บทที่ 7

อาร์เรย์ (Array)

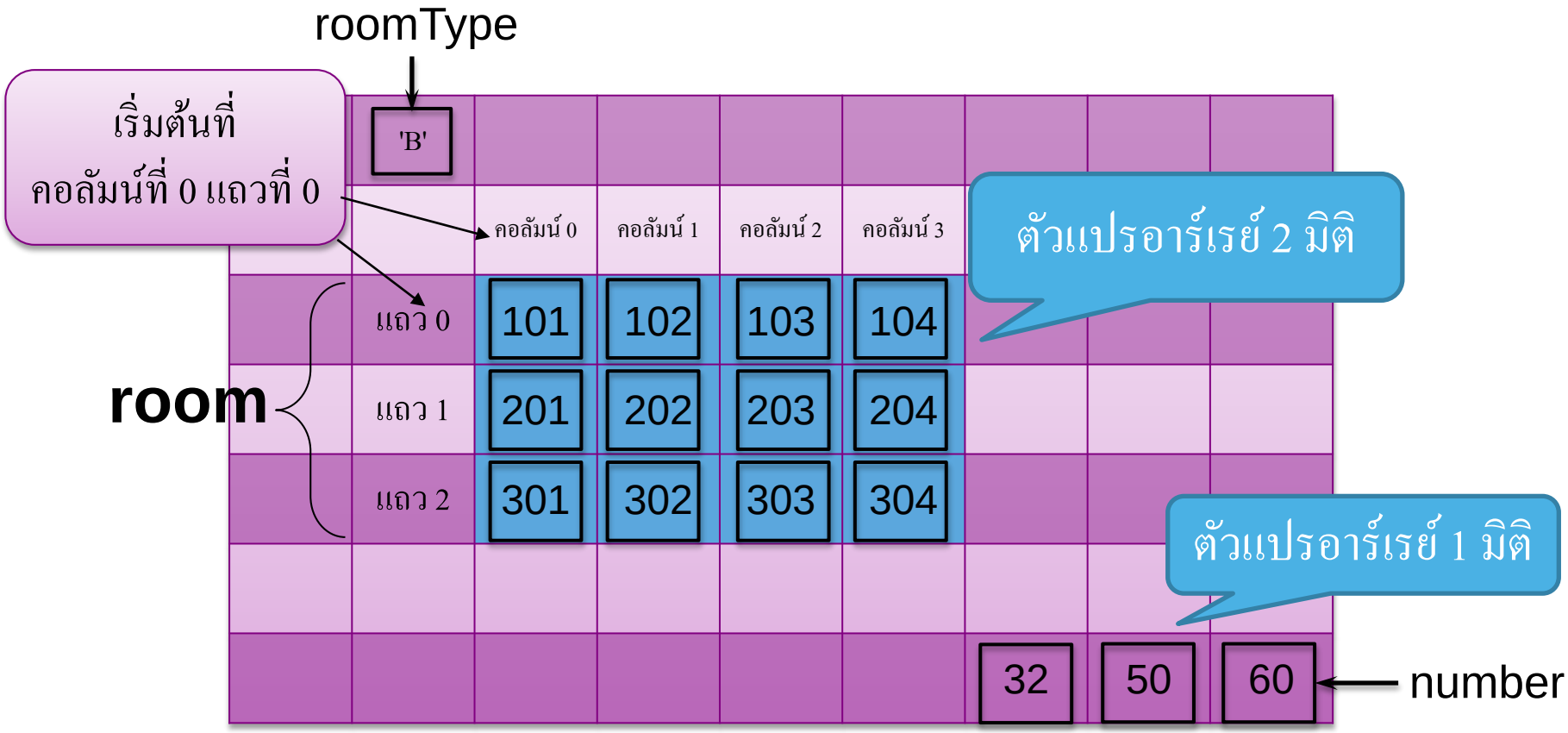
ตอนที่ 2

อาร์เรย์แบบ 1 มิติ

temp[7] = {23, 29, 29, 32, 31, 31, 32}

อาร์เรย์แบบ 2 มิติ

อาร์เรย์ 2 มิติบนหน่วยความจำ



จำลองตัวแปรชนิดต่างๆ บนหน่วยความจำ

การประกาศตัวแปรอาร์เรย์ 2 มิติ

ชนิดข้อมูล ชื่อตัวแปรอาร์เรย์[จำนวนแถว][จำนวนคอลัมน์];

