Array

Lab 06

ปัญหาจากอาทิตย์ที่แล้ว

Lab06 - Array

ปัญหาจากอาทิตย์ที่แล้ว

HW 05-03 : Reg System

Home > Cou

Description

ท่านได้รับว่าจ้างให้เขียนระบบทะเบียน โดยความต้องการของระบบมีดังนี้

ส่วนการรับข้อมูล

- ระบบจะรับข้อมูลของผู้ใช้ทั้งหมด 50 คน
- ข้อมูลของแต่ละคนที่รับเข้ามามี (จำนวนเต็มทั้งหมด)
  - อายุ
  - ส่วนสูง
  - น้ำหนัก

ส่วนการแสดงผล

- สรุปผลของข้อมูลโดยมีเงื่อนไขดังนี้
  1. จำนวนผู้ที่อายุตั้งแต่ 20 ปี ขึ้นไปและมีความสูงตั้งแต่ 160 ขึ้นไป
  2. จำนวนผู้ที่อายุต่ำกว่า 20 ปี และมีความสูงไม่เกิน 180 หรือ น้ำหนักไม่ต่ำกว่า 60
  3. จำนวนผู้ที่อายุตั้งแต่ 30 ปี ขึ้นไปและมีน้ำหนักในช่วง 40 - 80
  4. จำนวนผู้ที่อายุต่ำกว่า 40 ปี และมีน้ำหนักต่ำกว่า 85 หรือ ความสูงไม่เกิน 200
  5. ค่าเฉลี่ยอายุทั้งหมด (จำนวนเต็ม)
  6. ค่าเฉลี่ยความสูงทั้งหมด (ทศนิยม 2 ตำแหน่ง)
  7. ค่าเฉลี่ยน้ำหนักทั้งหมด (ทศนิยม 2 ตำแหน่ง)

***ทศนิยมให้ใช้ float**
***ห้ามใช้ Array**

by นายธนกฤต ทรัพย์ประสิทธิ์
25 July 2024, 13:24

Sample Output

```
Age >= 20 and Height >= 160: 36
Age < 20 and Height <= 180 or Weight >= 60: 6
Age >= 30 and Weight >= 40 and Weight <= 80: 16
Age < 40 and Weight < 85 or Height <= 200: 13
Average Age: 57
Average Height: 190.56
Average Weight: 87.88
```

ปัญหาจากอาทิตย์ที่แล้ว

ประกาศให้ตัวแปร = 0

```c
#include <stdio.h>
int main() {
    int age, height, weight
    int cond1=0, cond2=0, cond3=0, cond4=0;
    int num_of_person = 50, sum_age = 0;
    float sum_height = 0.0, sum_weight = 0.0;
    for (int i = 0; i < num_of_person; i++) {
      scanf("%d %d %d", &age, &height, &weight);
      ...
```

Code

```
Age >= 20 and Height >= 160: 36
Age < 20 and Height <= 180 or Weight >= 60: 6
Age >= 30 and Weight >= 40 and Weight <= 80: 16
Age < 40 and Weight < 85 or Height <= 200: 13
Average Age: 57
Average Height: 190.56
Average Weight: 87.88
```

Output

ไม่ประกาศให้ตัวแปร = 0

```c
#include <stdio.h>
int main() {
    int age, height, weight
    int cond1, cond2, cond3, cond4;
    int num_of_person = 50, sum_age;
    float sum_height, sum_weight;
    for (int i = 0; i < num_of_person; i++) {
      scanf("%d %d %d", &age, &height, &weight);
      ...
```

Code

```
Age >= 20 and Height >= 160: 1147568165
Age < 20 and Height <= 180 or Weight >= 60: -1913402690
Age >= 30 and Weight >= 40 and Weight <= 80: 17
Age < 40 and Weight < 85 or Height <= 200: 1804759613
Average Age: 36095253
Average Height: 190.56
Average Weight: 87.88
```

Output

Run on Unix like system

PhysicalCom — -zsh — 55×8

```
Age >= 20 and Height >= 160: -550698971
Age < 20 and Height <= 180 or Weight >= 60: -1913402690
Age >= 30 and Weight >= 40 and Weight <= 80: 17
Age < 40 and Weight < 85 or Height <= 200: 1876686125
Average Age: 37533783
Average Height: 190.56
Average Weight: 87.88
(base) tae@Tanakrits-MacBook-Air PhysicalCom %
```

