct answer is:try

Process returned 0 <0x0> execution time : 24.042 s

Press any key to continue.

Experiment-9

Q: Write a program in C and C++ to convert postfix to infix expression and infix to postfix expression.

C program to convert postfix to infix:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

#define MAX 20

char str[MAX], stack[MAX];
int top = -1;

char pop()
{
    return stack[top--];
}

void push(char ch)
{
    stack[++top] = ch;
}

void postfix_to_infix(char expression[])
{
    int count, length;
    char element, operator;
    length = strlen(expression);
    for(count = 0; count < MAX; count++)
    {
        stack[count] = 0;
    }
    printf("\nInfix Expression:");
    printf("%c", expression[0]);
    for(count = 1; count < length; count++)
    {
        if(expression[count] == '-' || expression[count] == '/' || expression[count] == '*' ||
           expression[count] == '+')
        {
            element = pop();
            operator = expression[count];
            printf(" %c %c", operator, element);
        }
        else
    }
}
```

```
        {
            push(expression[count]);
        }
    printf("%c", expression[top--]);
}

int main()
{
    char postfix_expression[50];
    printf("\nEnter Postfix Expression:");
    scanf("%s", postfix_expression);
    postfix_to_infix(postfix_expression);
    printf("\n");
    return 0;
}
```

C++ program to convert postfix to infix:

```
#include <bits/stdc++.h>

using namespace std;

bool isOperand(char x)
{
    return (x >= 'a' && x <= 'z') ||
           (x >= 'A' && x <= 'Z');
}

string getInfix(string exp)
{
    stack<string> s;

    for (int i=0; exp[i]!='\0'; i++)
    {
        if (isOperand(exp[i]))
        {
            string op(1, exp[i]);
            s.push(op);
        }

        else
        {
            string op1 = s.top();
            s.pop();
            string op2 = s.top();
            s.pop();
            s.push("(" + op2 + exp[i] +
                   op1 + ")");
        }
    }

    return s.top();
}

int main()
{
    string exp;
    cout<<"Enter Postfix Expression:";
    cin>>exp;
    exp=getInfix(exp);
    cout<<"\nInfix Expression:"<<exp<<endl; return 0;
```

Output

```
C:\Users\admin\Desktop\posttoinfix_c.exe
Enter Postfix Expression:abc+*
Infix Expression:a + c * b
Process returned 0 <0x0> execution time : 6.008 s
Press any key to continue.
```

```
C:\Users\admin\Desktop\posttoinfix_cp.exe
Enter Postfix Expression:abc++
Infix Expression:(a+(b+c))
Process returned 0 <0x0> execution time : 2.988 s
Press any key to continue.
```

C program to convert infix to postfix:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct Stack
{
    int top;
    unsigned capacity;
    int* array;
};

struct Stack* createStack( unsigned capacity )
{
    struct Stack* stack = (struct Stack*) malloc(sizeof(struct Stack));

    if (!stack)
        return NULL;

    stack->top = -1;
    stack->capacity = capacity;

    stack->array = (int*) malloc(stack->capacity * sizeof(int));

    if (!stack->array)
        return NULL;
    return stack;
}

int isEmpty(struct Stack* stack)
{
    return stack->top == -1 ;
}

char peek(struct Stack* stack)
{
    return stack->array[stack->top];
}

char pop(struct Stack* stack)
{
    if (!isEmpty(stack))
        return stack->array[stack->top--] ;
    return '$';
}

void push(struct Stack* stack, char op)
{
    stack->array[++stack->top] = op;
```

```
}

int isOperand(char ch)
{
    return (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z');
}

int Prec(char ch)
{
    switch (ch)
    {
        case '+':
        case '-':
            return 1;

        case '*':
        case '/':
            return 2;

        case '^':
            return 3;
    }
    return -1;
}

int infixToPostfix(char* exp)
{
    int i, k;

    struct Stack* stack = createStack(strlen(exp));
    if(!stack)
        return -1;

    for (i = 0, k = -1; exp[i]; ++i)
    {
        if (isOperand(exp[i]))
            exp[++k] = exp[i];

        else if (exp[i] == '(')
            push(stack, exp[i]);

        else if (exp[i] == ')')
        {
            while (!isEmpty(stack) && peek(stack) != '(')
                exp[++k] = pop(stack);
            if (!isEmpty(stack) && peek(stack) != '(')