// ประกาศตัวแปรอาร์เรย์ชื่อ room ใช้เก็บตัวเลข 12 ตัว โดยเก็บในแนว 3 แถว 2 คอลัมน์

int room[3][4];

// ประกาศตัวแปรอาร์เรย์ชื่อ scoreใช้เก็บตัวเลข 150 ตัว โดยเก็บในแนว 15 แถว 10 คอลัมน์

float score[15][10];

การกำหนดค่าเริ่มต้นให้กับอาร์เรย์ 2 มิติ

```
int room[3][4] = {  {101, 102, 103, 104},   <-- แถวที่ 1
                    {201, 202, 203, 204},   <-- แถวที่ 2
                    {301, 302, 303, 304} };  <-- แถวที่ 3
```

	คอลัมน์ 0	คอลัมน์ 1	คอลัมน์ 2	คอลัมน์ 3
แถว 0	101	102	103	104
แถว 1	201	202	203	204
แถว 2	301	302	303	304

การอ้างอิงข้อมูลในอาร์เรย์ 2 มิติ

ใช้ชื่อตัวแปรตามด้วยเครื่องหมาย [ ][ ] ซึ่งตัวแรกจะระบุหมายเลขแถว ส่วนตัวที่สองระบุหมายเลขคอลัมน์

ชื่ออาร์เรย์ [index ของแถว] [index ของคอลัมน์]

	[0]	[1]	[2]	[3]
room[0]	101	102	103	104
room[1]	201	202	203	204
room[2]	301	302	303	304

room[1][2]

room[2][3]

จำลองตัวแปรอาร์เรย์บนหน่วยความจำ

การกำหนดค่าและเข้าถึงข้อมูล

	[0]	[1]	[2]	[3]
room[0]	101	102	103	104
room[1]	201	202	203	204
room[2]	301	302	303	304

```
room[2][1] = 312;              // กำหนดค่าใหม่ให้สมาชิกที่อยู่แถวที่ 3 คอลัมน์ที่ 2
```

```
printf("%d", room[2][3]);      // จะได้ผลลัพธ์เป็น 304
```

```
scanf("%d", &room[0][1]);      // รับค่าจากแป้นพิมพ์ไปเก็บทับในแถวที่ 1 คอลัมน์ที่ 2
```

```
int sum = room[1][2] + room[2][0];      // sum จะได้ค่าเป็น 203+301 = 504
```

```
int i=0, j=3;
printf("%d", room[i][j]);              // จะได้ผลลัพธ์เป็น 104
```

การวนลูปเพื่อแสดงค่าในอาร์เรย์ 2 มิติ

```
1  void main() {
2      int room[3][4] = {
3          {101,102, 103, 104},
4          { 201, 202, 203, 204},
5          { 301, 302, 303, 304}  };
6      int i; // กำหนดให้ตัวแปร i ใช้ควบคุมตำแหน่งแถว (ลูปนอก)
7      int j; // กำหนดให้ตัวแปร j ใช้ควบคุมตำแหน่งคอลัมน์ (ลูปใน)
8      for (i=0; i<3; i++) { // ลูปนอก ใช้เลื่อนตำแหน่งของแถว
9        for (j=0; j<4; j++) { // ลูปใน ใช้เลื่อนตำแหน่งของคอลัมน์
10           printf("%d ", room[i][j]); // แสดงค่าจากอาร์เรย์ room แถวที่ i คอลัมน์ที่ j
11       } // จบลูป For ของคอลัมน์
12       printf("\n");
13     } // จบลูป For ของแถว
14 }
```

การวนลูปเพื่อแสดงค่าในอาร์เรย์ 2 มิติ

	j=0	j=1	j=2	j=3	j=4
i = 0 →	101	102	103	104	
i = 1 →	201	202	203	204	
i = 2 →	301	302	303	304	

room

```
for (i=0; i<3; i++) {
    for (j=0; j<4; j++) {
        printf("%d ", room[i][j]);
    }
    printf("\n");
}
```

การวนลูปเพื่อรับค่าในอาร์เรย์ 2 มิติ

```
1. void main() {
2.      int room[3][4];
3.      int i; // กำหนดให้ตัวแปร i ใช้เก็บตำแหน่งแถว
4.      int j; // กำหนดให้ตัวแปร j ใช้เก็บตำแหน่งคอลัมน์
5.      for (i=0; i<3; i++) {
6.          for (j=0; j<4; j++)  {
7.              scanf("%d", &room[i][j]);
8.          } // end inner loop
9.      } // end outer loop
10.}
```