PhysicalCom — -zsh — 55×8

```
Age >= 20 and Height >= 160: -1488486363
Age < 20 and Height <= 180 or Weight >= 60: -1913402690
Age >= 30 and Weight >= 40 and Weight <= 80: 17
Age < 40 and Weight < 85 or Height <= 200: 1861104941
Average Age: 37222160
Average Height: 190.56
Average Weight: 87.88
(base) tae@Tanakrits-MacBook-Air PhysicalCom %
```

PhysicalCom — -zsh — 55×8

```
Age >= 20 and Height >= 160: -686817243
Age < 20 and Height <= 180 or Weight >= 60: -1913402690
Age >= 30 and Weight >= 40 and Weight <= 80: 17
Age < 40 and Weight < 85 or Height <= 200: 1873802541
Average Age: 37476112
Average Height: 190.56
Average Weight: 87.88
(base) tae@Tanakrits-MacBook-Air PhysicalCom %
```

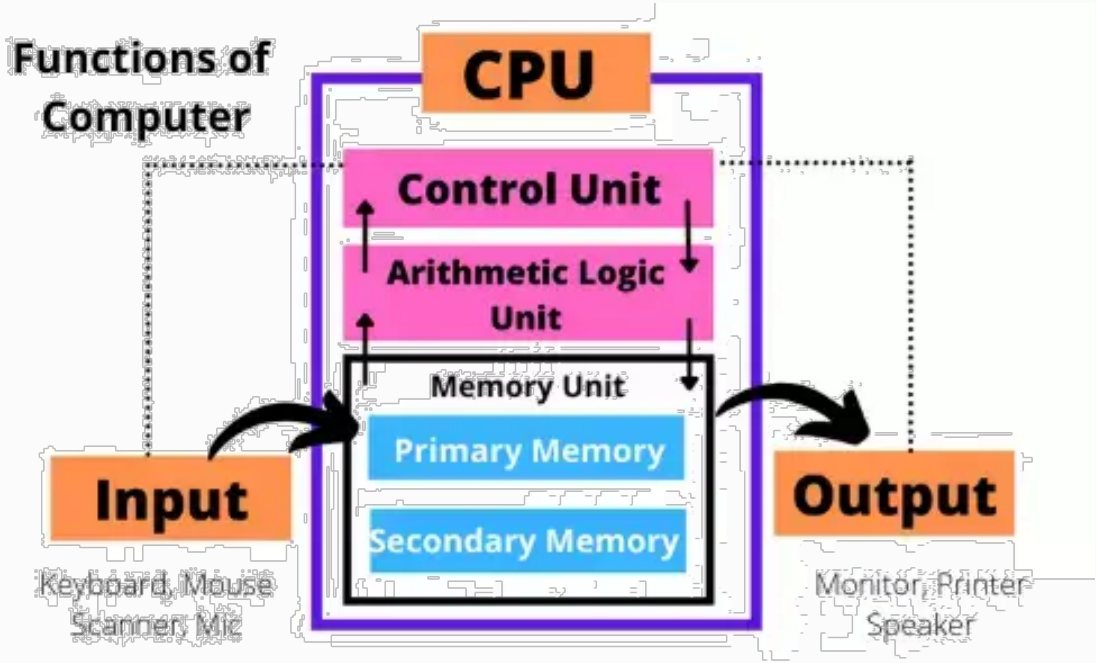
PhysicalCom — -zsh — 55×8

```
Age >= 20 and Height >= 160: 554762277
Age < 20 and Height <= 180 or Weight >= 60: -1913402690
Age >= 30 and Weight >= 40 and Weight <= 80: 17
Age < 40 and Weight < 85 or Height <= 200: 1862808877
Average Age: 37256238
Average Height: 190.56
Average Weight: 87.88
(base) tae@Tanakrits-MacBook-Air PhysicalCom %
```

