

1. Ascii_Display (20 คะแนน)

จงเขียนโปรแกรมภาษา C++ ซึ่งประกอบด้วยอย่างน้อย 4 ฟังก์ชัน ได้แก่ main(), Ascii_AllZ(), Ascii_allz() และ Ascii_num() โดยโปรแกรมมีการทำงานดังต่อไปนี้ (ตั้งชื่อไฟล์โปรแกรมว่า CS102Hw4_1_XXXXXXXXXX.c)

- ฟังก์ชัน main() แสดงเมนูให้ผู้ใช้เลือก ดังนี้

```
##### MENU #####
```

 1. Ascii table of characters A-Z
 2. Ascii table of characters a-z
 3. Ascii table of code 65-90
 0. Quit Program

Enter your choice <1, 2, 3 or 0> :
- ถ้าผู้ใช้ป้อนตัวเลือกไม่ถูกต้องให้แสดงข้อความ “Invalid Choice!!!” จากนั้นให้แสดงข้อความเพื่อรอรับตัวเลือกจากผู้ใช้
อีก จนกว่าจะป้อนตัวเลือกที่ถูกต้อง
- ถ้าผู้ใช้ป้อนตัวเลือก 1 ให้ฟังก์ชัน main() เรียกใช้ฟังก์ชัน Ascii_AllZ() ซึ่งจะแสดงตารางแอสกีของตัวอักษร A ถึง Z (ดูรูปแบบในตัวอย่างการรันข้างล่าง) ทั้งนี้ ให้นักศึกษากำหนดชนิดฟังก์ชัน จำนวนและชนิดของพารามิเตอร์เองตามที่เหมาะสม
จากนั้นให้แสดงเมนูอีก
- ถ้าผู้ใช้ป้อนตัวเลือก 2 ให้ฟังก์ชัน main() เรียกใช้ฟังก์ชัน Ascii_allz() ซึ่งจะแสดงตารางแอสกีของตัวอักษร a ถึง z (ดูรูปแบบในตัวอย่างการรันข้างล่าง) ทั้งนี้ ให้นักศึกษากำหนดชนิดฟังก์ชัน จำนวนและชนิดของพารามิเตอร์เองตามที่เหมาะสม
จากนั้นให้แสดงเมนูอีก
- ถ้าผู้ใช้ป้อนตัวเลือก 3 ให้ฟังก์ชัน main() เรียกใช้ฟังก์ชัน Ascii_num() ซึ่งจะแสดงตารางแอสกีตั้งแต่รหัส 65 ถึง 90 (ดูรูปแบบในตัวอย่างการรันข้างล่าง) ทั้งนี้ ให้นักศึกษากำหนดชนิดฟังก์ชัน จำนวนและชนิดของพารามิเตอร์เองตามที่เหมาะสม
จากนั้นให้แสดงเมนูอีก
- ถ้าผู้ใช้ป้อนตัวเลือก 0 ให้แสดงคำว่า Bye!! แล้วจบโปรแกรม

ตัวอย่างการทำงานของโปรแกรม (ตัวอย่างนี้ จากการรันรอบเดียว ตัวอักษรสีแดงหมายถึงข้อมูลเข้า)

```
##### MENU #####
```

```
1. Ascii table of characters A-Z
```

```
2. Ascii table of characters a-z
```

```
3. Ascii table of code 65-90
```

```
0. Quit Program
```

```
Enter your choice <1, 2, 3 or 0> : 4
```

```
Invalid Choice!!!
```

```
Enter your choice <1, 2, 3 or 0> : 8
```

```
Invalid Choice!!!
```

```
Enter your choice <1, 2, 3 or 0> : 1
```

```
-----
```

```
Ascii Table for A-Z
```

```
-----
```

```
A - 65
```

```
B - 66
```

C – 67

... (ละบรรทัดที่แสดงรหัสของตัว D ถึง Y ไว้)