String

- String คือ อาร์เรย์ของอักขระ ที่ถูกปิดท้ายด้วย '\0' อัตโนมัติ

```
char  name[] = "John";
```

'J'	'o'	'h'	'n'	'\0'

- ในทางตรงข้าม อาร์เรย์ของอักขระทั่วไปจะไม่ถูกปิดท้ายด้วย '\0'

```
char name[] = { 'J', 'o', 'h', 'n' };
```

'J'	'o'	'h'	'n'

การกำหนดค่าเริ่มต้นให้กับสตริง

```
char fruit[6] = "Mango";
```

หรือ

```
char fruit[] = "Mango";
```

การอ้างอิงอักขระใน String ทำเช่นเดียวกับอาร์เรย์ 1 มิติ เช่น

```
printf("%c", fruit[2]);   // จะได้ค่า 'n'
```

อาร์เรย์ของสตริง

```
void main() {
    char persons[ ][15] = {
        "Jim",
        "Peter",
        "John"
    };
    printf("%s", persons[1]);     // จะได้ "Peter"
    printf("%c", persons[0][2]); // จะได้ 'm'
}
```

	คอลัมน์ 0	คอลัมน์ 1	คอลัมน์ 2	คอลัมน์ 3	คอลัมน์ 4	คอลัมน์ 5
แถว 0	'J'	'i'	'm'	'\0'		
แถว 1	'P'	'e'	't'	'e'	'r'	'\0'
แถว 2	'J'	'o'	'h'	'n'	'\0'	

คำสั่งเกี่ยวกับ String

- การใช้คำสั่งที่เกี่ยวข้องกับสตริง จะต้องประกาศส่วนหัว

```
#include <string.h>
```

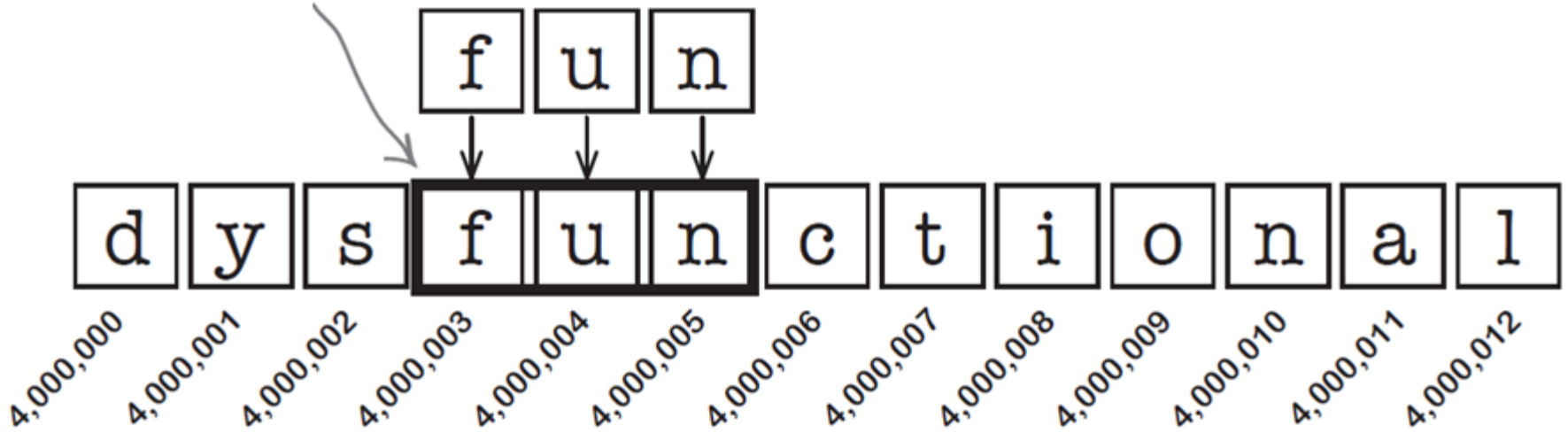
- คำสั่งในไลบรารี string.h
  - strstr(s1, s2) – ค้นหาสตริง s2 ว่ามีในสตริง s1 หรือไม่
  - strlen(s1) – หาความยาวสตริง s1
  - strcpy(s2, s1) – คัดลอกสตริง s1 ไปเก็บไว้ในตัวแปรสตริง s2
  - strcmp(s1, s2) – เปรียบเทียบสตริง s1 และ s2
  - strcat(s1, s2) – นำสตริงใน s2 ต่อกับ s1