Computer Functions

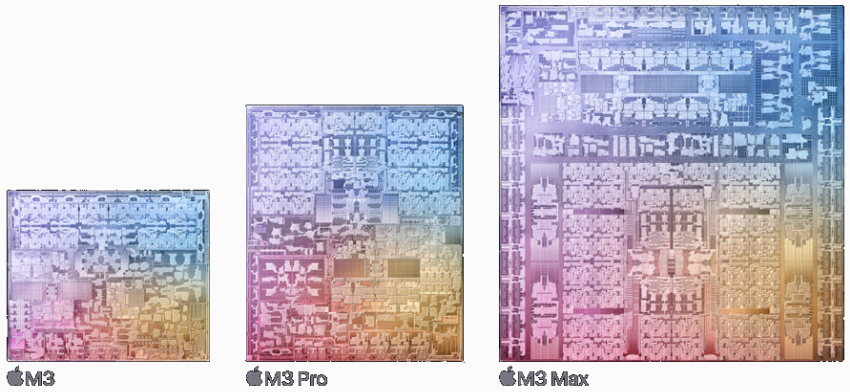
- Data คือ ตัวเลขหรือชุดคำสั่งของ Instruction (ชุดตัวเลข)
- **คอมพิวเตอร์จะทำงานกับตัวเลข** ในคอมพิวเตอร์นับว่าตัวเลขนั้นจะเป็นข้อมูล เมื่อมีหลายๆ ตัวรวมกันจะกลายเป็นรหัสคำสั่งเพื่อให้คอมพิวเตอร์ทำงาน คอมพิวเตอร์มีฟังก์ชันการทำงานได้แก่

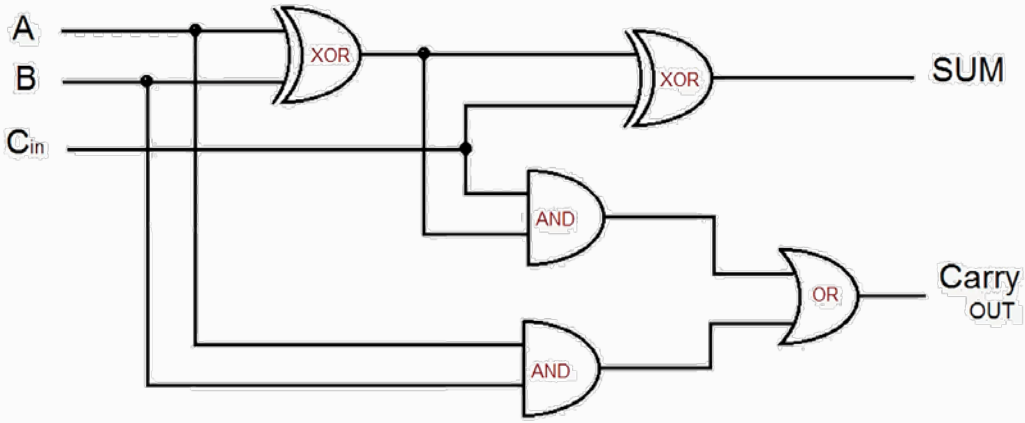
    1. **Data Processing** — คอมพิวเตอร์ทำการประมวลผลตัวเลข
    2. **Data Storage** — จัดเก็บตัวเลขเพื่อเอาไว้ใช้งานต่อไปและดึงค่าที่จัดเก็บเอามาใช้โดยที่ค่าที่เก็บไปต้องเหมือนเดิมเมื่อถูกเรียกใช้
    3. **Data Movement** — การเคลื่อนย้ายข้อมูลจากที่นึงไปยังอีกที่นึง (CPU -> Memory)



1. Processor / CPU (Central Processing Unit)

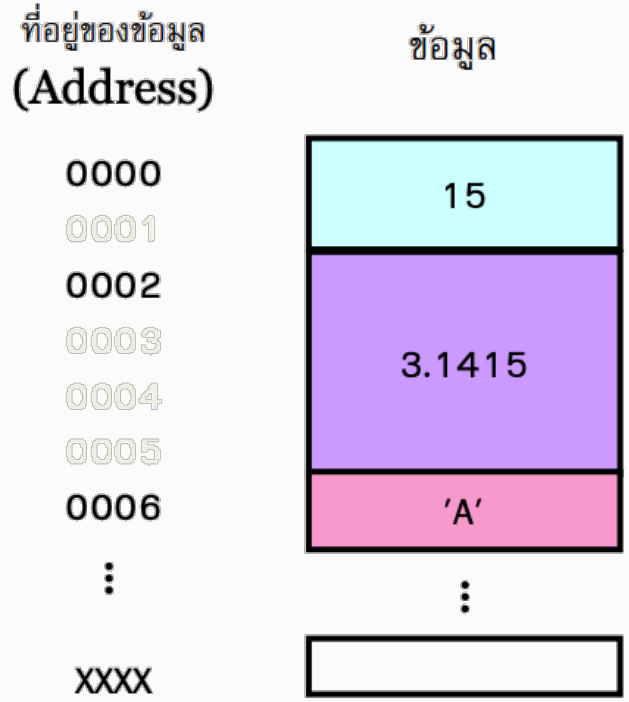
- ควบคุมการทำงานของเครื่องคอมพิวเตอร์
- ประมวลผลตัวเลขหรือข้อมูล (Data)
- เปลี่ยนแปลงค่าตัวเลข
- CPU ดึงคำสั่งที่อยู่ Main Memory มา Execute จะทำทีละคำสั่ง อย่างนี้ไปเรื่อย ๆ ทำงานไม่รู้จบ