Z – 90

MENU

1. Ascii table of characters A-Z

2. Ascii table of characters a-z

3. Ascii table of code 65-90

0. Quit Program

Enter your choice <1, 2, 3 or 0> : 2

Ascii Table for a-z

a – 97

b – 98

c – 99

... (ละบรรทัดที่แสดงรหัสของตัว d ถึง y ไว้)

z – 122

MENU

1. Ascii table of characters A-Z

2. Ascii table of characters a-z

3. Ascii table of code 65-90

0. Quit Program

Enter your choice <1, 2, 3 or 0> : 3

Ascii Table for code 65-90

65 – A

66 – B

67 – C

... (ละบรรทัดที่แสดงรหัสของตัว 68 ถึง 89 ไว้)

90 – Z

MENU

1. Ascii table of characters A-Z

2. Ascii table of characters a-z

3. Ascii table of code 65-90

0. Quit Program

Enter your choice <1, 2, 3 or 0> : 0

Bye!!

2. จตุรัสมหัศจรรย์ (20 คะแนน)

ตารางจตุรัสมหัศจรรย์ (Magic Square) คือ ตารางจัตุรัสของเลขจำนวนเต็มซึ่งมีตัวเลขแต่ละตัวในตารางไม่ซ้ำกันเลย และมีผลรวมของตัวเลขทุกแถว ทุกคอลัมน์ และทุกแนวทแยงมุมเท่ากันทั้งหมด เรียกค่าผลรวมนี้ว่า **Magic Constant**

ให้นักศึกษาเขียนโปรแกรมภาษา C เพื่อรับขนาดตารางเข้ามาเป็นจำนวนเต็ม N (กำหนดให้ N มีค่าเป็นไปได้ตั้งแต่ 3-20) และรับข้อมูลตัวเลขจำนวนเต็มภายในตารางจำนวน $N * N$ ตัว และตรวจสอบว่าตารางจัตุรัสที่รับเข้ามาเป็นจตุรัสมหัศจรรย์หรือไม่

ถ้าตารางนั้นเป็น Magic Square ให้แสดงข้อความที่หน้าจอผลลัพธ์บอกว่า เป็น Magic Square แล้วตามด้วยค่าผลรวมของแนวใดแนวหนึ่ง (ค่า Magic Constant) พร้อมทั้งแสดงค่าของตารางนั้นลงในไฟล์ผลลัพธ์ชื่อ **myMagic.txt** ด้วย แต่ถ้าตารางนั้นไม่ใช่ Magic Square ให้แสดงข้อความบอกว่าไม่ใช่ แล้วตามด้วยค่าผลรวมตัวเลขของแนว (แนวแถว แนวคอลัมน์ หรือแนวทแยงมุมก็ได้) ที่มีค่าผลรวมมากที่สุดออกมา

โดยกำหนดให้โปรแกรมของนักศึกษาต้องประกอบด้วยฟังก์ชันต่อไปนี้ ทั้งนี้ นักศึกษาสามารถกำหนดชนิดของฟังก์ชัน จำนวน และชนิดของพารามิเตอร์เอง และอาจเพิ่มเติมฟังก์ชันอื่นๆ ได้ตามที่เห็นสมควร ตั้งชื่อไฟล์โปรแกรมว่า **CS102Hw4_2_XXXXXXXXXX.c**

ก. **main ()** เรียกใช้ฟังก์ชันอื่นๆ

ข. **readSquare ()** รับค่าจำนวนเต็มมาใส่ลงในตารางตัวเลข

ค. **findMaxSum ()** รับพารามิเตอร์เป็นตารางตัวเลขจำนวนเต็ม แล้วส่งค่ากลับเป็นผลรวมตัวเลขของแนว (แนวแถว แนวคอลัมน์ หรือแนวทแยงมุมก็ได้) ที่มีค่าผลรวมมากที่สุดในตารางนั้น

ตัวอย่างการทำงานของโปรแกรม (ตัวอักษรสีแดงหมายถึงข้อมูลเข้า)

ตัวอย่างการรันครั้งที่ 1

Enter size of your square: 4

Enter your square:

8 5 3 3

1 6 5 1

7 2 7 4

4 6 1 9

Your square is NOT a magic square.

