

Lab 3

1) ให้เขียนการทำงานของการทำงานหาผลบวกของตัวเลขโดดที่ไม่ใช่ 0 ทุกตัวในตัวเลขที่ป้อน ไปเรื่อยๆจนกว่าจะเป็นเลขหลักเดียวเช่นเมื่อป้อน input เป็น 123456 จะได้ผลลัพธ์คือ 123456 -> 21-> 3

Draft Pseudo Code

1.รับ Input ตัวเลขเข้ามา

2.เช็คค่าตัวเลขที่รับเข้ามว่าเป็นเลขโดดหรือไม่

2.1 ถ้าไม่ใช่

1.แบ่งตัวเลขออกเป็นเลขโดด

2.นำเลขโดดที่ได้บวกกันทั้งหมด

3.เก็บค่าที่บวกได้มาคิดซ้ำจนกว่าจะบวกกันแล้วเหลือแค่เลขโดด

4.เมื่อบวกกันจนเหลือแค่เลขโดดแล้วจึงแสดงผล

2.2 ถ้าใช่

1.แสดงผลค่าที่รับเข้ามา

Simple Command Pseudo Code

1.รับ Input ตัวเลขจำนวนเต็มมาเก็บไว้ในตัวแปร n

2.แสดงผลค่าตัวเลขในตัวแปร n ที่รับเข้ามา

3.ในขณะที่ n มากกว่า 9

 แสดงผล ->

 กำหนดให้ sum = 0

 ในขณะที่ n > 0

 นำ n มาหารด้วย 10 แบบเอาเศษแล้วเก็บค่าลงในตัวแปร sum

 นำ n มาหารด้วย 10 เก็บค่าที่ได้ลงในตัวแปร n ใหม่แล้วกลับไปเช็คซ้ำ

ให้ตัวแปร n มีค่าเท่ากับ sum ที่คิดได้

แสดงผล n แล้วกลับไปใช้ซ้ำ

C Program Code

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    printf("%d", n);
    while (n > 9)
    {
        printf(" -> ");
        int sum = 0;
        while (n > 0)
        {
            sum = sum + (n % 10);
            n = n / 10;
        }
        n = sum;
        printf("%d", n);
    }
    return 0;
}
```

2. ให้ $P(x)$ คือผลรวมตัวเลขตั้งแต่ $1..(x-1)$ ที่หาร x ได้ลงตัว เช่น $P(28) = 1+2+4+7+14 = 28$ ให้

เขียนการทำงานในการหาตัวเลข 1-10,000 ว่ามีตัวเลขใดบ้างที่ให้ค่า $P(x)=x$

Draft Pseudo Code

1. กำหนดว่าต้องหาตั้งแต่ตัวเลข 1-10000

2. หาผลรวมของตัวเลขตั้งแต่ 1 จนถึง $x-1$ ที่หาร x ลงตัว

3. เช็คค่าผลรวมของตัวเลขที่หารตัวเลขที่เรากำหนดลงตัวนั้น

เท่ากับเลขที่เรากำหนดหรือไม่

ถ้าใช่ให้แสดงผล x

ถ้าไม่ใช่ให้วนกลับไปหาเลขต่อไปจนกว่าจะถึง 10000

สำหรับ $i=1$ ทำงานซ้ำเมื่อ $i \leq x-1$ แล้ว x เพิ่มขึ้น 1

ถ้า x หารด้วย i แบบเอาเศษแล้ว $= 0$

$sum = sum + i$

Simple Command Pseudo Code

1. สร้างฟังก์ชัน P(x)

กำหนด $sum = 0$ โดยที่ sum คือค่าของ $P(x)$

สำหรับ $i=1$ ทำงานซ้ำเมื่อ $i \leq x-1$ แล้ว i เพิ่มขึ้น 1

ถ้า x หารด้วย i แบบเอาเศษแล้วเท่ากับ 0

ให้ $sum = sum + i$

2. กำหนดตัวเลขที่ต้องการหาเท่ากับ 1-10000

สำหรับ $x=1$ ทำงานซ้ำเมื่อ $x \leq 10000$ แล้ว x เพิ่มขึ้น 1

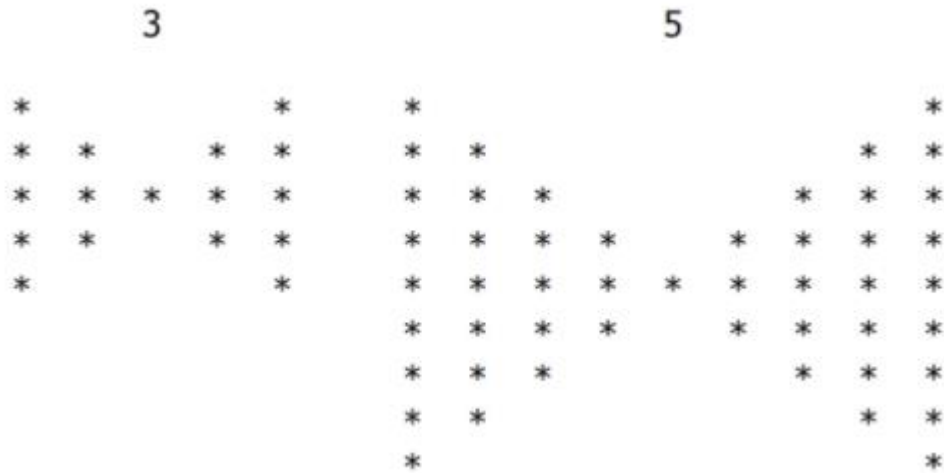
นำฟังก์ชัน $P(x)$ ที่สร้างไว้มาใช้เทียบ

ถ้า $P(x) = x$ ให้แสดงผล x

C Program Code

