ปฏิบัติการครั้งที่ 5

UNIT AND INTEGRATION TESTING

Top down
Bottom up
Sandwich Testing

Unit Testing

- ทดสอบการทำงานทีละหน่วยหรือโมดูล
- ต้องการ โมดูลไดร์เวอร์ (Driver module) ในการทดสอบ
- โมดูลไดร์เวอร์ ทำหน้าที่
  - กำหนดค่าเริ่มต้นให้กับพารามิเตอร์ของโมดูลที่ถูกทดสอบ
  - เรียกโมดูลที่ต้องการทดสอบด้วยส่งผ่านค่าพารามิเตอร์ที่โมดูลนั้นต้องการ
  - รับค่ากลับ ที่เป็นผลจากโมดูลที่ถูกทดสอบ
- ในการทดสอบอาจจำเป็นต้องเขียนกลุ่มหรือชุดคำสั่งที่เขียนเพื่อแทนโมดูล เรียกกลุ่มคำสั่งนี้ว่า Stub

Demo Stub

- จงพิจารณาผลการทำงานของฟังก์ชันต่อไปนี้ หากผลการทำงานไม่ถูกต้องสมบูรณ์ให้ปรับแก้ จนกระทั่งได้ ฟังก์ชันที่ทำงานถูกต้องในทุกๆ กรณี

```
void  count_Down (int N) {
        while (N > 0) {
           printf("%d\t", N);
           N--;
        }
        printf("\n...BLAST OFF....\a");
    }
```

Integration Testing

- ทดสอบการทำงานเมื่อมีการผูกรวมโมดูลเข้าด้วยกัน โดยการเพิ่มเข้าทีละโมดูล
- สามารถทดสอบการทำงานทั้งในสภาวการณ์ปกติ และกรณีที่โปรแกรมอาจจะมีปัญหาหรือเป็นกรณียกเว้น
- ข้อผิดพลาดที่อาจตรวจพบได้
  - ส่วนต่อประสานที่ไม่สอดคล้องกัน (Interface incompatibility)
  - การส่งผ่านค่าที่ไม่ถูกต้อง (Incorrect parameter values) เช่น ผิดชนิด หรือผิดสถานะ หรือผิดความหมาย เป็นต้น
  - Run-time exceptions
  - การตอบสนองหรือลักษณะการทำงานของโปรแกรมซึ่งไม่คาดว่าจะเกิดขึ้น (Unexpected state interactions)

Demo Stub

```
void  Count_Down (int N) {
      if (N <= 0)
        printf("The lower bound is reached...");
     else
        printf("Reduction instructions are placed here....");
  }
```

จงเขียนฟังก์ชัน BitPrint() ซึ่งมีพารามิเตอร์เป็นข้อมูลชนิดจำนวนเต็ม เพื่อทำการแสดงค่าแต่ละบิตของพารามิเตอร์นั้น

```
#include <stdio.h>
#include <limits.h>
void BitPrint(int );
void main() {
  int c;
  printf("Integer to be displayed its bit representation:");
  scanf("%d", &c);    BitPrint(c);
}
/* Bit print of an integral expression*/

}
```