The maximum sum is 30.

ตัวอย่างการรันครั้งที่ 2

Enter size of your square: 3

Enter your square:

6 1 8

7 5 3

2 9 4

Your square is a magic square!!

The magic constant is 15.

และในไฟล์เดอร์เดียวกับที่ไฟล์โปรแกรมอยู่ จะต้องมีไฟล์ชื่อ **myMagic.txt** ซึ่งมีข้อมูลดังนี้

This is a 3x3 magic square.

The magic constant is 15.

6 1 8

7 5 3

2 9 4

โปรแกรมตัดเกรดนักศึกษา

ข้อมูลต่อไปนี้ ใช้สำหรับเขียนโปรแกรมการบ้าน ข้อ 3. และ 4.

จงเขียนโปรแกรมภาษา C สำหรับตัดเกรดวิชา คพ.102 โดย

- อ่านข้อมูลนักศึกษาซึ่งเก็บอยู่ในแฟ้มข้อมูล (File) ชื่อ **scores.txt** โดยแต่ละรายการประกอบด้วย รหัสนักศึกษา คะแนนสอบกลางภาค คะแนนสอบปลายภาค คะแนนการบ้าน มาเก็บในอาร์เรย์ (array) โดยคะแนนที่เก็บเป็นคะแนนสุทธิที่ใช้ในการพิจารณาตัดเกรดโดยคิดจากคะแนนรวมทั้งหมด 100 คะแนน แบ่งเป็น
 - คะแนนสอบกลางภาค 40%
 - คะแนนสอบปลายภาค 40%
 - คะแนนการบ้าน 20%
- ประมวลผลข้อมูลตัดเกรดนักศึกษาโดยใช้คะแนนรวมพิจารณาจากเกณฑ์ ดังนี้
 - คะแนนตั้งแต่ 80 ขึ้นไป → เกรด A
 - คะแนนตั้งแต่ 75 – 80 (ไม่รวม 80) → เกรด B
 - คะแนนตั้งแต่ 65 – 75 (ไม่รวม 75) → เกรด C
 - คะแนนตั้งแต่ 55 – 65 (ไม่รวม 65) → เกรด D
 - คะแนนน้อยกว่า 55 → เกรด F
- พิมพ์รายงานการประมวลผลเกรดนักศึกษา โดยสร้างแฟ้มข้อมูลผลลัพธ์ ชื่อ **Student_Grade.out** ซึ่งในแต่ละแถว (ระเบียน) ของแฟ้มข้อมูลประกอบด้วยข้อมูล ดังต่อไปนี้
 - รหัสนักศึกษา
 - คะแนนทั้ง 3 ส่วนของนักศึกษา
 - คะแนนรวมของนักศึกษา
 - เกรดที่ได้

คำสั่ง จากข้อมูลข้างต้น ให้เขียนโปรแกรม ข้อ 3. และ 4. โดยกำหนดให้นักศึกษาเขียนโปรแกรมตามขั้นตอนต่อไปนี้