```
#include<stdio.h>
int P(int x)
{
    int sum = 0;
    for (int i = 1; i <= x-1; i++)
    {
        if (x % i == 0)
        {
            sum = sum + i;
        }
    }
    return sum;
}
int main()
{
    for (int x=1; x <= 10000; x++)
        if (P(x) == x)
        {
            printf("%d\n", x);
        }
    return 0;
}
```

3. ให้เขียนการทำงานของโปรแกรมแสดงรูปผีเสื้อที่มีขนาดตามตัวเลขที่ป้อน



Draft Pseudo Code

- 1.รับ Input ตัวเลขจำนวนเต็มมาเก็บไว้ในตัวแปร n
- 2.นำตัวเลขมาแปลงให้เป็นรูปผีเสื้อโดยขนาดเป็นไปตามค่า n ที่รับเข้ามา
- 3.แสดงผล “ ” และ * ตามสูตรที่กำหนด

Simple Command Pseudo Code

- 1.รับ Input ตัวเลขจำนวนเต็มมาเก็บไว้ในตัวแปร n
- 2.สำหรับ i = 1 ทำงานซ้ำเมื่อ i <= n-1 แล้ว n เพิ่มขึ้น 1

 สำหรับ j = 0 ทำงานซ้ำเมื่อ j < i แล้ว j เพิ่มขึ้น 1

 แสดงผล *

 สำหรับ k = 0 ทำงานซ้ำเมื่อ k < (2*n-2*i)-1 แล้ว k เพิ่มขึ้น 1

 แสดงผล “ ”

 สำหรับ l = 0 ทำงานซ้ำเมื่อ l < i แล้ว l เพิ่มขึ้น 1

 แสดงผล *

 ขึ้นบรรทัดใหม่

3.สำหรับ $i = 0$ ทำงานซ้ำเมื่อ $i < (2*n)-1$ แล้ว i เพิ่มขึ้น 1

แสดงผล *

4.ขึ้นบรรทัดใหม่

5.กำหนดตัวแปร $space = n-1$

6.สำหรับ $i = 0$ ทำงานซ้ำเมื่อ $i \leq n-1$ แล้ว i เพิ่มขึ้น 1

สำหรับ $j = 0$ ทำงานซ้ำเมื่อ $j < space$ แล้ว j เพิ่มขึ้น 1

แสดงผล *

สำหรับ $k = 0$ ทำงานซ้ำเมื่อ $k < (2*i)+1$ แล้ว k เพิ่มขึ้น 1

แสดงผล “ ”

สำหรับ $l = 0$ ทำงานซ้ำเมื่อ $l < space$ แล้ว l เพิ่มขึ้น 1

แสดงผล *

ให้ $space$ ลดลง 1

ขึ้นบรรทัดใหม่

C Program Code

```
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= n - 1; i++)
    {
        for (int j = 0; j < i; j++)
        {
            printf("* ");
        }
        for (int k = 0; k < (2*n-2*i)-1; k++)
        {
            printf(" ");
        }
        for (int l = 0; l < i; l++)
        {
            printf("* ");
        }
        printf("\n");
    }
    for (int i = 0; i < (2*n)-1; i++)
    {
        printf("* ");
    }
    printf("\n");

    int space = n-1;
    for(int i = 0; i<n-1;i++){
        for (int j = 0; j < space; j++)
        {
            printf("* ");
        }
        for (int k = 0; k < (2*i)+1; k++){
            printf(" ");
        }
        for (int l = 0; l < space; l++)
        {
            printf("* ");
        }
        space--;
        printf("\n");
    }
    return 0;
}
```