การค้นหาสตริง

```
strstr("dysfunctional", "fun")
```

ค้นหาสตริง "fun"

เจอที่ตำแหน่ง 4,000,003



การค้นหาสตริง

```
1. #include <stdio.h>
2. #include <string.h>
3. void main() {
4.       int position = strstr("dysfunctional", "fun");
5.       if (position > 0) {
6.         printf("Found !!!\n");
7.         printf("Position in memory is %d", position);
8.       }
9. }
```

Output ของโปรแกรม

```
Found !!!

Position in memory is 4206631
```

การหาความยาวสตริง

```
1. #include <stdio.h>
2. #include <string.h>
3. void main() {
4.     char s1[30] = "Structured Programming";
5.     int n;
6.     n = strlen(s1);
7.     printf("%d", n);
8. }
```

Output ของโปรแกรม

```
22
```

ตัวอย่าง

การวนลูปเพื่อพิมพ์ String กลับหลัง

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  void main() {
4.      int i;
5.      char word[100];
6.      printf("Enter word : ");
7.      scanf("%[^\n]", word);
8.      printf("Reverse of your word is ");
9.      for(i=strlen(word)-1; i>=0; i--) {
10.             printf("%c", word[i]);
11.     }
12.}
```

Output ของโปรแกรม

```
Enter word : I love you

Reverse of your word is: uoy evol I
```

การคัดลอกข้อมูลสตริง

```
1. #include <stdio.h>
2. #include <string.h>
3. void main() {
4.       char s1[30] = "Structure Programming";
5.       char s2[30];
6.       strcpy(s2, s1);
7.       printf("%s", s2);
8. }
```

s2 = s1;

strcpy(s2, s1);

Output ของโปรแกรม

```
Structure Programming
```

การเปรียบเทียบสตริง

❌ if (s1==s2)

✅ if(strcmp(s1, s2)==0)

```
1. #include <stdio.h>
2. #include <string.h>
3. void main() {
4.      char s1[30], s2[30];
5.      scanf("%s", s1);
6.      scanf("%s", s2);
7.      if (strcmp(s1, s2)==0)
8.          printf("equal");
9.      else
10.         printf("not equal");
11.}
```

str1 เหมือนกับ str2	ผลที่ได้คือ 0
str1 < str2	ผลที่ได้เป็นค่าลบ (น้อยกว่า 0)
str1 > str2	ผลที่ได้เป็นค่าบวก (มากกว่า 0)

การต่อสตริง

❌ s1 = s1 + s2;

✅ strcat(s1, s2);

```
1. #include <stdio.h>
2. #include <string.h>
3. void main() {
4.      char s1[30] = "Structure ";
5.      char s2[30] = "Programming";
6.      strcat(s1, s2);
7.      printf("%s", s1);
8. }
```

ตัวอย่าง

นักเรียนห้องหนึ่งมี 4 คน แต่ละคนมีวิชาเรียน 3 วิชา จงเขียนโปรแกรมเก็บคะแนนทั้ง 3 วิชาของนักเรียนแต่ละคนลงในตัวแปรอาร์เรย์ 2 มิติ หลังจากนั้นแสดงผลรวมสรุปคะแนนของแต่ละคนเมื่อกรอกคะแนนครบทุกคนแล้ว

แบ่งการทำงานออกเป็น 2 ส่วน

- รับค่าเก็บลงอาร์เรย์
- หาผลรวมแต่ละแถว และแสดงผล

ตัวอย่างหน้าจอโปรแกรม

C:\ppp\bin\Debug\ppp.exe

```
Student 1
Score 1: 10
Score 2: 10
Score 3: 10

Student 2
Score 1: 30
Score 2: 30
Score 3: 30

Student 3
Score 1: 10
Score 2: 30
Score 3: 10

Student 4
Score 1: 20
Score 2: 61
Score 3: 11

Summary
Student 1: 10 10 10  = 30
Student 2: 30 30 30  = 90
Student 3: 10 30 10  = 50
Student 4: 20 61 11  = 92
```

รับค่าเก็บลงอาร์เรย์

หาผลรวมแต่ละแถว และแสดงผล

รับค่าเก็บลงอาร์เรย์

	[0]	[1]	[2]
[0]	[0][0]	[0][1]	[0][2]
[1]	[1][0]	[1][1]	[1][2]
[2]	[2][0]	[2][1]	[2][2]
[3]	[3][0]	[3][1]	[3][2]

1. สร้างอาร์เรย์ขนาด 4 x 3 เพื่อเก็บจำนวนเต็ม

