



รายงาน

เรื่อง โปรแกรม Recursion M<sup>N</sup>

จัดทำโดย

นายพงษ์พันธุ์ เลาวพงศ์

รหัสนักศึกษา 66543206019-2

เสนอ

อาจารย์ปิยพล ยืนยงสถาวร

รายงานนี้เป็นส่วนหนึ่งของวิชา

ENGCE124

โครงสร้างข้อมูลและขั้นตอนวิธี

(Data Structures and Algorithms)

หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่

ภาคเรียนที่ 1 ปีการศึกษา 2567

รายงาน

เรื่อง โปรแกรม Recursion M<sup>N</sup>

จัดทำโดย

นายพงษ์พันธุ์ เลาวพงศ์

รหัสนักศึกษา 66543206019-2

เสนอ

อาจารย์ปิยพล ยืนยงสถาวร

รายงานนี้เป็นส่วนหนึ่งของวิชา

ENGCE124

โครงสร้างข้อมูลและขั้นตอนวิธี

(Data Structures and Algorithms)

หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์

สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่

ภาคเรียนที่ 1 ปีการศึกษา 2567

## คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา ENGCE124 โครงสร้างข้อมูลและขั้นตอนวิธี (Data Structures and Algorithms) หลักสูตร วศ.บ.วิศวกรรมคอมพิวเตอร์ สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา ภาคพายัพ เชียงใหม่ ในระดับปริญญาตรีปีที่ 2 โดยมีจุดประสงค์ในการอธิบายโค้ดของโปรแกรม Recursion  $M^N$  รวมถึงอธิบายหลักการทำงานของโปรแกรม Recursion  $M^N$  และอธิบายผลลัพธ์การใช้งานโปรแกรม Recursion  $M^N$

ผู้จัดทำรายงานหวังว่า รายงานฉบับนี้จะเป็นประโยชน์กับผู้สนใจ หรือนักศึกษาทุกท่านที่กำลังหาศึกษาในหัวข้อของโปรแกรม Recursion  $M^N$  หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้ และขออภัยมา ณ ที่นี้

ผู้จัดทำ

นายพงษ์พันธุ์ เลาวพงศ์

วันที่ 01/08/2567

## สารบัญ

	หน้า
คำนำ	ก
สารบัญ	๗
โค้ดของโปรแกรม Recursion $M^N$ พร้อมคำอธิบาย	1
หลักการทำงานของโปรแกรม Recursion $M^N$	3
ผลลัพธ์การใช้งานโปรแกรม Recursion $M^N$	6
บรรณานุกรม	7

## โค้ดของโปรแกรม Recursion $M^N$ พร้อมคำอธิบาย

```
#include <stdio.h> // ใช้ฟังก์ชัน printf()

#include <conio.h> // ใช้ฟังก์ชัน getch()

int Base, Exponent, result; // ประกาศตัวแปรสำหรับฐาน, ยกกำลัง, และผลลัพธ์

// ฟังก์ชันรีเคอร์ซีฟสำหรับคำนวณ  $M^N$ 

int Power(int M, int N) {
    if (N == 0) { // กรณีฐาน: ถ้า N เท่ากับ 0
        printf(".....Roll Back Point\n"); // แสดงจุดที่ทำการ
        ย้อนกลับ

        return 1; // ส่งคืนค่า 1 เพราะ  $M^0 = 1$ 
    } else {
        // แสดงข้อความก่อนการเรียกฟังก์ชันรีเคอร์ซีฟ

        printf("%2d^%2d = %2d * %2d^%2d\n", M, N, M, M, N-1);

        int partial_result = Power(M, N - 1); // เรียกฟังก์ชัน Power ด้วย N-1

        // แสดงข้อความหลังการเรียกฟังก์ชันรีเคอร์ซีฟ

        printf("%2d^%2d = %2d * %3d = %5d\n", M, N, M,
        partial_result, M * partial_result);

        return M * partial_result; // ส่งคืนผลลัพธ์ M * partial_result
    }
}

int main() {
    printf("RECURSIVE (POWER) PROGRAM\n");
    printf("=====\n");
    while (1) {
        printf("Enter Base (-999 to END) : ");
        scanf("%d", &Base);

        if (Base == -999) break; // สิ้นสุดโปรแกรมถ้าผู้ใช้ป้อน -999
    }
}
```

