



ใบงานที่ 8

เรื่อง Recursion MN

เสนอ

อาจารย์ ปิยพล ยืนยงสถาวร

จัดทำโดย

นายธีรเดช ประเสริฐวงศ์พนา

65543206016-9

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา โครงสร้างข้อมูลและขั้นตอนวิธี

หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

ประจำภาคที่ 1 ปีการศึกษา 2566

คำสั่ง/คำชี้แจง

* ให้ระบุคำสั่ง/คำชี้แจง/คำอธิบาย ตามที่กำหนดให้

1. สร้างโปรแกรม Convert infix to postfix ตามตัวอย่างในเอกสารประกอบการสอน 4
2. แสดงโค้ดโปรแกรมเป็นส่วนๆ พร้อมอธิบาย
3. แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน
4. สรุปผลการทดลอง

ลำดับขั้นตอนการทดลอง

* ให้ระบุขั้นตอนการทดลอง ผลลัพธ์ที่ได้ โดยใช้รูปภาพประกอบ และอธิบายอย่างละเอียด

แสดงโค้ดโปรแกรมเป็นส่วนๆ พร้อมอธิบาย

ส่วนที่1

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

int Power;
double Number, ans;
```

ส่วนที่ 1 เป็นที่ประกาศการใช้ library และประกาศตัวแปร ในโค้ดดังนี้

-Library ที่ใช้มี 2 ตัวดังนี้

<stdio.h> คือ ใช้สำหรับฟังก์ชัน printf() ที่ใช้แสดงผลบนหน้าจอ

<conio.h> คือ ใช้สำหรับฟังก์ชัน getch() ที่รอรับอักขระที่ผู้ใช้ป้อนจากแป้นพิมพ์โดยไม่แสดงผลบนหน้าจอ

-ส่วนตัวแปร Number และ ans เป็นตัวแปรที่ถูกประกาศเพื่อใช้เก็บค่าที่ใช้ในการคำนวณแฟกทอเรียลและผลลัพธ์ โดยมีประเภทข้อมูลเป็น int

ส่วนที่ 2

```
double PowerStep(double M, int N) // PowerStep Function
{
    int i;
    double result;
    if (N == 0)
    {
        return (1);
    }
    else
    {
        result = PowerStep(M, N - 1);
        printf("%.21f ^ %d : ", M, N); // Display Step After Recursive

        printf("%.21f * %.21f : ", M, result);

        printf("= %.21f\n", M * result); // Calculation results for each round After Recursive
        return (M * result);
    }
} // End Fn
```

ส่วนที่ 2 ฟังก์ชัน PowerStep โดยรับค่า Parameter เข้ามาคือ M และ N เป็นฟังก์ชันการใช้เทคนิคการเรียกตัวเองซ้ำเพื่อคำนวณค่าของยกกำลัง MN โดยที่ M คือตัวเลขที่ต้องการยกกำลังและ N คือค่าของกำลังที่ต้องการ และประกาศตัวแปรแบบ Local ชื่อ result เพื่อเก็บค่าผลลัพธ์ในแต่ละรอบการทำงานต่อมาก็ตรวจสอบเงื่อนไข If โดยเมื่อ N หรือเลขยกกำลังเป็น 0 ให้ทำการคืนค่า 1 กลับไปเลย ถ้าหากเงื่อนไข If เป็นเท็จ หรือ N ไม่ได้เป็น 0 ให้ทำงานในเงื่อนไข Else

ให้ทำการเรียกฟังก์ชันตัวเองซ้ำ โดยครั้งถัดไปที่เรียกให้ลดค่า N ลด 1 ค่า แต่ค่า M ยังคงเท่าเดิม โดยเมื่อค่า N ลดลงเป็น 0 จะกลับไปเข้าเงื่อนไข If แทน จากนั้นจะเริ่ม Return ค่า

มาเก็บไว้ในตัวแปร result และให้ทำการแสดงผลรูปแบบการยกกำลัง ขั้นตอนที่กำลังถูกคำนวณ โดยแสดงตัวเลข M และ N ที่กำลังถูกคำนวณในขณะนั้น

ทำการแสดงผลค่าที่กำลังถูกคำนวณในขณะนั้นแสดงผลลัพธ์ของค่าที่ถูกคำนวณในขณะนั้นซึ่งคือ $M * result$ เพื่อแสดงค่าของกำลัง MN หลังจากคำนวณเสร็จในรอบนั้น ๆ

Return ค่าผลลัพธ์ของการคำนวณที่ได้ในรอบนั้น เพื่อนำไปใช้คำนวณในรอบถัดไปหรือ

Return ค่าออกจากฟังก์ชันเมื่อออกจากรอบสุดท้ายของการเรียกตัวเองซ้ำ

ส่วนที่ 3