```
int score[4][3];
```

2. สร้างลูปซ้อนลูป ลูปนอกใช้ i ควบคุมแถว ลูปในใช้ j ควบคุมคอลัมน์

จำนวนแถว

จำนวนคอลัมน์

```
1. for (i=0; i<4; i++) {
2.       for(j=0; j<3; j++) {
3.         scanf("%d", &score[i][j]);
4.       }
5. }
```

หาผลรวมแต่ละแถว และแสดงผล

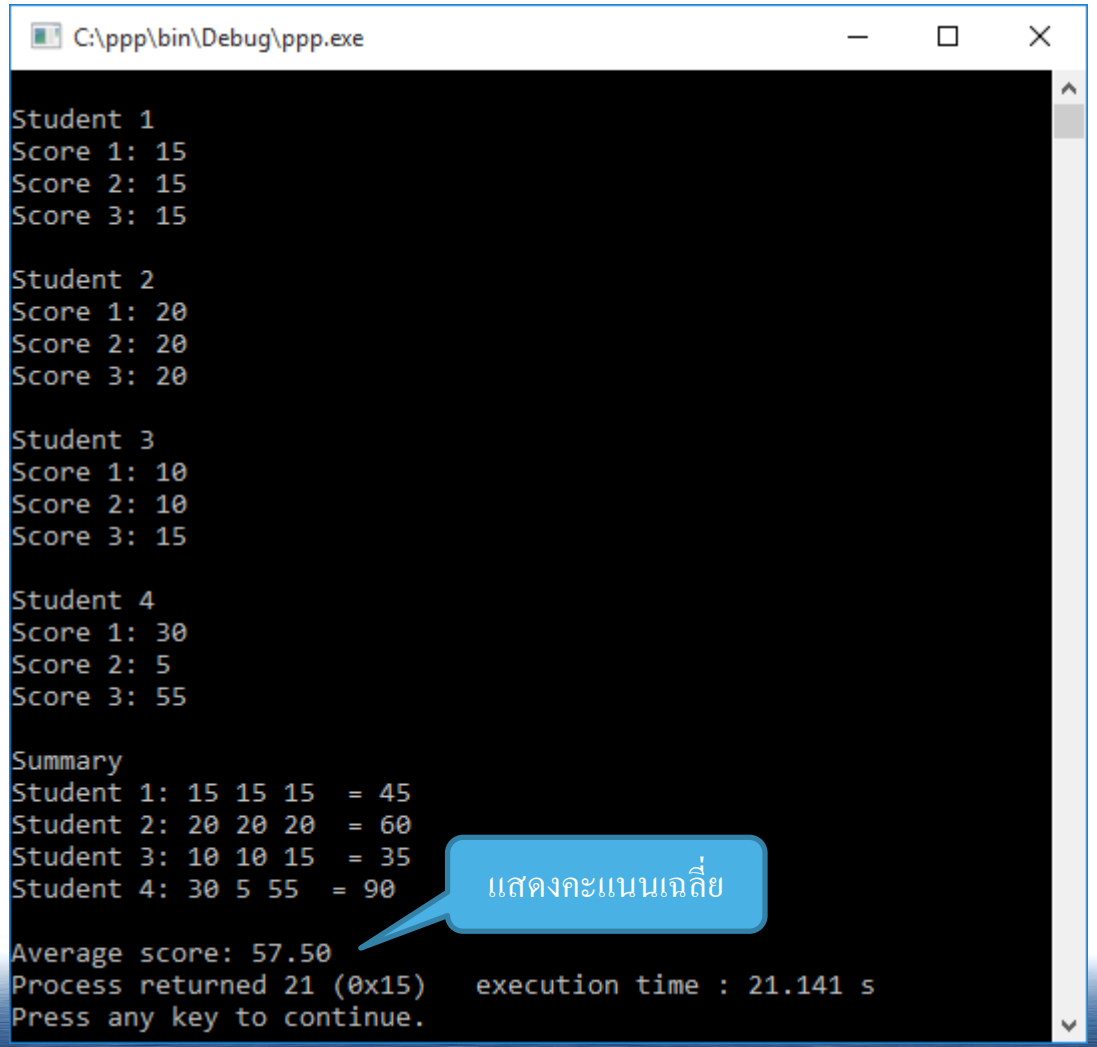
	[0]	[1]	[2]
[0]	[0][0] 20	[0][1] 10	[0][2] 10
[1]	[1][0] 5	[1][1] 5	[1][2] 3
[2]	[2][0] 2	[2][1] 2	[2][2] 1
[3]	[3][0] 7	[3][1] 8	[3][2] 8

วนลูปแสดงค่าในอาร์เรย์