## โค้ดของโปรแกรม Recursion $M^N$ พร้อมคำอธิบาย (ต่อ)

```

printf("Enter Exponent (-999 to END) : ");
scanf("%d", &Exponent);

if (Exponent == -999) break; // สิ้นสุดโปรแกรมถ้าผู้ใช้ป้อน -999

if (Exponent >= 0) {
    printf("M^N = M * M^(N-1)\n");
    printf("-----\n");
    result = Power(Base, Exponent); // เรียกใช้ฟังก์ชัน Power
    printf("\nAnswer %d^%d = %d\n", Base, Exponent,
result);
    printf("-----Finished\n");
    getch();
}
}
return 0;
}

```

## หลักการการทำงานของโปรแกรม Recursion $M^N$

### 1. การนำเข้าไลบรารีและการกำหนดค่าตัวแปร

```
#include <stdio.h> // ใช้ฟังก์ชัน printf()

#include <conio.h> // ใช้ฟังก์ชัน getch()

int Base, Exponent, result; // ประกาศตัวแปรสำหรับฐาน, ยกกำลัง, และผลลัพธ์
```

ในส่วนของการนำเข้าไลบรารีและการกำหนดค่าตัวแปร ได้ทำการนำเข้าไลบรารีที่จำเป็น ได้แก่ `stdio.h` ใช้สำหรับฟังก์ชันการพิมพ์และรับข้อมูล เช่น `printf()` และ `scanf()`. และ `conio.h` ใช้สำหรับฟังก์ชันการจัดการกับคีย์บอร์ด ได้แก่คำสั่ง `getch()` โดยใช้งานเพื่อรอการกดคีย์จากผู้ใช้ก่อนที่จะดำเนินการต่อ (มักใช้เพื่อหยุดโปรแกรมชั่วคราวจนกว่าผู้ใช้จะกดคีย์) ในส่วนของตัวแปรจะประกอบไปด้วยตัวแปร `Base` และ `Exponent` สำหรับเก็บค่าฐานและยกกำลังตามลำดับ และตัวแปร `result` สำหรับเก็บผลลัพธ์ที่คำนวณได้ ซึ่งการกำหนดตัวแปรเหล่านี้เป็นส่วนสำคัญในการจัดการข้อมูลที่ได้รับเข้ามาและผลลัพธ์ที่คำนวณได้ในโปรแกรม

## หลักการทำงานของโปรแกรม Recursion $M^N$ (ต่อ)

### 2. ฟังก์ชัน Power

```
int Power(int M, int N) {
    if (N == 0) { // กรณีฐาน: ถ้า N เท่ากับ 0
        printf(".....Roll Back Point\n"); // แสดงจุดที่ทำการ
        ย้อนกลับ
        return 1; // ส่งคืนค่า 1 เพราะ  $M^0 = 1$ 
    } else {
        // แสดงข้อความก่อนการเรียกฟังก์ชันรีเคอร์ซีฟ
        printf("%2d^%2d = %2d * %2d^%2d\n", M, N, M, M, N-1);
        int partial_result = Power(M, N - 1); // เรียกฟังก์ชัน Power ด้วย N-1
        // แสดงข้อความหลังการเรียกฟังก์ชันรีเคอร์ซีฟ
        printf("%2d^%2d = %2d * %3d = %5d\n", M, N, M,
        partial_result, M * partial_result);
        return M * partial_result; // ส่งคืนผลลัพธ์ M * partial_result
    }
}
```

ฟังก์ชัน Power(int M, int N) คำนวณค่า  $M^N$  โดยใช้วิธีการเรียกใช้ฟังก์ชันแบบรีเคอร์ซีฟ โดยในกรณีฐาน ถ้าหากเท่ากับ 0 ฟังก์ชันจะส่งคืนค่า 1 เพราะ  $M^0 = 1$  และกรณีทั่วไป ถ้า N ไม่เท่ากับ 0, ฟังก์ชันจะแสดงข้อความเพื่อแสดงขั้นตอนการคำนวณ, ทำการเรียกฟังก์ชัน Power(M, N - 1) (รีเคอร์ซีฟ) เพื่อคำนวณ  $M^{N-1}$  จากนั้นคำนวณ  $M^N$  โดยการคูณ M กับผลลัพธ์จากการคำนวณ  $M^{N-1}$