3. (30 คะแนน) จงเขียนโปรแกรมเพื่อทดสอบการอ่านข้อมูลจากไฟล์มาใส่ในอาร์เรย์โดย

- a. เขียนฟังก์ชัน **input_data()** ซึ่งเป็นฟังก์ชันที่มีพารามิเตอร์ 3 ตัว ได้แก่ อาร์เรย์หนึ่งมิติสำหรับเก็บรหัสนักศึกษาที่อ่านได้จากไฟล์ อาร์เรย์สองมิติสำหรับเก็บคะแนนทั้งสามส่วนและคะแนนรวม (คะแนนสอบกลางภาค คะแนนสอบปลายภาค คะแนนการบ้าน) และตัวแปรไฟล์นำเข้าข้อมูล ฟังก์ชันจะต้องเปิดไฟล์นำเข้าและอ่านข้อมูลจากไฟล์เฮดสามเก็บที่อาร์เรย์ 1 มิติ และคะแนนทั้งสามส่วนมาเก็บที่อาร์เรย์ 2 มิติ พร้อมทั้งคำนวณหาคะแนนรวมใส่ในคอลัมน์ที่ 4 จนจบไฟล์ พร้อมทั้งนับข้อมูลที่อ่านเข้ามาได้ว่ามีกี่รายการ แล้วส่งจำนวนที่นับได้เป็นค่าผลลัพธ์กลับ
- b. เขียนฟังก์ชัน **print_data()** ซึ่งเป็นฟังก์ชันที่รับพารามิเตอร์ 3 ตัว เป็นอาร์เรย์ 1 มิติ และ 2 มิติ และจำนวนรายการที่ต้องการพิมพ์ แล้วทำการพิมพ์ข้อมูลในอาร์เรย์ เฉพาะ รหัสนักศึกษาและคะแนนทั้งสี่ส่วน (คะแนนสอบกลางภาค คะแนนสอบปลายภาค คะแนนการบ้าน และคะแนนรวม) ตามจำนวนรายการที่รับเข้ามาเป็นพารามิเตอร์
- c. ในส่วนของ **main()** ให้ประกาศอาร์เรย์คู่ขนาน 2 ตัว เป็นอาร์เรย์ 1 มิติสำหรับเก็บรหัสนักศึกษา และ 2 มิติ ขนาด 100 x 4 สำหรับเก็บคะแนนสอบสามส่วน (คะแนนสอบกลางภาค คะแนนสอบปลายภาค คะแนนการบ้าน) และคะแนนรวม พร้อมทั้งกำหนดค่าเริ่มต้นให้เป็น 0 จากนั้นเรียกใช้ฟังก์ชัน **input_data()** เพื่ออ่านข้อมูลจากไฟล์มาเก็บในอาร์เรย์ และเก็บจำนวนรายการที่อ่านได้ในตัวแปรชื่อ **numStudent** แล้วเรียกใช้ **print_data()** เพื่อแสดงผลออกทางจอภาพ
- d. ทดสอบการทำงานให้ถูกต้องแล้วส่งโปรแกรมในข้อนี้โดย ใช้ชื่อ **CS102Hw4_3_XXXXXXXXXX.c** ในกล่องรับการบ้าน 4. ข้อ 3. (XXXXXXXXXX หมายถึงเลขทะเบียนนักศึกษา)

