

1. ให้เขียนการทำงานของการทำงานหาผลบวกของตัวเลขโดดที่ไม่ใช่ 0 ทุกตัวในตัวเลขที่ป้อน ไปเรื่อยๆ จนกว่าจะเป็นเลขหลักเดียวเช่นเมื่อป้อน input เป็น 123456 จะได้ผลลัพธ์คือ 123456 -> 21-> 3

### Draft Pseudo Code

- รับค่าตัวแปรเก็บไว้ใน X
- วนลูปจนกว่าผลบวกของเลขจะเป็นเลขเดียว
- แสดงผลค้ผลรวม



- รับค่าตัวแปรเก็บไว้ใน x
- กำหนด i=0,sum=0
- ทำซ้ำในขณะที่ i<strlen(x)
  - sum += x[i]-'0'
  - i++
- ใช้ sprintf เพื่อย้ายค่า sum ที่เป็น int ไปอยู่ใน y ที่เป็น Array of Character



- ทำซ้ำในขณะที่ strlen(y) != 1
  - i=0,sum=0
  - ทำซ้ำในขณะที่ i<strlen(y)
    - sum+=y[i]-'0'
    - i++
  - ใช้ sprintf เพื่อย้ายค่า sum ที่เป็น int ไปอยู่ใน y ที่เป็น Array of Character

### Simple Command Pseudo Code

- กำหนดตัวแปร x และ y เป็น Array of Character 1,000,100 ช่อง ไว้เป็น  
Global Variable
- รับค่าตัวแปรเก็บไว้ใน x
- กำหนด i=0,sum=0
- ทำซ้ำในขณะที่ i<strlen(x)
  - sum += x[i]-'0'
  - i++
- ใช้ sprintf เพื่อย้ายค่า sum ที่เป็น int ไปอยู่ใน y ที่เป็น Array of  
Character
- ทำซ้ำในขณะที่ strlen(y) != 1
  - i=0,sum=0
  - ทำซ้ำในขณะที่ i<strlen(y)
    - sum+=y[i]-'0'
    - i++
  - ใช้ sprintf เพื่อย้ายค่า sum ที่เป็น int ไปอยู่ใน y ที่เป็น Array of  
Character
- แสดงผลค่า y

## C Program

```
#include <stdio.h>
#include <string.h>
char x[1000100];
char y[1000100];
int main()
{
    int sum = 0;
    scanf("%s", x);
    for (int i = 0; i < strlen(x); i++)
    {
        sum += x[i] - '0';
    }
    sprintf(y, "%d", sum);
    while (strlen(y) != 1)
    {
        sum = 0;
        for (int i = 0; i < strlen(y); i++)
        {
            sum+=y[i] - '0';
        }
        sprintf(y, "%d", sum);
    }
    printf("%s",y);
    return 0;
}
```

2. ให้  $P(x)$  คือผลรวมตัวเลขตั้งแต่  $1..(x-1)$  ที่หาร  $x$  ได้ลงตัว เช่น  $P(28) = 1+2+4+7+14 = 28$  ให้เขียนการทำงานในการหาตัวเลข 1-10,000 ว่ามีตัวเลขใดบ้างที่ให้ค่า  $P(x)=x$

### Draft Pseudo Code

- วนลูปหาค่าผลรวมของตัวเลขที่หาร  $x$  ลงตัว
- ถ้าผลรวมของตัวเลขเท่ากับ  $P(x)$  ให้แสดงผลค่านั้น

- ทำซ้ำในขณะที่  $i < 10000$ 
  - กำหนดค่า  $sum = 0, j = 1$
  - ทำซ้ำในขณะที่  $j < i$ 
    - ถ้า  $i \% j == 0$ 
      - $sum += j$
    - $j++$
  - $i++$