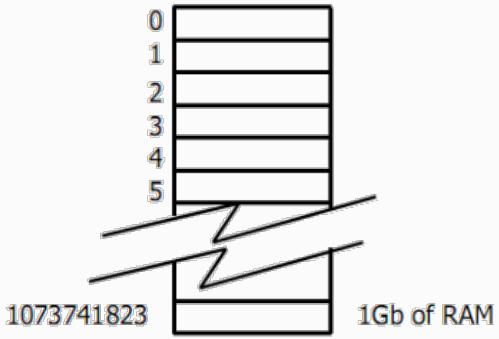




2. Main Memory

- จัดเก็บสิ่งต่างๆ ในการทำงานของคอมพิวเตอร์, RAM ย่อมาจากคำว่า Random Access Memory
- เอาไว้ให้เก็บข้อมูลไว้หลายๆ ส่วน
- ประกอบด้วย **ตำแหน่ง (Address)** กับ **ค่าข้อมูลที่เก็บ (Values)**
- **ใน Memory จะเก็บเป็นช่องๆ จะมีการอ้างตำแหน่งของช่องนั้นๆ เรียกว่า Address** และจะมีการแบ่งส่วนของ Memory





2. Main Memory

F45AAD4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AAE4	1	0	0	1	0	1	0	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1
F45ABF4	0	0	1	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0
F45AC14	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1	0	1	1	1	0	0	0	0	1	1	1	0	1	1	0	0	0
F45AC24	1	0	1	0	0	0	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0
F45AC34	1	1	0	0	1	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	1	0	0	1	1	0	1	1	0	0	1	1
F45AC44	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1
F45AC54	1	1	0	0	1	1	0	1	0	1	1	1	0	0	1	1	1	0	1	0	1	0	1	0	1	0	0	0	1	1	1	0
F45AC64	0	0	0	0	1	0	1	1	0	1	1	1	1	0	0	0	1	1	1	0	1	0	1	1	1	0	0	0	0	1	1	0
F45AC74	1	0	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	0	1	1	0	0	1	0	1	0	0	1
F45AC84	1	0	1	1	0	1	1	1	1	1	1	1	0	0	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	0
F45AC94	0	1	0	1	1	0	1	0	1	1	0	0	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	0	0	1	0

Signed Integer (int)

- มีขนาดของข้อมูล 32 bits (4 bytes) สำหรับคอมส่วนใหญ่ในปัจจุบัน
- เก็บค่าแบบ Two's Complement Number

$= (2^{25} * 1) + (2^{24} * 1) + (2^{23} * 1) + (2^{22} * 1) + (2^{21} * 1) + (2^{15} * 1) + (2^{14} * 1) + (2^{13} * 1) + (2^{10} * 1) + (2^3 * 1) + (2^0 * 1)$

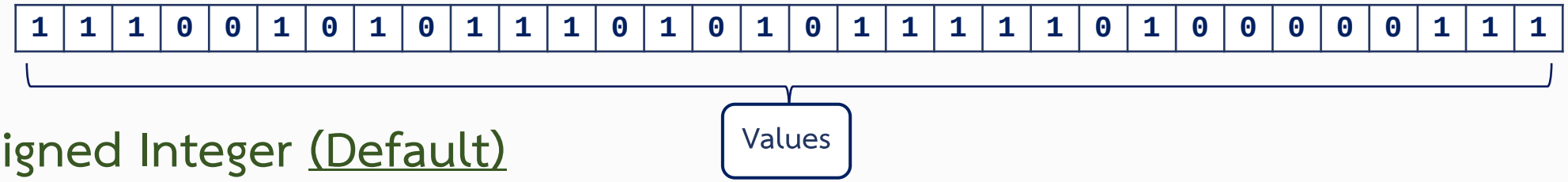
$= 33554432 + 16777216 + 8388608 + 4194304 + 2097152 + 32768 + 16384 + 8192 + 1024 + 8 + 1$

