**แบบฝึกปฏิบัติการครั้งที่ ๕ การทดสอบและการแก้จุดบกพร่อง**

**จุดประสงค์** เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถ

1. ใช้เครื่องมือช่วยในการทำความเข้าใจการทำงานของโปรแกรมได้
2. หาจุดบกพร่องและปรับแก้โปรแกรมได้
3. สร้างชุดข้อมูลทดสอบที่มีคุณภาพได้
4. พัฒนาโปรแกรมที่ซับซ้อนได้

**กิจกรรม** อาจารย์ผู้คุมปฏิบัติการทบทวนและแนะนำ

1. วิธีการทดสอบโปรแกรม (Unit testing และ integrated testing)
2. การสร้างชุดข้อมูลทดสอบ(Test Case = Case name + test data + expected result)

นักศึกษา

1. ทดลองสั่งดำเนินการโปรแกรมตัวอย่าง แล้วสังเกตุผลจากการรันโปรแกรม
2. ทดลองใช้คำสั่งหรือฟังก์ชันตามที่อาจารย์แนะนำ สังเกตุและบันทึกผลจากการรันโปรแกรม
3. ทำแบบฝึกหัด

**คำสั่ง**

1. ข้อ 1 ให้จับกลุ่ม 2 คน พิจารณาโปรแกรม และทำตามคำสั่งในแต่ละข้อ เขียนสรุปในรูปแบบรายงาน โดยใส่ชื่อเพื่อนในกลุ่ม ลงในต้นไฟล์ตามรูปแบบที่ระบุใน Website ของรายวิชา
2. ข้อ 2 ให้เขียนโปรแกรมส่งเป็นงานเดี่ยว

**แบบฝึกหัด**

1. (Lab05\_1\_5xxxxxxxx.pdf)
2. ฟังก์ชัน count\_down\_2\_steps**(***int* n**)** ทำหน้าที่ นับถอยหลังจากจำนวนเต็ม *n* โดยมีการนับลดลงทีละ 2 จนถึง 0 และจบการทำงาน
   1. Black Box Testing: ให้เขียนกรณีทดสอบโดยพิจารณาจาก Specification ของโจทย์ โดยแต่ละ Test case จะต้องประกอบด้วย Case name, Test data และ Expected Result พร้อมคำอธิบายเหตุผล
   2. White Box Testing: พิจารณาฟังก์ชัน count\_down\_2\_steps**(***int* n**)** ด้านล่างแล้วสร้างกรณีทดสอบด้วย Basis Path Testing พร้อมวาดผังงานพร้อมระบุ path ในแต่ละกรณีทดสอบ

|  |
| --- |
| *void* count\_down\_2\_steps**(***int* n**)**  **{**  **if** **(**n **==** 0**)**  printf**(**""**);**  **else** **{**  printf**(**"%d\t"**,** n**);**  n **-=** 2**;**  **}**  **}** |

* 1. ให้ทำการทดสอบโปรแกรมด้วยกรณีทดสอบจากข้อ i และ ii แล้วเขียนผลที่ได้ลงในตารางในรูปแบบ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case name | Test Data | Expected Result | Test Result | Passed/Failed |

หากโปรแกรมทำงานผิดพลาดให้ปรับแก้โปรแกรมกระทั่งได้ฟังก์ชันที่ทำงานถูกต้องในทุกๆ กรณี แล้วเขียนชุดคำสั่งที่ปรับแก้ลงในช่องด้านล่าง

|  |
| --- |
|  |

1. ให้นักศึกษาทำความเข้าใจการทำงานของโปรแกรมต่อไปนี้ แล้วติดตามดูค่าของตัวแปร *x* และ *y* แล้วบันทึกผลการรันเทียบกับแต่ละบรรทัดที่ทำการพิมพ์ หากบรรทัดคำสั่งนั้นถูกดำเนินการหลายครั้ง ให้เขียน  *n* กำกับด้วย โดย *n* หมายถึงครั้งที่ถูกดำเนินการ

|  |  |
| --- | --- |
| ***#include <stdio.h>***  *int* A**(***void***)**  **{**  **return** **(**getchar**()** **+** 5**);**  **}**  *void* B**(***int* x**)**  **{**  putchar**(**x**);** x **+=** 1**;**  **}**  *int* C**(***int* x**,** *int* y**)**  **{**  printf**(**"\nSub C:%d %d\n"**,** x**,** y**);**  **return** x **+** y **;**  **}**  *int* main**()**  **{**  *int* x **=** 8**,** y**;**  B**(**A**());**  y **=** C**(**x**++,** A**());**  printf**(**"main :%d %d\n"**,** x**,** y**);**  y **=** C**(++**x**,** A**()** **/** 2**);**  printf**(**"main :%d %d\n"**,** x**,** y**);**  **return** 0**;**  **}** |  |

1. (Lab05\_2\_5XXXXXXXX.c) ให้เขียนโปรแกรมช่วยคำนวณเกรดเฉลี่ยของนักศึกษาไม่ทราบจำนวน โดยรับข้อมูลเกรดของนักศึกษาจากไฟล์ ในรูปแบบ input redirection โดยรายละเอียดของข้อมูลแต่ละบรรทัดเป็นผลการเรียน 1 ภาคการศึกษาของนักเรียนหนึ่ง นักศึกษาแต่ละคนลงทะเบียนไม่จำกัดจำนวนวิชา และทุกวิชามีจำนวนหน่วยกิตเท่ากันคือ 3 หน่วยกิต และเกรดที่เป็นไปได้ คือ A/B/C/D/F เท่านั้น

|  |  |
| --- | --- |
| **Input** | **Output** |
| A A B C A  F F B C C D A  B | 3.40  1.71  3.00 |