- ถ้า  $sum == i$ 
  - แสดงผลค่า  $sum$

## Simple Command Pseudo Code

- กำหนดค่า  $i=1$
- ทำซ้ำในขณะที่  $i < 10000$ 
  - กำหนดค่า  $sum = 0, j=1$
  - ทำซ้ำในขณะที่  $j < i$ 
    - ถ้า  $i \% j == 0$ 
      - $Sum += j$
      - $j++$
  - ถ้า  $sum == i$ 
    - แสดงผลค่า  $sum$
  - $i++$

## C Program

```
#include <stdio.h>
int main()
{
    //Original By Barw
    for (int i = 1; i <= 10000; i++)
    {
        int sum = 0;
        for (int j = 1; j < i; j++)
        {
            if (i % j == 0)
            {
                sum += j;
            }
        }
        if (sum == i)
        {
            printf("%d ", sum);
        }
    }
    return 0;
}
```



ดูופןนที่เป็น 3 เหลี่ยม 2 รูป

- กำหนดค่า  $i = 1$
- ทำซ้ำในขณะที่  $i \leq x-1$ 
  - กำหนดค่า  $j=0, k=0, l=0$
  - ทำซ้ำในขณะที่  $j < i$ 
    - แสดงผล “\* ”
    - $j++$
  - ทำซ้ำในขณะที่  $k < (2*(x-i))-1$ 
    - แสดงผล “ ”
    - $k++$
  - ทำซ้ำในขณะที่  $l < i$ 
    - แสดงผล “\* ”
    - $l++$
  - แสดงผล Newline
  - $i++$

## ดูปลากลาง 2x-1

- กำหนดค่า  $i=0$
- ทำซ้ำในขณะที่  $i < (x*2)-1$ 
  - แสดงผล "\*" "
  - $i++$
- แสดงผล Newline

ดูปล่า่งที่เป็น 3 เหลี่ยมมกลับหัว 2 รูป

- กำหนดค่า  $dot = x-1, i=0$
- ทำซ้ำในขณะที่  $i < x-1$ 
  - กำหนดค่า  $j=0, k=0, l=0$
  - ทำซ้ำในขณะที่  $j < dot$ 
    - แสดงผล "\*" "
    - $j++$
  - ทำซ้ำในขณะที่  $k < 2*i+1$ 
    - แสดงผล " "
    - $k++$
  - ทำซ้ำในขณะที่  $l < dot$ 
    - แสดงผล "\*" "
    - $l++$
  - $dot--$
  - แสดงผล Newline
  - $i++$



## Simple Command Pseudo Code

- รับค่าตัวแปรเก็บไว้ใน X
- กำหนดค่า  $i = 1$
- ทำซ้ำในขณะที่  $i \leq x-1$ 
  - กำหนดค่า  $j=0, k=0, l=0$
  - ทำซ้ำในขณะที่  $j < i$ 
    - แสดงผล "\*" "
    - $j++$
  - ทำซ้ำในขณะที่  $k < (2*(x-i))-1$ 
    - แสดงผล " "
    - $k++$
  - ทำซ้ำในขณะที่  $l < i$ 
    - แสดงผล "\*" "
    - $l++$
  - แสดงผล Newline
  - $i++$
- กำหนดค่า  $i=0$
- ทำซ้ำในขณะที่  $i < (x*2)-1$ 
  - แสดงผล "\*" "
  - $i++$
- แสดงผล Newline

- กำหนดค่า  $\text{dot} = x-1, i=0$
- ทำซ้ำในขณะที่  $i < x-1$ 
  - กำหนดค่า  $j=0, k=0, l=0$
  - ทำซ้ำในขณะที่  $j < \text{dot}$ 
    - แสดงผล “\* ”
    - $j++$
  - ทำซ้ำในขณะที่  $k < 2*i+1$ 
    - แสดงผล “ ”
    - $k++$
  - ทำซ้ำในขณะที่  $l < \text{dot}$ 
    - แสดงผล “\* ”
    - $l++$
  - $\text{dot}--$
  - แสดงผล Newline
  - $i++$

## C Program

```
#include <stdio.h>

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