



รายงาน

เรื่อง โปรแกรม Recursion

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

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 ภาควิชาฯ เชียงใหม่ ในระดับปริญญาตรีปีที่ 2 โดยมีจุดประสงค์ในการอธิบายโค้ดของโปรแกรม Recursion รวมถึงอธิบายหลักการทำงานของโปรแกรม Recursion และอธิบายผลลัพธ์การทำงานของโปรแกรม Recursion

ผู้จัดทำรายงานหวังว่า รายงานฉบับนี้จะเป็นประโยชน์กับผู้สนใจ หรือนักศึกษาทุกท่านที่กำลังหาศึกษาในหัวข้อของโปรแกรม Recursion หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้

ผู้จัดทำ

นายพงษ์พันธุ์ เลาหวงศ์

วันที่ 01/08/2567

สารบัญ

	หน้า
คำนำ	ก
สารบัญ	๗
โค้ดของโปรแกรม Recursion พร้อมคำอธิบาย	1
หลักการทำงานของโปรแกรม Recursion	3
ผลลัพธ์การใช้งานโปรแกรม Recursion	6
บรรณานุกรม	8

โค้ดของโปรแกรม Recursion พร้อมคำอธิบาย

```

/*
Program Recursion can..
    1. Calculate Factorial Function
    2. Show each step of recursive result
    3. Show final result
=====
*/

#include <stdio.h> //use printf() //ใช้งานฟังก์ชัน printf()
#include <conio.h> //use getch() //ใช้งานฟังก์ชัน getch()

int Number , ans ;

int Factorial( int N ) { //ฟังก์ชันการคำนวณแฟคทอเรียล

    int x , y ;
    if( N == 0 ) {
        printf( ".....Roll Back Point\n" ) ;
        return( 1 ) ;
    } else {
        x = N - 1 ;

        printf( "%2d! = %2d * %2d!\n" , N , N , x ) ; //แสดงผลก่อนการ
เรียกซ้ำ (Recursive)

        y = Factorial( x ) ;
        printf( "%2d! = %2d * %3d = %5d\n" , N , N , y , y * N )
; //แสดงผลหลังการเรียกซ้ำ

        return( N * y ) ;
    }
} //จบการทำงานฟังก์ชัน Factorial

```

โค้ดของโปรแกรม Recursion พร้อมคำอธิบาย (ต่อ)

```
int main() {
    printf( "RECRSIVE(FACTORIAL) PROGRAM\n" ) ;
    printf( "=====\n" ) ;
    //N=0;
    while( Number != -999 ) {
        printf( "Enter Number (-999 is END) : " ) ;
        scanf( "%d" , &Number ) ;
        if( Number >= 0 ) {
            printf( "N! = N(N-1)!\n" ) ;
            printf( "-----\n" ) ;
            ans = Factorial( Number ) ; //เรียกฟังก์ชันซ้ำ
            printf( "\nAnswer N! = %d\n" , ans ) ;
            printf( "-----Finished\n" ) ;
            getch() ;
        } //จบการทำงานของ if
    } //จบการทำงานของ While Loop
    return(0);
} //จบการทำงานฟังก์ชัน Main
```

หลักการทำงานของโปรแกรม Recursion

1. การนำเข้าไลบรารีและการกำหนดค่าตัวแปร

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

#include <conio.h> //use getch() //ใช้งานฟังก์ชัน getch()

int Number , ans ;
```

ในส่วนของการนำเข้าไลบรารีและการกำหนดค่าตัวแปร ได้ทำการนำเข้าไลบรารีที่จำเป็น ได้แก่ stdio.h ใช้สำหรับฟังก์ชันการพิมพ์และรับข้อมูล เช่น printf() และ scanf(). และ conio.h ใช้สำหรับฟังก์ชันการจัดการกับคีย์บอร์ด ได้แก่คำสั่ง getch() โดยใช้งานเพื่อรอการกดคีย์จากผู้ใช้ก่อนที่จะดำเนินการต่อ (มักใช้เพื่อหยุดโปรแกรมชั่วคราวจนกว่าผู้ใช้จะกดคีย์) ในส่วนของตัวแปรจะประกอบไปด้วยตัวแปร Number ใช้เก็บค่าที่ผู้ใช้ป้อนเข้ามา ซึ่งจะเป็นตัวเลขที่ต้องการคำนวณแฟคทอเรียล และตัวแปร ans ใช้เก็บผลลัพธ์ของการคำนวณแฟคทอเรียลที่ได้จากการเรียกฟังก์ชัน Factorial ซึ่งการกำหนดตัวแปรเหล่านี้เป็นส่วนสำคัญในการจัดการข้อมูลที่ได้รับเข้ามาและผลลัพธ์ที่คำนวณได้ในโปรแกรม

หลักการการทำงานของโปรแกรม Recursion (ต่อ)

2. ฟังก์ชัน Factorial