```
int sum;
for (i=0; i<4; i++) {
    sum = 0;
    printf("Student %d: ", i+1);
    for(j=0; j<3; j++) {
        printf("%d ", score[i][j]);
        sum += score[i][j];
    }
    printf(" = %d\n", sum);
}
```

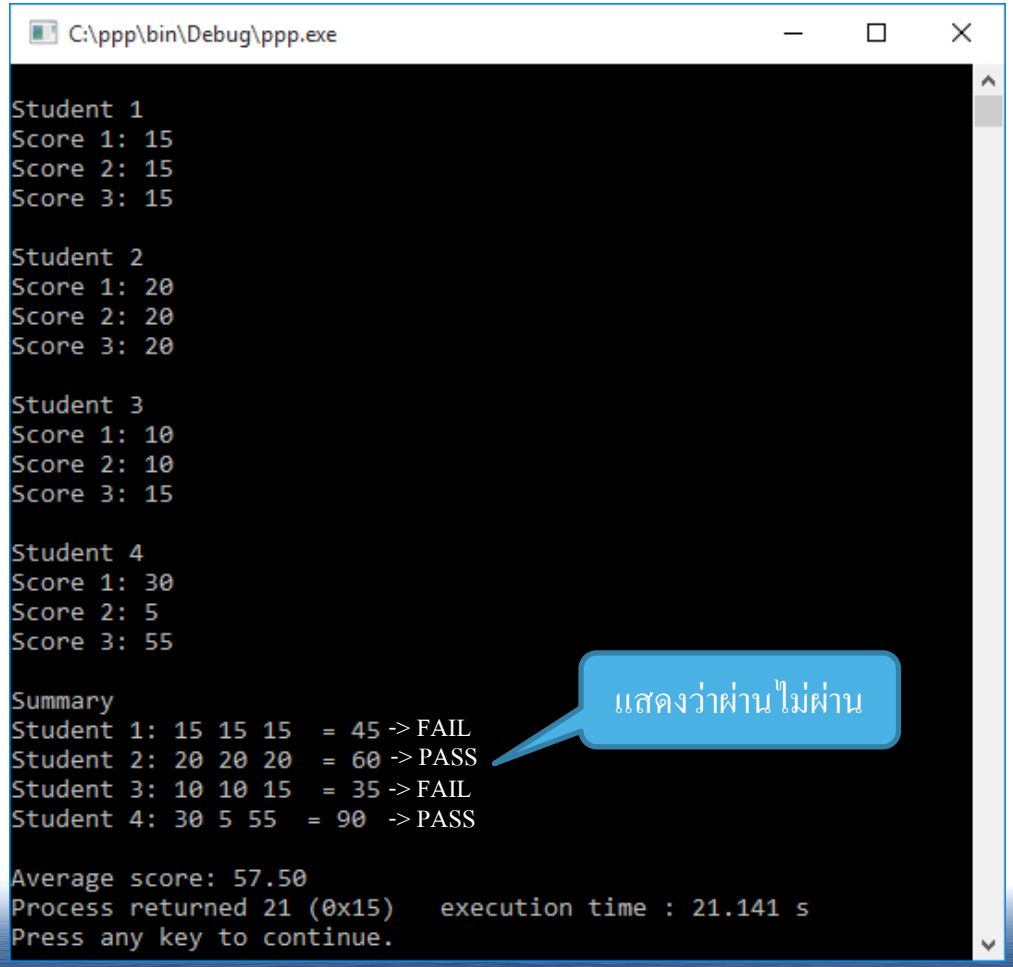
กิจกรรม

- จากตัวอย่างก่อนหน้านี้จงแสดงค่าเฉลี่ยของคะแนนรวมทั้งหมด



กิจกรรม

- จากตัวอย่างก่อนหน้านี้จงแสดงคำว่า FAIL เมื่อคะแนนรวมของนักเรียนคนนั้นไม่ถึง 50 คะแนน และแสดงคำว่า Pass เมื่อเกิน 50 คะแนน



กิจกรรม

- จงรับค่า String จากผู้ใช้ หลังจากนั้นให้แสดงอักขระที่อยู่ใน String โดยแบ่งการแสดงผลออกเป็นบรรทัดละ 5 อักขระ

ตัวอย่างหน้าจอโปรแกรม

