#include <stdio.h>

typedef struct {

} myMath;

static double power(double x, int y) {

double result = 1.0;

for (int i = 0; i < y; i++) {

result \*= x;

}

return result;

}

static int fact(int x) {

int result = 1;

for (int i = 1; i <= x; i++) {

result \*= i;

}

return result;

}

int main() {

int n;

double x;

printf("Enter the value of x: ");

scanf("%lf", &x);

printf("Enter the value of n: ");

scanf("%d", &n);

double e\_result = 1.0;

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

double term = power(x, i) / fact(i);

e\_result += term;

}

double p\_result = 1.0;

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

double term = power(n, i) \* power(x, i) / fact(i);

p\_result += term;

}

printf("e^x = %.4lf\n", e\_result);

printf("(1 + x)^n = %.4lf\n", p\_result);

return 0;

}

A screenshot of a computer

Description automatically generated

class myMath:

@staticmethod

def power(x, y):

result = 1.0

for \_ in range(y):

result \*= x

return result

@staticmethod

def fact(x):

result = 1

for i in range(1, x+1):

result \*= i

return result

if \_\_name\_\_ == '\_\_main\_\_':

x = float(input("Enter the value of x: "))

n = int(input("Enter the value of n: "))

e\_result = 1.0

for i in range(1, n+1):

term = myMath.power(x, i) / myMath.fact(i)

e\_result += term

p\_result = 1.0

for i in range(1, n+1):

term = myMath.power(n, i) \* myMath.power(x, i) / myMath.fact(i)

p\_result += term

print("e^x =", e\_result)

print("(1 + x)^n =", p\_result)

A screenshot of a computer program

Description automatically generated