```
int Factorial( int N ) { //ฟังก์ชันการคำนวณแฟคทอเรียล

    int x , y ;
    if( N == 0 ) {
        printf( ".....Roll Back Point\n" ) ;
        return( 1 ) ;
    } else {
        x = N - 1 ;

        printf( "%2d! = %2d * %2d!\n" , N , N , x ) ; //แสดงผลก่อนการ
เรียกซ้ำ (Recursive)

        y = Factorial( x ) ;
        printf( "%2d! = %2d * %3d = %5d\n" , N , N , y , y * N )
; //แสดงผลหลังการเรียกซ้ำ

        return( N * y ) ;
    }
} //จบการทำงานฟังก์ชัน Factorial
```

ฟังก์ชัน Factorial ทำหน้าที่ในการคำนวณค่าแฟคทอเรียลของตัวเลขที่ป้อนเข้ามา โดยใช้การเรียกซ้ำ โดยรับพารามิเตอร์ N ซึ่งเป็นตัวเลขที่ต้องการคำนวณแฟคทอเรียล ซึ่งหลักการทำงาน ถ้า N เท่ากับ 0 (กรณีฐานของการเรียกซ้ำ) ฟังก์ชันจะพิมพ์ ".....Roll Back Point" และคืนค่า 1 (เพราะ $0! = 1$) แต่ถ้า N ไม่เท่ากับ 0 ฟังก์ชันจะคำนวณค่า x เป็น $N - 1$ จากนั้นจะแสดงขั้นตอนการคำนวณแฟคทอเรียลก่อนการเรียกซ้ำ โดยพิมพ์ว่า $\%2d! = \%2d * \%2d!$ ซึ่งแสดงการคำนวณในรูปแบบ $N! = N * (N-1)!$ และเรียกฟังก์ชัน Factorial โดยส่งค่า x เป็นพารามิเตอร์ (การเรียกซ้ำ) หลังจากได้รับผลลัพธ์ y จากการเรียกซ้ำ, แสดงผลลัพธ์การคำนวณที่เสร็จแล้วในรูปแบบ $\%2d! = \%2d * \%3d = \%5d$ ซึ่งแสดงผลในรูปแบบ $N! = N * y = y * N$ และสุดท้ายจะคืนค่า $N * y$ ซึ่งเป็นผลลัพธ์สุดท้ายของการคำนวณ

หลักการการทำงานของโปรแกรม Recursion (ต่อ)

3. ฟังก์ชัน main

```
int main() {
    printf( "RECRSIVE (FACTORIAL) PROGRAM\n" ) ;
    printf( "=====\n" ) ;
    //N=0;
    while( Number != -999 ) {
        printf( "Enter Number (-999 is END) : " ) ;
        scanf( "%d" , &Number ) ;
        if( Number >= 0 ) {
            printf( "N! = N(N-1)!\n" ) ;
            printf( "-----\n" ) ;
            ans = Factorial( Number ) ; //เรียกฟังก์ชันซ้ำ
            printf( "\nAnswer N! = %d\n" , ans ) ;
            printf( "-----Finished\n" ) ;
            getch() ;
        } //จบการทำงานของ if
    } //จบการทำงานของ While Loop
    return(0) ;
} //จบการทำงานฟังก์ชัน Main
```

ฟังก์ชัน main ทำหน้าที่ในการทำงานเป็นเมนูหลักของโปรแกรมที่รับข้อมูลจากผู้ใช้และเรียกฟังก์ชัน Factorial เพื่อคำนวณแฟคทอเรียล โดยหลักการทำงาน เริ่มต้นจาก แสดงข้อความเริ่มต้นว่า "โปรแกรม คำนวณแฟคทอเรียลด้วยการเรียกซ้ำ" จากนั้นจะใช้งาน ใช้ลูป while เพื่อรับค่าจากผู้ใช้จนกว่าผู้ใช้จะป้อน - 999 (ซึ่งจะหยุดลูป) โดยถ้าผู้ใช้ป้อนหมายเลขที่มากกว่าหรือเท่ากับ 0 จะแสดงข้อความอธิบายวิธีการคำนวณ ว่า $N! = N(N-1)!$ โดยการเรียกฟังก์ชัน Factorial โดยส่งค่าที่ผู้ใช้ป้อนเข้ามา จากนั้นแสดงผลแฟคทอเรียล ที่คำนวณได้ และ แสดงข้อความว่า "เสร็จสิ้น" และรอการกดคีย์จากผู้ใช้ (ใช้ getch())

โดยสรุปการทำงานของโปรแกรม เริ่มต้นที่ main, โปรแกรมจะแสดงข้อความเริ่มต้นและเริ่มลูป while เพื่อรับค่าจากผู้ใช้ จากนั้นโปรแกรมจะคำนวณค่าแฟคทอเรียลโดยเรียกฟังก์ชัน Factorial ที่จะคำนวณแบบเรียกซ้ำ และจะแสดงผลของการคำนวณและขั้นตอนการคำนวณจะถูกแสดงให้ผู้ใช้เห็น แต่สุดท้าย โปรแกรมจะหยุดเมื่อผู้ใช้ป้อนค่า -999

ผลลัพธ์การใช้งานโปรแกรม Recursion