```
int main()
{
    printf("RECRSIVE(POWER) PROGRAM\n");
    printf("=====\n");
    while (Power >= 0)
    {
        printf("Enter Number : ");
        scanf("%lf", &Number);
        printf("Enter Power (-Power is END): ");
        scanf("%d", &Power);
        if (Power >= 0)
        {
            printf("-----Step-----\n");
            ans = PowerStep(Number, Power); // Recursive it self
            printf("\nAnswer Power %.2lf^%d = %.2lf\n", Number, Power, ans);
            printf("-----Finished-----\n");
            getch();
        }
    } // END while
    return (0);
} // End Main
```

ส่วนที่ 3 ฟังก์ชัน Main รับค่าตัวเลข (Number) และค่ากำลัง (Power) จากผู้ใช้และนำไปใช้ในการเรียกใช้ฟังก์ชัน PowerStep ที่มีการทำซ้ำในการคำนวณค่าของยกกำลัง และแสดงผลลัพธ์ของการคำนวณ ทุกครั้งที่คำนวณเสร็จสิ้นและแสดงผลชื่อโปรแกรม

-**ส่วนกรอบสีแดง** While Loop วนทำงานจนกว่าค่าของ Power เป็นค่าลบหรือน้อยกว่า 0 หมายความว่าถ้า User กรอกค่า Power เป็นค่าลบ จะจบการทำงานของโปรแกรมและแสดงผลข้อความแจ้งให้ User กรอกค่าตัวเลขที่ต้องการจะยกกำลังแล้วนำไปเก็บไว้ในตัวแปร Number ต่อมาจะแสดงผลข้อความแจ้งให้ User กรอกค่ายกกำลังแล้วนำไปเก็บไว้ในตัวแปร Power และตรวจสอบเงื่อนไข If โดยเมื่อ Power มากกว่าหรือเท่ากับ 0 ให้ทำงานในเงื่อนไข If ให้แสดงผลแจ้งขั้นตอนการทำงาน โดยทำการเรียกใช้งานฟังก์ชัน PowerStep โดยส่งค่า Parameter คือ Number และ Power จากนั้นเมื่อคืนค่ากลับให้เก็บค่าไว้ในตัวแปร ans ซึ่งจะแสดงรูปแบบการยกกำลังและผลลัพธ์ของการคำนวณค่ายกกำลังที่ได้จากฟังก์ชัน

PowerStep และแสดงผลข้อความสิ้นสุดโปรแกรม โดยรอผู้ใช้กดปุ่มใด ๆ บนแป้นพิมพ์เพื่อดำเนินการต่อ Return 0 กลับไป

แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน

```
RECRSIVE(POWER) PROGRAM
=====
Enter Number : |
```

โปรแกรมจะขึ้นให้ User กรอก ตัวเลขที่จะยกกำลังและตัวยกกำลัง

```
Enter Number : 3
Enter Power (-Power is END): 2
```

หาก User กรอกเลขยกกำลังติดลบ โปรแกรมจะจบการทำงาน

```
RECRSIVE(POWER) PROGRAM
=====
Enter Number : 3
Enter Power (-Power is END): 2
-----Step-----
3.00 ^ 1 : 3.00 * 1.00 : = 3.00
3.00 ^ 2 : 3.00 * 3.00 : = 9.00

Answer Power 3.00^2 = 9.00
-----Finished-----
```

เมื่อ User ใส่ข้อมูลเรียบร้อยแล้ว โปรแกรมจะคำนวณเลขยกกำลัง (M ยกกำลัง N) พร้อมแสดงขั้นตอนการคำนวณและผลลัพธ์ออกมา

สรุปผลการทดลอง

โปรแกรมใช้เทคนิคการเขียนฟังก์ชันทำซ้ำ (Recursive Function) เพื่อคำนวณค่าของยกกำลัง (Power) ของเลขที่ User ป้อนเข้ามา โดยแสดงขั้นตอนการทำงานในแต่ละรอบออกทางหน้าจอ

สื่อ / เอกสารอ้างอิง

-อาจารย์ปิยพล ชื่นขงสถาวร : เอกสารประกอบการสอน 4 นิพจน์อินฟิสิกส์โพสฟิสิกส์การหาค่าผลลัพธ์
นิพจน์โพสฟิสิกส์การเขียนโปรแกรมเรียกตนเอง(Recursive)