



ใบงานที่ 8

เรื่อง Program Recursion M^N

เสนอ

อาจารย์ ปิยพล ยืนยงสถาวร

จัดทำโดย

นายกฤษฎา วิทยา

65543206041-7

ใบงานนี้เป็นส่วนหนึ่งของรายวิชา โครงสร้างข้อมูลและขั้นตอนวิธี

หลักสูตรวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา

ประจำภาคที่ 1 ปีการศึกษา 2566

คำสั่ง/คำชี้แจง

- แสดงโค้ดโปรแกรมเป็นส่วนๆพร้อมทั้งอธิบาย
- แสดงผลการรันโปรแกรม พร้อมอธิบายการทำงาน
- สรุปผลการทดลอง

ลำดับขั้นตอนการทดลอง

```
1 #include <stdio.h>
2 #include <conio.h>
```

- **#include <stdio.h>:** เป็นส่วนของโค้ดที่ใช้ระบุว่าโปรแกรมต้องการใช้ฟังก์ชันที่เกี่ยวข้องกับการป้อนออกข้อมูล
- **#include <conio.h>:** เป็นส่วนของโค้ดที่ใช้ระบุว่าโปรแกรมต้องการใช้ฟังก์ชันที่เกี่ยวข้องกับการควบคุมอินพุตทางคีย์บอร์ด getch()

```
4 double power( double m, int n ) {
5     double x, y;
6     if( n > 0 ) {
7         x = n - 1;
8         printf("%.0lf^%.0lf = %.0lf * %.0lf\n", m, n, m, m, x);
9         y = power( m, x );
10        printf("%.0lf^%.0lf = %.0lf * %.0lf = %.0lf\n", m, n, m, y, (m * y));
11        return (m * y);
12    }
13    else if( n < 0 ) {
14        x = n + 1;
15        printf("%.0lf^%.0lf = (1/%.0lf) * %.0lf\n", m, n, m, m, x);
16        y = power( m, x );
17        printf("%.0lf^%.0lf = (1/%.0lf) * %.0lf = %.0lf\n", m, n, m, y, ((1 / m) * y));
18        return ((1 / m) * y);
19    }
20    else {
21        printf(".....Roll Back Point\n");
22        return 1;
23    }
24 }
```

double power(double m, int n): เป็นฟังก์ชันการยกกำลัง รับค่า m เป็นเลขฐานและ n เป็นเลขตัวดำเนินการที่ต้องการยกกำลัง ฟังก์ชันนี้เริ่มต้นด้วยการตรวจสอบเงื่อนไขในการเลือกทำงาน:

- ถ้า n มากกว่า 0 ให้ทำงานในส่วนของบวก (เมื่อยกกำลังเป็นเลขจำนวนเต็มบวก)
- ถ้า n น้อยกว่า 0 ให้ทำงานในส่วนของลบ (เมื่อยกกำลังเป็นเลขจำนวนเต็มลบ)
- ถ้า n เป็น 0 ให้คืนค่า 1 ตามนิพจน์ $a^0 = 1$ (เมื่อ a ไม่เท่ากับ 0)
- ในส่วนของการยกกำลังเป็นบวก $n > 0$, ฟังก์ชันจะทำงานตามขั้นตอนนี้:
 - นำ n ลบ 1 และเก็บในตัวแปร x

- เรียกฟังก์ชัน `power(m, x)` เพื่อคำนวณค่า m^x และเก็บผลลัพธ์ในตัวแปร `y`
- คำนวณค่าผลลัพธ์ $m * y$
- ในส่วนของการยกกำลังเป็นลบ $n < 0$, ฟังก์ชันจะทำงานตามขั้นตอนนี้:
 - นำ `n` บวก 1 และเก็บในตัวแปร `x`
 - เรียกฟังก์ชัน `power(m, x)` เพื่อคำนวณค่า m^x และเก็บผลลัพธ์ในตัวแปร `y`
 - คำนวณค่าผลลัพธ์ $(1/m) * y$
- ในกรณี `n` เป็น 0, ฟังก์ชันจะแสดงข้อความ ".....Roll Back Point" และคืนค่า 1

```

26 int main() {
27     int M, N;
28     int status = 1;
29     printf("POWER ( M^N ) PROGRAM\n");
30     printf("=====\n");
31     while (status != 2) {
32         printf("[ '1'=Start : '2'=Exit ] : ");
33         scanf("%d", &status);
34         if(status != 2) {
35             // Input
36             printf("Enter Number M : ");
37             scanf("%d", &M);
38             printf("Enter Number N : ");
39             scanf("%d", &N);
40             // Display
41             double ans = power(M, N);
42             printf("\nAnswer M^N = %lf\n", ans);
43             printf("-----Finished\n");
44             getch();
45         }
46     }
47     return 0;
48 }