$= 65070089$

Integer in C

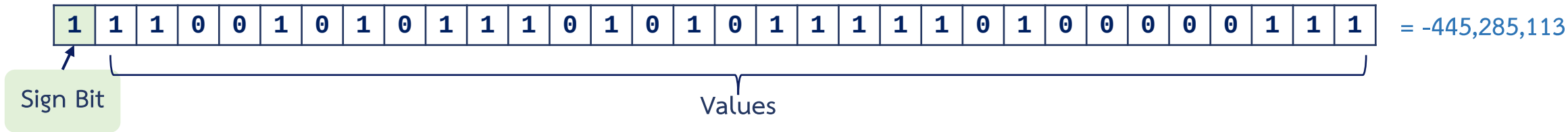
- **Unsigned Integer**
  - จำนวนบิตทั้งหมดที่เก็บจะเป็นค่าตัวเลข
  - สามารถเก็บค่าได้แค่ค่าบวกเท่านั้น
  - can hold values from 0 to $2^{32} - 1$ *(4,294,967,295)*

```
unsigned int num_x=3849682183;
```



- **Signed Integer (Default)**
  - 1 บิตตัวที่ซ้ายที่สุดจะเป็นตัวระบุค่าในการคำนวณ (Sign Bit)
  - เก็บค่าแบบ Two's Complement Number
  - สามารถเก็บค่าทั้งค่าบวกและค่าลบได้
  - can hold values from $-2^{32}/2 - 1$ *(-2,147,483,648)* to $2^{32}/2 - 1$ *(2,147,483,647)*

```
int num_x=3849682183;
```

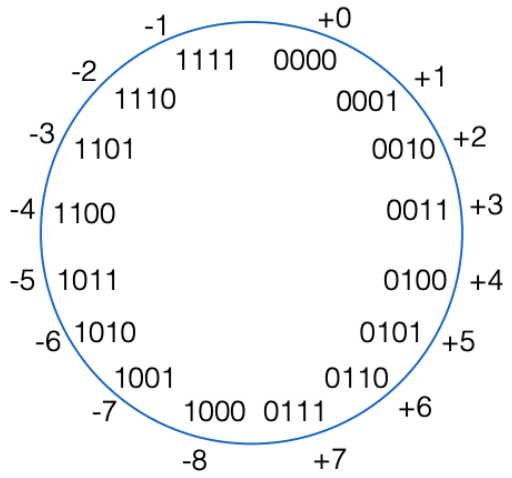


Two’s Complement System

Two’s Complement System

- ยังใช้ MSB เป็น Sign Bit สำหรับจำนวนบวก บิตนี้จะเป็น 0 และสำหรับจำนวนลบบิตนี้จะเป็น 1
- การแทนจำนวนลบทำได้โดยหาคอมพลีเมนต์ของจำนวนบวกที่มีค่า Magnitude ตรงกับจำนวลบที่ต้องการ แล้วนำค่าคอมพลีเมนต์นั้นมาบวกด้วย 1
- ตัวอย่าง: +7 จะถูกแทนด้วย 0111 และ -7 สามารถหาได้โดยนำ 0111 มาหา คอมพลีเมนต์ ซึ่งจะได้ 1000 และบวก 1 เข้าไปจะได้ผลลัพธ์คือ 1001 ดังนั้น -7 ในระบบ Two Complement จะถูกแทนด้วย 1001

Two’s Complement Number Wheel



Integer in C (cont.)

Signed Integer (Default)

```c
#include <stdio.h>
int main() {
    int num;
    scanf("%d", &num);
    printf("%d", num);
    return 0;
}
```

Code

Output

Unsigned Integer

```c
#include <stdio.h>
int main() {
    unsigned int num;
    scanf("%u", &num);
    printf("%u", num);
    return 0;
}
```

Code

Output

When you declare an integer and assign it a value

```
int x=2;
```

เมื่อประกาศคำสั่งการทำงานจะเกิดขึ้นดังนี้

- จองเนื้อที่ใน RAM เพื่อเก็บค่าตัวเลขจำนวนเต็ม
- นำชื่อตัวแปร x ไปเป็นชื่อตำแหน่ง address ของ RAM
- จัดเก็บค่าข้อมูล 2 ลงไปในตำแหน่งนั้น

x คือ Location name

2 คือ Values at Location

F45AB14 คือ Address number

- เป็น Address อื่นที่ว่างอยู่ได้ ไม่จำเป็นต้องเป็นหมายเลข F45AB14

F45AAF4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AB04	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	0
F45AB14	1	1	0	0	1	1	0	1	0	1	1	1	0	0	1	1	1	0	1	0	1	0	1	0	1	0	0	0	1	1	1	0
F45AB24	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
F45AB34	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
F45AB44	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1