## หลักการการทำงานของโปรแกรม Recursion $M^N$ (ต่อ)

### 3. ฟังก์ชัน main

```
int main() {
    printf("RECURSIVE (POWER) PROGRAM\n");
    printf("=====\n");
    while (1) {
        printf("Enter Base (-999 to END) : ");
        scanf("%d", &Base);

        if (Base == -999) break; // สิ้นสุดโปรแกรมถ้าผู้ใช้ป้อน -999

        printf("Enter Exponent (-999 to END) : ");
        scanf("%d", &Exponent);

        if (Exponent == -999) break; // สิ้นสุดโปรแกรมถ้าผู้ใช้ป้อน -999

        if (Exponent >= 0) {
            printf("M^N = M * M^(N-1)\n");
            printf("-----\n");

            result = Power(Base, Exponent); // เรียกใช้ฟังก์ชัน Power

            printf("\nAnswer %d^%d = %d\n", Base, Exponent, result);
            printf("-----Finished\n");
            getch();
        }
    }
    return 0;
}
```

ฟังก์ชัน main จะแสดงข้อความเริ่มต้นโปรแกรม และจะใช้ลูป while (1) เพื่อให้โปรแกรมทำงานอย่างต่อเนื่องจนกว่าผู้ใช้จะป้อนค่า -999 จากนั้นจะขอให้ผู้ใช้ป้อนค่าฐาน (Base) และยกกำลัง (Exponent) ถ้าผู้ใช้ป้อนค่า -999 โปรแกรมจะหยุดทำงาน ถ้ายกกำลังเป็นเลขบวกหรือศูนย์ โปรแกรมจะเรียกใช้ฟังก์ชัน Power() เพื่อคำนวณค่ากำลัง จากนั้นจะแสดงผลลัพธ์ของการคำนวณ โดยใช้ getch() เพื่อรอให้ผู้ใช้กดคีย์ใดๆ ก่อนที่จะทำงานต่อ.

## ผลลัพธ์การใช้งานโปรแกรม Recursion $M^N$

```

E:\ENGCE124\Coding 8 Progr  X + v
RECURSIVE (POWER) PROGRAM
=====
Enter Base (-999 to END) : 2
Enter Exponent (-999 to END) : 3
M^N = M * M^(N-1)
=====
2^ 3 = 2 * 2^ 2
2^ 2 = 2 * 2^ 1
2^ 1 = 2 * 2^ 0
.....Roll Back Point
2^ 1 = 2 * 1 = 2
2^ 2 = 2 * 2 = 4
2^ 3 = 2 * 4 = 8

Answer 2^3 = 8
-----Finished

```

สมมติเมื่อผู้ใช้ป้อนค่า M (Base) มีค่าเท่ากับ 2 และป้อนค่า N (Exponent) มีค่าเท่ากับ 3 โดยโปรแกรมจะพิมพ์ขั้นตอนการคำนวณระหว่างการเรียกใช้ฟังก์ชัน Power() ซึ่งผลลัพธ์ที่ได้จะแสดงผลดังนี้

1. ก่อนการเรียกฟังก์ชันรีเคอร์ซีฟ ผลลัพธ์ที่ได้จะเป็น

$$\begin{aligned}
 2^3 &= 2 * 2^2 \\
 2^2 &= 2 * 2^1 \\
 2^1 &= 2 * 2^0
 \end{aligned}$$

2. กรณีกำลังเท่ากับ 0 ผลลัพธ์ที่ได้จะเป็น

$$2^0 = 1$$

3. หลังการเรียกฟังก์ชันรีเคอร์ซีฟ

$$\begin{aligned}
 2^1 &= 2 * 1 = 2 \\
 2^2 &= 2 * 2 = 4 \\
 2^3 &= 2 * 4 = 8
 \end{aligned}$$

4. การแสดงผลสุดท้าย

$$\text{Answer } 2^3 = 8$$

หลังจากแสดงผลลัพธ์ โปรแกรมจะรอให้ผู้ใช้กดคีย์ใดๆ ก่อนที่จะทำการป้อนข้อมูลใหม่หรือสิ้นสุดการทำงาน

## บรรณานุกรม

ChatGPT. ( - ). Calculating Powers Using Recursive Functions in C. สืบค้น 1 สิงหาคม 2567,  
จาก <https://chatgpt.com/>