```

`int main()`: เป็นฟังก์ชันหลักที่เริ่มการทำงานของโปรแกรม ในส่วนนี้มีเมนูเพื่อให้ผู้ใช้เลือกทำงานหรือออกจากโปรแกรม:

- ใช้ตัวแปร `status` เพื่อเก็บค่าสถานะในการทำงานของโปรแกรม ถ้า `status` เท่ากับ 1 แสดงว่าผู้ใช้ต้องการดำเนินการยกกำลัง และถ้า `status` เท่ากับ 2 แสดงว่าผู้ใช้ต้องการออกจากโปรแกรม
- ทำการให้ผู้ใช้ป้อนเลขฐาน `M` และเลขตัวดำเนินการ `N` เพื่อคำนวณการยกกำลัง
- เรียกใช้ฟังก์ชัน `power(M, N)` เพื่อคำนวณและแสดงผลทางหน้าจอ
- ใช้ `getch()` เพื่อรอให้ผู้ใช้กดปุ่มใดๆ เพื่อดูผลลัพธ์และสิ้นสุดรอบการทำงาน

Run Program

```
krits D:\WorkAndProject\C\Data_Structures\Week_05 main pwsh
$ .\power.exe
POWER ( M^N ) PROGRAM
=====
['1'=Start : '2'=Exit] : 1
Enter Number M : 2
Enter Number N : 3
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 M^N = 8.000000
-----Finished
['1'=Start : '2'=Exit] : 1
Enter Number M : 2
Enter Number N : -3
2^-3 = (1/2) * 2^-2
2^-2 = (1/2) * 2^-1
2^-1 = (1/2) * 2^0
.....Roll Back Point
2^-1 = (1/2) * 1.000000 = 0.500000
2^-2 = (1/2) * 0.500000 = 0.250000
2^-3 = (1/2) * 0.250000 = 0.125000

Answer M^N = 0.125000
-----Finished
['1'=Start : '2'=Exit] : 1
Enter Number M : 2
Enter Number N : 0
.....Roll Back Point

Answer M^N = 1.000000
-----Finished
```

- ในส่วนของการยกกำลังเป็นบวก $n > 0$, ฟังก์ชันจะทำงานตามขั้นตอนนี้:
 - แสดงผลลัพธ์ทางหน้าจอว่า $m^n = m * m^x$
 - แสดงผลลัพธ์ทางหน้าจอว่า $m^n = m * y$ โดยใช้ค่า m^x ที่คำนวณมาแล้ว
- ในส่วนของการยกกำลังเป็นลบ $n < 0$, ฟังก์ชันจะทำงานตามขั้นตอนนี้:
 - แสดงผลลัพธ์ทางหน้าจอว่า $m^n = (1/m) * m^x$
 - แสดงผลลัพธ์ทางหน้าจอว่า $m^n = (1/m) * y$ โดยใช้ค่า m^x ที่คำนวณมาแล้ว
- ในกรณี n เป็น 0, ฟังก์ชันจะแสดงข้อความ ".....Roll Back Point" และคืนค่า 1
- และนำคำตอบมาแสดง "Answer M^N = power(M, N)"

สรุปผลการทดลอง

โปรแกรมฟังก์ชันการยกกำลังที่ให้ผู้ใช้นำค่าฐานและตัวดำเนินการเพื่อคำนวณและแสดงผลทางหน้าจอว่าฐานยกกำลังตัวดำเนินการเท่ากับเท่าใด และแสดงขั้นตอนการทำงานของการทำงานของการยกกำลังในแต่ละขั้นตอนด้วยการแสดงข้อความทางหน้าจอ