```

E:\ENGCE124\Coding 7 Recur  X  +  v
RECRSIVE(FACTORIAL) PROGRAM
=====
Enter Number (-999 is END) : 4
N! = N(N-1)!
=====
4! = 4 * 3!
3! = 3 * 2!
2! = 2 * 1!
1! = 1 * 0!
.....Roll Back Point
1! = 1 * 1 = 1
2! = 2 * 1 = 2
3! = 3 * 2 = 6
4! = 4 * 6 = 24

Answer N! = 24
-----Finished
|

```

เมื่อผู้ใช้ป้อนหมายเลขที่เป็นค่าบวก (เช่น 4) โปรแกรมจะทำการคำนวณแฟคทอเรียลของหมายเลขนั้นโดยใช้ฟังก์ชัน Factorial ดังนี้

1. การคำนวณและการเรียกซ้ำ (สมมติว่าผู้ใช้ป้อนหมายเลข 4)
 - การคำนวณจากแฟคทอเรียลของ 4
 - โปรแกรมจะแสดง : $4! = 4 * 3!$
 - แล้วจะเรียก Factorial(3)
 - การคำนวณจากแฟคทอเรียลของ 3
 - โปรแกรมจะแสดง : $3! = 3 * 2!$
 - แล้วจะเรียก Factorial(2)
 - การคำนวณจากแฟคทอเรียลของ 2
 - โปรแกรมจะแสดง : $2! = 2 * 1!$
 - แล้วจะเรียก Factorial(1)
 - การคำนวณจากแฟคทอเรียลของ 1
 - โปรแกรมจะแสดง : $1! = 1 * 0!$
 - แล้วจะเรียก Factorial(0)
 - การคำนวณจากแฟคทอเรียลของ 0 (กรณีฐาน)
 - โปรแกรมจะแสดง :Roll Back Point
 - คืนค่า 1 (เพราะ $0! = 1$)

2. การย้อนกลับ (Backtracking) หลังจากการคำนวณที่ระดับ 0 เสร็จสิ้น, โปรแกรมจะเริ่มย้อนกลับ

- การคำนวณจากแฟคทอเรียลของ 1
 - โปรแกรมจะแสดง : $1! = 1 * 1 = 1$
- การคำนวณจากแฟคทอเรียลของ 2
 - โปรแกรมจะแสดง : $2! = 2 * 1 = 2$
- การคำนวณจากแฟคทอเรียลของ 3
 - โปรแกรมจะแสดง : $3! = 3 * 2 = 6$
- การคำนวณจากแฟคทอเรียลของ 4
 - โปรแกรมจะแสดง : $4! = 4 * 6 = 24$
- การแสดงผลลัพธ์
 - โปรแกรมจะแสดง : Answer N! = 24
 - โปรแกรมจะแสดง : -----Finished

เมื่อผู้ใช้งานต้องการหยุดโปรแกรม โปรแกรมจะรอให้ผู้กดคีย์ (โดยใช้ getch()) ก่อนที่จะดำเนินการต่อไปยังรอบถัดไปในลูป while หรือสิ้นสุดโปรแกรมหากผู้ใช้ป้อน -999

โปรแกรมนี้นี้ช่วยให้ผู้ใช้เห็นวิธีการคำนวณแฟคทอเรียลในรูปแบบของการเรียกซ้ำ รวมถึงการแสดงผลลัพธ์ระหว่างการเรียกซ้ำและผลลัพธ์สุดท้าย

บรรณานุกรม

ChatGPT. (-). Recursive Factorial Calculation in C Programming . สืบค้น 1 สิงหาคม 2567,
จาก <https://chatgpt.com/>