```
Enter sentence: aaaaa+++++yyyyy ↵

aaaaa
+++++
yyyyy
```

กิจกรรม

- จงเขียนโปรแกรมรับค่าประโยค 1 ประโยค หลังจากนั้นนำแต่ละคำในประโยคแยกแสดง 1 คำ ต่อ 1 บรรทัด แล้วแสดงจำนวนคำทั้งหมด เช่น

ตัวอย่างหน้าจอโปรแกรม

```
Enter the sentence: This is a cat ↵

This
is
a
cat

The sentence has 4 words.
```

Assignment#7

ข้อ 1. จงเขียนโปรแกรมเก็บค่าเมตริกซ์จำนวนเต็ม ซึ่งมีขนาดแถวและหลักตามที่ผู้ใช้กำหนด โดยใช้อาร์เรย์ 2 มิติ หลังจากนั้นให้ทำการรับค่าสมาชิกให้ครบทุกช่อง และสร้างอาร์เรย์ใหม่เพื่อเก็บ Transpose Matrix และแสดงผลลัพธ์เมตริกซ์ปกติ และเมตริกซ์ที่ Transpose แล้ว

แบ่งการทำงานออกเป็น 3 ส่วน

- รับค่าจำนวนแถว/คอลัมน์, สร้างอาร์เรย์, รับค่าเก็บลงอาร์เรย์
- สร้างอาร์เรย์ Transpose Matrix

รับค่าจำนวนแถว/คอลัมน์, สร้างอาร์เรย์, รับค่า

อาร์เรย์ a

a[0][0]	a[0][1]
a[1][0]	a[1][1]
a[2][0]	a[2][1]
a[3][0]	a[3][1]

1. รับค่าจำนวนแถว และจำนวนคอลัมน์

```
scanf("%d", &row);          // รับค่า 4
scanf("%d", &column);       // รับค่า 2
```

2. สร้างอาร์เรย์ 2 มิติ

```
int a[row][column];
```

3. สร้างลูปซ้อนลูป ลูปนอกใช้ i ควบคุมแถว ลูปในใช้ j ควบคุมคอลัมน์

```
for (i=0; i<row; i++) {
    for(j=0; j<column; j++) {
        scanf("%d", &a[i][j]);
    }
}
```

row = 4

column = 5

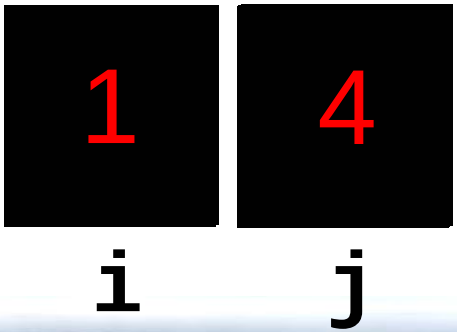
Transpose Matrix

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \qquad A^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

สร้างอาร์เรย์ Transpose Matrix

อาร์เรย์ a

a[0][0] 5	a[0][1] 2
a[1][0] 1	a[1][1] 3
a[2][0] 7	a[2][1] 8
a[3][0] 4	a[3][1] 6



สร้างอาร์เรย์ใหม่โดยใช้ จำนวนแถว สลับกับจำนวนคอลัมน์

int aT[column][row];

row = 4

column = 5

อาร์เรย์ aT

aT[0][0] 5	aT[0][1] 1	aT[0][2] 7	aT[0][3] 4
a[1][0] 2	a[1][1] 3	aT[1][2] 8	aT[1][3] 6

ลูปนอกจะวนตามจำนวนคอลัมน์
(ปกติลูปนอกจะวนตามแถว)

