

## Lab 4 อาร์เรย์

### วัตถุประสงค์

1. เข้าใจการทำงานกับอาร์เรย์ ทั้งการประกาศขนาดแบบสถิต (static) และแบบพลวัต (dynamic)
2. เข้าใจการทำงานร่วมกันระหว่างอาร์เรย์และ Pointers อย่างง่าย
3. สามารถเขียนโค้ดทำงานกับอาร์เรย์อย่างง่ายได้

### การส่งงาน

ให้พิมพ์คำตอบลงในฟอร์มที่กำหนดให้ บนเว็บไซต์รายวิชา

กำหนดส่ง: ภายในวันอังคารที่ 10 กันยายน 2567 เวลา 10.00 น.

การส่งงานล่าช้าจะทำให้ไม่ได้รับคะแนนในส่วนนี้

**อาร์เรย์ (array)** เป็นการจองพื้นที่ต่อเนื่องในหน่วยความจำ (Memory) ของคอมพิวเตอร์เพื่อใช้เก็บชุดของข้อมูลประเภทเดียวกัน เช่น อาร์เรย์ขนาด 10 ของเลขจำนวนเต็ม เก็บชุดข้อมูลตัวเลขได้ 10 ตัว เป็นต้น การอ้างถึงสมาชิกแต่ละตัวในอาร์เรย์ใช้เครื่องหมาย [ ] โดยระบุตำแหน่งสมาชิกที่ต้องการ ซึ่งตำแหน่งที่เป็นไปได้จะมีค่าเริ่มต้นที่ 0 จนถึงขนาดสูงสุดของมัน - 1 เช่นอาร์เรย์ขนาด 10 ตำแหน่งที่เป็นไปได้จะมีค่า 0 ถึง 9 นอกจากนี้เมื่อประกาศอาร์เรย์สามารถกำหนดค่าเริ่มต้นให้กับอาร์เรย์ได้ ในกรณีนี้อาจกำหนดขนาดหรือไม่กำหนดขนาดของอาร์เรย์ก็ได้

หมายเหตุ อาร์เรย์ของ C++ ใช้เครื่องหมาย [ ] ท้ายตัวแปร ไม่สามารถประกาศไว้หน้าตัวแปรแบบจาวา

**ตัวชี้ (pointer)** สามารถใช้อ้างถึงที่ตั้งของสมาชิกแต่ละตัวในอาร์เรย์ได้ การขยับค่าด้วยการบวกหรือลบตำแหน่งตัวชี้ จะทำให้เกิดการเคลื่อนจากตำแหน่งไปเท่ากับตัวเลขที่เพิ่มหรือลบ เช่น

```
int a[] = {1,2,3,4};           // a with size 4 with initial values 1 to 4
int *pA = a;                   // declare pointer and assign it to point to a
cout << *pA << endl;           // print 1
pA++;                           // move pointer to the next element
cout << *pA << endl;           // print 2
```

เพื่อให้เข้าใจแนวคิดเกี่ยวกับอาร์เรย์ให้นักศึกษารัน Code ที่กำหนดให้และตอบคำถามในแต่ละข้อต่อไปในแบบฟอร์มที่กำหนดให้

### ข้อที่ 1

```
#include <iostream>
using namespace std;

void print(int data[], int size) {
    cout << "Print array" << endl;
    for (int i=0; i<size; i++) {
        cout << data[i] << ", ";
    }
    cout << endl;
}

int main(int argc, char** argv) {
    int arr1[] = {2, 3, 5, 7, 11};
    int arr2[10];

    print(arr1, 5);
    print(arr2, 10);

    std::copy(std::begin(arr1)+1, std::end(arr1), arr2);
    print(arr2, 10);

    return 0;
}
```

- 1.1 หลังจากรันโปรแกรมบนหน้า Console ของนักศึกษาจะมีข้อความใดปรากฏขึ้นมา (หากเกิด Error ให้รายงานว่าเกิด Error messages ว่าอย่างไรบ้าง)
- 1.2 ให้นักศึกษาลองอธิบายว่าทำไมผลการรันจึงได้ดังนั้น
- 1.3 เมื่อกัด copy, begin, end ของ std ทำหน้าที่อะไร

