**A Practical Activity Report For**

**Data Structures and Algorithms (UCS406)**

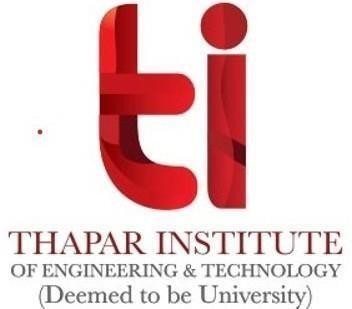
Submitted By: **Vivek Arora**

**101715178**

**(ENC 8)**

Submitted To:

**Dr. Sanjay Sharma**



**ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT THAPAR INSTITUTE OF ENGINEERING &TECHNOLOGY, (DEEMED TO BEUNIVERSITY),**

**PATIALA, PUNJAB**

**ASSIGNMENT 3**

**QUESTION 1(a) (e^x with iteration)**

#include <iostream>

using namespace std;

double e(int x, int n){

double s=1,num=1,den=1;

for(int i=1;i&lt;=n;i++){

num\*=x;

den\*=i;

s+=num/den;

}

return s;

}

int main(){

int x,n;

cin>>x>>n;

cout<<e(x,n);

return 0;

}

**QUESTION 1(a) (e^x With Recursion)**

#include <iostream>

using namespace std;

double e(int x, int n){

static double s;

if(n==0)

return s;

s=1+x\*s/n;

return e(x,n-1);

}

int main(){

int x,n;

cin>>x>>n;

cout<<e(x,n);

return 0;

}

**QUESTION 1(b): (sinx with Iteration)**

#include <iostream>

using namespace std;

int main(){

int i, n;

float x, sum, t;

cin>>x>>n;

x=x\*3.141/180;

t=x;

sum=x;

for(i=1;i<=n;i++){

t=(t\*(-1)\*x\*x)/(2\*i\*(2\*i+1));

sum=sum+t;

}

cout<<sum;

return 0;

}

**QUESTION 1(b)(sinx with Recursion )**

#include <iostream>

using namespace std;

double e(int x, int n){

static double t;

if(n==0)

return t;

t=1+t\*(-1)\*x\*x/(2\*n\*(2\*n+1));

return e(x,n-1);

}

int main(){

int i, n;

float x, sum, t;

cin>>x>>n;

x=x\*3.141/180;

cout<<e(x,n);

return 0;

}

**QUESTION 1(c) (cosx with iteration )**

#include <iostream>

using namespace std;

int main(){

int i, n;

float x, sum, t;

cin>>x>>n;

x=x\*3.141/180;

t=x;

sum=x;

for(i=1;i<=n;i++){

t=(t\*(-1)\*x\*x)/(2\*i\*(2\*i+1));

sum=sum+t;

}

cout<<sum;

return 0;

}

**QUESTION 1(c) (cosx with recursion )**

#include <iostream>

using namespace std;

double e(int x, int n){

static double t;

if(n==0)

return t;

t=1+t\*(-1)\*x\*x/(2\*n\*(2\*n+1));

return e(x,n-1);

}

int main(){

int i, n;

float x, sum, t;

cin>>x>>n;

x=x\*3.141/180;

cout<<e(x,n);

return 0;

}

**QUESTION 2 (power recursion with reduced number of multiplications )**

#include <iostream>

using namespace std;

int pow(int b, int p)

{

if (p != 0)

return (b\*pow(b, p-1));

else

return 1;

}

int main()

{

int n,x, result;

cout << "Enter base of number ";

cin >> n;

cout << "Enter power of number ";

cin >> x;

result = pow(n, x);

cout << result;

}

**QUESTION 3 (Combinational formula NCR with Recursion )**

#include <iostream>

using namespace std;

int arr[1001][1001] = { 0 };

void initialize()

{

arr[0][0] = 1;

for (int i = 1; i < 1001; i++)

{

arr[i][0] = 1;

for (int j = 1; j < i + 1; j++)

{

arr[i][j] = (arr[i - 1][j - 1] + arr[i - 1][j]);

}

}

}

int NCR(int n, int r)

{

return arr[n][r];

}

int main()

{

initialize();

int n,r;

cout<<"Enter values of n and r: "<<endl; cin>>n>>r;

cout <<"The result is:"<<NCR(n, r);

}

**QUESTION 4 (Tower of Hanoi using Recursion )**

#include<iostream>

using namespace std;

void TOH(int N,char S,char D,char H)

{

if(N<=0)

return;

TOH(N-1,S,H,D);

cout<<"moved ring"<<N<<"from"<<S<<"to"<<D<<endl;

TOH(N-1,H,D,S);

}

int main()

{

int N;

char S,H,D;

cin>>N>>S>>D>>H;

TOH(N,S,D,H);

return 0;

}

**QUESTION 5 (Fibonacci using Recursion )**

#include<iostream>

using namespace std;

int fibo(int n)

{

if((n==1)||(n==0))

{

return(n);

}

else

{

return(fibo(n-1)+fibo(n-2));

}

}

int main()

{

int n,i=0;

cout<<"Enter the number upto which you want to calculate fibonacci";

cin>>n;

cout<<endl;

while(i<n)

{

cout<<" "<<fibo(i);

i++;

}

}