4. (30 คะแนน) จงเขียนโปรแกรมเพื่อทดสอบฟังก์ชันการคำนวณค่าคะแนนรวม และการคำนวณเกรดโดยใช้โปรแกรมที่ได้ในข้อ 1 มาแก้ไขเพิ่มเติมดังนี้
- e. ฟังก์ชัน **compute_grade()** คำนวณหาค่าเกรดของนักศึกษาแต่ละคน โดยรับค่าพารามิเตอร์เป็นอาร์เรย์สองมิติที่มี 4 คอลัมน์ และแถวที่ต้องการหาเกรด และส่งผลลัพธ์เป็นค่าเกรดที่ได้กลับมายังผู้เรียกใช้ (ฟังก์ชันจะให้ผลลัพธ์กลับมาแค่เกรดของรายการที่ส่งไปให้คำนวณ) เช่น ต้องการส่งค่า score[100][4] ไปหาค่าเกรดของรายการที่ 3 ก็เรียกใช้ compute_grade(score, 3) ผลลัพธ์ที่ได้กลับมามีค่าเป็นเกรด เช่น 'A'
 - f. แก้ไขฟังก์ชัน **print_data()** ให้มีพารามิเตอร์ 4 ค่าได้แก่ อาร์เรย์ 1 มิติ (เก็บรหัสนักศึกษา), อาร์เรย์ 2 มิติ (เก็บคะแนนสอบและคะแนนรวม), จำนวนรายการที่ต้องการพิมพ์ และ ตัวแปรไฟล์สำหรับพิมพ์ผลลัพธ์ ให้ฟังก์ชันนี้เรียกใช้ฟังก์ชัน compute_grade() เพื่อคำนวณเกรดของแต่ละรายการ แล้วพิมพ์ข้อมูลรหัสนักศึกษา คะแนนทั้งสามส่วน คะแนนรวม และเกรดที่ได้รับ ตามจำนวนรายการที่รับเข้ามาเป็นพารามิเตอร์ แต่ผลลัพธ์ที่พิมพ์ให้เก็บไว้ที่ไฟล์
 - g. เขียนฟังก์ชัน **average_score()** รับพารามิเตอร์ 3 ตัวเป็นอาร์เรย์ 2 มิติ 1 ตัวที่มี 4 คอลัมน์ ซึ่งเก็บค่าคะแนน 4 ส่วน พารามิเตอร์ตัวที่สองเป็นค่าจำนวนรายการที่ต้องการให้คำนวณ ส่วนพารามิเตอร์ตัวที่ 3 สำหรับระบุคอลัมน์ที่ต้องการหาค่าเฉลี่ย จากนั้นนำมาคำนวณค่าเฉลี่ยของคะแนนตามคอลัมน์ที่ระบุ แล้วส่งค่าผลลัพธ์กลับเป็นค่าเฉลี่ยที่ได้ ตัวอย่างเช่น ต้องการหาค่าเฉลี่ยของคะแนนรวม score[100][4] ซึ่งอยู่ที่คอลัมน์ที่ 4 และมี 50 รายการ ก็เรียกใช้ average_score(score, 50, 3) ผลลัพธ์ที่ได้กลับมาคือค่าเฉลี่ยของคะแนนรวม 50 รายการ
 - h. เขียนฟังก์ชัน **min_score()** รับพารามิเตอร์ 3 ตัวเป็นอาร์เรย์ 2 มิติ 1 ตัวที่มี 4 คอลัมน์ ซึ่งเก็บค่าคะแนน 4 ส่วน พารามิเตอร์ตัวที่สองเป็นค่าจำนวนรายการที่ต้องการให้คำนวณ ส่วนพารามิเตอร์ตัวที่ 3 สำหรับระบุคอลัมน์ที่ต้องการหาค่าต่ำสุด ส่งค่าผลลัพธ์กลับเป็นค่าต่ำสุดที่ได้
 - i. เขียนฟังก์ชัน **max_score()** รับพารามิเตอร์ 3 ตัวเป็นอาร์เรย์ 2 มิติ 1 ตัวที่มี 4 คอลัมน์ ซึ่งเก็บค่าคะแนน 4 ส่วน พารามิเตอร์ตัวที่สองเป็นค่าจำนวนรายการที่ต้องการให้คำนวณ ส่วนพารามิเตอร์ตัวที่ 3 สำหรับระบุคอลัมน์ที่ต้องการหาค่าสูงสุด ส่งค่าผลลัพธ์กลับเป็นค่าสูงสุดที่ได้
 - j. ในส่วนของ main() ให้ทำงานดังต่อไปนี้
 - เปิดใช้ไฟล์สำหรับเป็นอ่านค่าข้อมูลเข้ามาใช้ (scores.txt) และไฟล์สำหรับเก็บผลลัพธ์ (Student_Grade.out)
 - ประกาศอาร์เรย์คู่ขนาน 2 ตัว เป็นอาร์เรย์ 1 มิติสำหรับเก็บรหัสนักศึกษา และ 2 มิติ ขนาด 100 x 4 สำหรับเก็บคะแนนสอบสามส่วน (คะแนนสอบกลางภาค คะแนนสอบปลายภาค คะแนนการบ้าน) และคะแนนรวม พร้อมทั้งกำหนดค่าเริ่มต้นให้เป็น 0
 - เรียกใช้ฟังก์ชัน **input_data()** เพื่ออ่านข้อมูลจากไฟล์ scores.txt มาเก็บในอาร์เรย์และเก็บจำนวนรายการที่อ่านได้ในตัวแปรชื่อ numStudent
 - เรียกใช้ **print_data()** เพื่อพิมพ์ข้อมูลคะแนนและเกรดของทุกคน ไปที่ไฟล์ Student_Grade.out
 - เรียกใช้ฟังก์ชัน หาค่าเฉลี่ย ค่าต่ำสุด ค่าสูงสุด ของคะแนนสอบกลางภาค คะแนนสอบปลายภาค คะแนนการบ้าน และคะแนนรวม พร้อมทั้งพิมพ์ผลลัพธ์ที่ไฟล์ Student_Grade.out
 - k. ทดสอบการทำงานให้ถูกต้องแล้วส่งโปรแกรมในข้อนี้โดย ใช้ชื่อ CS102Hw4_4_XXXXXXXXXX.c ในกล่องรับการบ้าน 4 ข้อ 4 (XXXXXXXXXX หมายถึงเลขทะเบียนนักศึกษา)