## ข้อ 2

1.	#include <iostream>
2.	#include <string>
3.	
4.	using namespace std;
5.	void print(string data[], int size) {
6.	cout << "Print array" << endl;
7.	for (int i=0; i<size; i++) {
8.	cout << data[i] << ", ";
9.	}
10.	cout << endl;
11.	}
12.	
13.	int main(int argc, char** argv) {
14.	string arr1[] = {"Apple", "Banana", "Coconut", "Durian"};
15.	string *arr2 = new string[6];
16.	
17.	// 1) เติมโค้ดที่ 2 บรรทัดนี้เพื่อให้เมื่อสั่งให้ทำงานพิมพ์ผลลัพธ์ดังตัวอย่างที่ให้
18.	
19.	
20.	std::copy(std::begin(arr1), std::end(arr1), arr2);
21.	print(arr2, 6);
22.	// 2) เติมโค้ดเพื่อคืนพื้นที่ของ arr2 ให้กับระบบปฏิบัติการ
23.	return 0;
24.	}

เมื่อให้ทำงาน แสดงผลลัพธ์ดังนี้ในหน้า console

Print array

Apple, Banana, Coconut, Durian, Eggplant, AppleBanana,

- 2.1 ให้เติมโค้ดสำหรับบรรทัดที่ 17 และ 18 เพื่อให้ค่าของสมาชิกในอาร์เรย์ได้ผลลัพธ์เป็นดังตัวอย่าง
- 2.2 ให้เติมโค้ดตรงบรรทัดที่ 22 เพื่อคืนค่าพื้นที่อ้างอิงถึงโดย arr2 ก่อนยุติโปรแกรม

## ข้อที่ 3

1.	#include <iostream>
2.	using namespace std;
3.	
4.	int main(int argc, char** argv) {
5.	int arr[] = {1, 2, 3, 4, 5};
6.	int *ptr = arr;

7.	
8.	cout << *ptr << endl;
9.	cout << *++ptr << endl;
10.	cout << ++*ptr << endl;
11.	cout << *ptr++ << endl;
12.	cout << *ptr << endl;
13.	
14.	return 0;
15.	}

3.1 หลังจากรันโปรแกรมบนหน้า Console ของนักศึกษาจะมีข้อความใดปรากฏขึ้นมา (หากเกิด Error ให้รายงานว่าเกิด Error messages ว่าอย่างไรบ้าง)

3.2 เมื่อให้ทำงานและก่อนที่โปรแกรมจะยุติ ให้นักศึกษาบอกค่าแต่ละสมาชิกใน arr ว่าเป็นเท่าใด และ ptr ที่อยู่ตำแหน่งใดของอาร์เรย์ arr (ระบุหมายเลขดัชนี) ทำไมผลลัพธ์จึงเป็นเช่นนั้น

#### ข้อ 4

1.	#include <iostream>
2.	using namespace std;
3.	
4.	void print(int data[], int size) {
5.	cout << "Print array" << endl;
6.	for (int i=0; i<size; i++) {
7.	cout << data[i] << ", ";
8.	}
9.	cout << endl;
10.	}
11.	void setZero(int* a, int size) {
12.	for (int i=0; i<size; i++) {
13.	*a = 0;
14.	a++;
15.	}
16.	}
17.	
18.	void setVal(int a[], int size, int val) {
19.	for (int i=0; i<size; i++) {
20.	a[i] = val;
21.	}
22.	}
23.	
24.	int main(int argc, char** argv) {
25.	const int MAX_SIZE = 5;
26.	
27.	int *arr = new int[MAX_SIZE];
28.	// int arr[MAX_SIZE];
29.	
30.	setVal(arr, MAX_SIZE, 12);
31.	print(arr, MAX_SIZE);
32.	
33.	setZero(arr, MAX_SIZE);
34.	print(arr, MAX_SIZE);
35.	
36.	return 0;
37.	}

#### 4.1 ฟังก์ชัน setZero และ setVal ทำหน้าที่อะไร

## 4.2 ถ้าฟังก์ชัน `setZero` ในบรรทัดที่ 11

**void setZero(int\* a, int size)** เปลี่ยนเป็น

```
void setZero(int* &a, int size)
```

ส่งผลกับการทำงานของโปรแกรมอย่างไร

4.3 จากโค้ดที่ให้เริ่มต้น ถ้าเปลี่ยนบรรทัด 27 เป็นโค้ดเป็นบรรทัด 28 (comment 27 และ uncomment 28) แทน การทำงานและการจัดการหน่วยความจำของโปรแกรมนี้อาจต่างกันหรือไม่ อย่างไร

๑๑