```
for(i=0; i<column; i++) {
    for(j=0; j<row; j++) {
        aT[ i ][ j ] = a[ j ][ i ];
    }
}
```

ใช้ j เป็นตัวระบุคอลัมน์

Assignment#7

ข้อ 2. จงเขียนโปรแกรมหาผลบวกและลบเมตริกซ์ A และ B ซึ่งเมตริกซ์เก็บจำนวนเต็มในตัวแปรอาร์เรย์ 2 มิติขนาด 2 x 3 โดยเริ่มต้นโปรแกรมจะรับข้อมูลเพื่อเก็บลงเมตริกซ์ A และ B จนครบทุกช่อง หลังจากนั้นจะแสดงเมตริกซ์ A, B, A+B, และ A-B

```
Enter element of matrix A
 A[0][0]: 5 ↵
 A[0][1]: 1 ↵
 A[0][2]: 2 ↵
 A[1][0]: 3 ↵
 A[1][1]: 4 ↵
 A[1][2]: 2 ↵
Enter element of matrix B
 B[0][0]: 1 ↵
 B[0][1]: 2 ↵
 B[0][2]: 3 ↵
 B[1][0]: 4 ↵
 B[1][1]: 5 ↵
 B[1][2]: 6 ↵
```

ตัวอย่างการทำงาน

```
== Matrix A ==
5 1 2
3 4 2

== Matrix B ==
1 2 3
4 5 6

== Matrix A + B ==
6 3 5
7 9 8

== Matrix A - B ==
4 -1 -1
-1 -1 -4
```

ตัวอย่างการทำงาน

Assignment#7

ข้อ 3. จงเขียนโปรแกรมหาผลรวมของตัวเลขในแนวทแยงของเมตริกซ์จัตุรัส ขนาด NxN ทั้งในแนวทแยงซ้ายและขวา โดยเริ่มต้นให้รับข้อมูลเพื่อเก็บลงเมตริกซ์ หลังจากนั้นให้แสดงเมตริกซ์ที่รับเข้ามา และแสดงผลรวม

```
Enter N to create matrix NxN : 2
Enter element of matrix A
 A[0][0]: -3 ↵
 A[0][1]: 2 ↵
 A[1][0]: 5 ↵
 A[1][1]: 6 ↵

== Matrix NxN ==
-3 2
5 6

sum \ = 3
sum / = 7
```

ตัวอย่างการทำงาน

```
Enter N to create matrix NxN : 3
Enter element of matrix A
 A[0][0]: 1 ↵
 A[0][1]: 5 ↵
 A[0][2]: 0 ↵
 A[1][0]: 3 ↵
 A[1][1]: 4 ↵
 A[1][2]: 6 ↵
 A[2][0]: 2 ↵
 A[2][1]: 3 ↵
 A[2][2]: 2 ↵

== Matrix NxN ==
1 5 0
3 4 6
2 3 2

sum \ = 7
sum / = 6
```

ตัวอย่างการทำงาน

Assignment#7

ข้อ 4. จงเขียนโปรแกรมเพื่อค้นหา Keyword จากข้อความที่รับเข้ามา โดยแสดงตำแหน่งเริ่มต้นของ Keyword ที่ค้นหาเจอ

Test case:

Input	Output
Input text : I am programmer Input keyword : programmer	Found at position : 6
Input text : c programming Input keyword : programming	Found at position : 3
Input text : programming is great fun Input keyword : fun	Found at position : 22

Assignment#7

ข้อ 5. จงเขียนโปรแกรมเพื่อรับชื่อและน้ำหนักของนักศึกษาจำนวน 5 คน หลังจากนั้นจะสรุปชื่อนักศึกษาที่มีน้ำหนักสูงที่สุด น้อยที่สุด และค่าน้ำหนักเฉลี่ย

```
Enter student name 1: John ↵
Enter weight: 45 ↵
Enter student name 2: Bob ↵
Enter weight: 72 ↵
Enter student name 3: Peter ↵
Enter weight: 60 ↵
Enter student name 4: Robert ↵
Enter weight: 63 ↵
Enter student name 5: Smith ↵
Enter weight: 42 ↵

Minimum: Smith 42 Kg.
Maximum: Bob 72 Kg.
Average Weight: 56.40 Kg.
```

ตัวอย่างการทำงาน

*หมายเหตุ

การกำหนดค่าให้กับ String ไม่สามารถใช้ max_name = "John"; ได้ ต้องใช้ strcpy(max_name, "John");