หมายเหตุ

อาร์เรย์คู่ขนาน (Parallel Arrays) คือ กลุ่มของอาร์เรย์ที่จัดเก็บข้อมูลซึ่งมีความสัมพันธ์กันแต่เป็นข้อมูลต่างประเภทกันทำให้ไม่สามารถจัดเก็บในอาร์เรย์เดียวกันได้ ยกตัวอย่างเช่น อาร์เรย์ซึ่งเก็บเลขทะเบียนนักศึกษา (Stu_ID_Array) และ อาร์เรย์ซึ่งเก็บคะแนนสอบกลางภาคของนักศึกษาแต่ละคน (Score_Array) ดังแสดงในภาพตัวอย่างด้านล่างนี้

Stu_ID_Array	Score_Array	
5120610010	50.25	คะแนนของนักศึกษาเลขทะเบียน 5120610010
5120610024	38.75	คะแนนของนักศึกษาเลขทะเบียน 5120610024
5120610037	52.50	คะแนนของนักศึกษาเลขทะเบียน 5120610037
5120610045	45.23	คะแนนของนักศึกษาเลขทะเบียน 5120610045
5120610053	12.8	คะแนนของนักศึกษาเลขทะเบียน 5120610053
5120610061	20.05	คะแนนของนักศึกษาเลขทะเบียน 5120610061
5120610077	31.84	คะแนนของนักศึกษาเลขทะเบียน 5120610077
5120610086	28.20	คะแนนของนักศึกษาเลขทะเบียน 5120610086
⋮	⋮	

ภาพแสดงตัวอย่างอาร์เรย์คู่ขนาน (Parallel Array)

- อาร์เรย์คู่ขนานไม่จำเป็นต้องเป็นอาร์เรย์ที่มีจำนวนมิติเท่ากัน ยกตัวอย่างเช่น ไม่จำเป็นต้องเป็นอาร์เรย์หนึ่งมิติเหมือนกัน ไม่จำเป็นต้องเป็นอาร์เรย์สองมิติเหมือนกัน เป็นต้น
- อาร์เรย์คู่ขนานอาจจะมีจำนวนมากกว่า 2 อาร์เรย์ก็ได้ นั่นคือ อาร์เรย์คู่ขนานเป็นกลุ่มของอาร์เรย์ที่เก็บข้อมูลที่มีความสัมพันธ์กันไว้ในตำแหน่งเดียวกันของอาร์เรย์เหล่านั้นก็ได้