```

```
        return -1;
    else
        pop(stack);
}
else
{
    while (!isEmpty(stack) && Prec(exp[i]) <= Prec(peek(stack)))
        exp[++k] = pop(stack);
    push(stack, exp[i]);
}

while (!isEmpty(stack))
    exp[++k] = pop(stack );

exp[++k] = '\0';
printf("\nPostfix Expression:%s", exp );
}

int main()
{
    char exp[50];
    printf("\nEnter Infix Expression:");
    scanf("%s", exp);
    infixToPostfix(exp);
    return 0;
}
```

C++ program to convert infix to postfix:

```
#include<bits/stdc++.h>
using namespace std;

int prec(char c)
{
    if(c == '^')
        return 3;
    else if(c == '*' || c == '/')
        return 2;
    else if(c == '+' || c == '-')
        return 1;
    else
        return -1;
}

void infixToPostfix(string s)
{
    std::stack<char> st;
    st.push('N');
    int l = s.length();
    string ns;
    for(int i = 0; i < l; i++)
    {
        if((s[i] >= 'a' && s[i] <= 'z')||(s[i] >= 'A' && s[i] <= 'Z'))
            ns+=s[i];
        else if(s[i] == '(')
            st.push('(');
        else if(s[i] == ')')
        {
            while(st.top() != 'N' && st.top() != '(')
            {
                char c = st.top();
                st.pop();
                ns += c;
            }
            if(st.top() == '(')
            {
                char c = st.top();
                st.pop();
            }
        }
        else{
            while(st.top() != 'N' && prec(s[i]) <= prec(st.top()))
                ns += st.top();
            st.push(s[i]);
        }
    }
    while(st.top() != 'N')
        ns += st.top();
    cout << ns;
}
```

```
{  
    char c = st.top();  
    st.pop();  
    ns += c;  
}  
st.push(s[i]);  
}  
}  
  
while(st.top() != 'N')  
{  
    char c = st.top();  
    st.pop();  
    ns += c;  
}  
cout<<"Postfix Expression:"<<ns<<endl;  
}  
  
int main()  
{  
    string exp;  
    cout<<"Enter Infix Expression:";  
    cin>>exp;  
    infixToPostfix(exp);  
    return 0;  
}
```

Output

```
C:\Users\admin\Desktop\inpost_c.exe

Enter Infix Expression:a+b*c
Postfix Expression:abc**+
Process returned 0 (0x0) execution time : 4.278 s
Press any key to continue.
```

```
C:\Users\admin\Desktop\itopost_cp.exe

Enter Infix Expression:a+b*c
Postfix Expression:abc**+
Process returned 0 (0x0) execution time : 5.463 s
Press any key to continue.
```

Experiment-10

Q: Write a program to implement consumer producer problem using threads.

```
import java.util.LinkedList;

public class pc
{
    public static void main(String[] args)
        throws InterruptedException
    {
        final PC pc = new PC();

        Thread t1 = new Thread(new Runnable()
        {
            @Override
            public void run()
            {
                try
                {
                    pc.produce();
                }
                catch(InterruptedException e)
                {
                    e.printStackTrace();
                }
            }
        });
        Thread t2 = new Thread(new Runnable()
        {
            @Override
            public void run()
            {
                try
                {
                    pc.consume();
                }
                catch(InterruptedException e)
                {
                    e.printStackTrace();
                }
            }
        });
        t1.start();
        t2.start();
    }
}
```

```
t1.join();
t2.join();

}

public static class PC
{
    LinkedList<Integer> list = new LinkedList<>();
    int capacity = 2;

    public void produce() throws InterruptedException
    {
        int value = 0;
        while (true)
        {
            synchronized (this)
            {
                while (list.size()==capacity)
                    wait();

                System.out.println("Producer produced-"
                                   + value);
                list.add(value++);
                notify();
                Thread.sleep(1000);
            }
        }
    }

    public void consume() throws InterruptedException
    {
        while (true)
        {
            synchronized (this)
            {

                while (list.size()==0)
                    wait();

                int val = list.removeFirst();

                System.out.println("Consumer consumed-"
                                   + val);
                notify();
                Thread.sleep(1000);
            }
        }
    }
}
```

Projects

Output - JavaApplication9 (run) #3



run:

Producer produced-0
Producer produced-1
Consumer consumed-0
Consumer consumed-1
Producer produced-2
Producer produced-3
Consumer consumed-2
Consumer consumed-3
Producer produced-4
Consumer consumed-4