When you declare an integer and assign it a value

```
int x=2;
```

เมื่อประกาศคำสั่งการทำงานจะเกิดขึ้นดังนี้

- **จองเนื้อที่ใน RAM เพื่อเก็บค่าตัวเลขจำนวนเต็ม**
- นำชื่อตัวแปร x ไปเป็นชื่อตำแหน่ง address ของ RAM
- จัดเก็บค่าข้อมูล 2 ลงไปในตำแหน่งนั้น

x คือ Location name

2 คือ Values at Location

F45AB14 คือ Address number

- เป็น Address อื่นที่ว่างอยู่ได้ ไม่จำเป็นต้องเป็นหมายเลข F45AB14

F45AAF4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AB04	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	0
F45AB14	1	1	0	0	1	1	0	1	0	1	1	1	0	0	1	1	1	0	1	0	1	0	1	0	1	0	0	0	1	1	1	0
F45AB24	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
F45AB34	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
F45AB44	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1

When you declare an integer and assign it a value

```
int x=2;
```

เมื่อประกาศคำสั่งการทำงานจะเกิดขึ้นดังนี้

- จองเนื้อที่ใน RAM เพื่อเก็บค่าตัวเลขจำนวนเต็ม
- **นำชื่อตัวแปร x ไปเป็นชื่อตำแหน่ง address ของ RAM**
- จัดเก็บค่าข้อมูล 2 ลงไปในตำแหน่งนั้น

x คือ Location name

2 คือ Values at Location

F45AB14 คือ Address number

- เป็น Address อื่นที่ว่างอยู่ได้ ไม่จำเป็นต้องเป็นหมายเลข F45AB14

x

F45AAF4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AB04	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	0
F45AB14	1	1	0	0	1	1	0	1	0	1	1	1	0	0	1	1	1	0	1	0	1	0	1	0	1	0	0	0	1	1	1	0
F45AB24	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
F45AB34	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
F45AB44	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1

When you declare an integer and assign it a value

```
int x=2;
```

เมื่อประกาศคำสั่งการทำงานจะเกิดขึ้นดังนี้

- จองเนื้อที่ใน RAM เพื่อเก็บค่าตัวเลขจำนวนเต็ม
- นำชื่อตัวแปร x ไปเป็นชื่อตำแหน่ง address ของ RAM
- **จัดเก็บค่าข้อมูล 2 ลงไปในตำแหน่งนั้น**

x คือ Location name

2 คือ Values at Location

F45AB14 คือ Address number

- เป็น Address อื่นที่ว่างอยู่ได้ ไม่จำเป็นต้องเป็นหมายเลข F45AB14

Divide by the base 2 to get the digits from the remainders:

Division by 2	Quotient	Remainder (Digit)	Bit #
(2)/2	1	0	0
(1)/2	0	1	1

$= (10)_2$

x

F45AAF4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AB04	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	0
F45AB14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
F45AB24	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
F45AB34	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
F45AB44	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1

When you declare an integer and assign it a value

```
int x=2;
```

เมื่อประกาศคำสั่งการทำงานจะเกิดขึ้นดังนี้

- จองเนื้อที่ใน RAM เพื่อเก็บค่าตัวเลขจำนวนเต็ม
- นำชื่อตัวแปร x ไปเป็นชื่อตำแหน่ง address ของ RAM
- จัดเก็บค่าข้อมูล 2 ลงไปในตำแหน่งนั้น

x คือ Location name

2 คือ Values at Location

F45AB14 คือ Address number

- เป็น Address อื่นที่ว่างอยู่ได้ ไม่จำเป็นต้องเป็นหมายเลข F45AB14

x

F45AAF4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AB04	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	0
F45AB14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
F45AB24	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
F45AB34	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
F45AB44	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1

When you declare an integer and not assign it a value

```
int y;
```

เมื่อประกาศคำสั่งการทำงานจะเกิดขึ้นดังนี้

- จองเนื้อที่ใน RAM เพื่อเก็บค่าตัวเลขจำนวนเต็ม
- นำชื่อตัวแปร y ไปเป็นชื่อตำแหน่ง address ของ RAM
- ~~จัดเก็บค่าข้อมูล 2 ลงไปในตำแหน่งนั้น~~

y คือ Location name

~~**2** คือ Values at Location~~

F45AB24 คือ Address number

- เป็น Address อื่นที่ว่างอยู่ได้ ไม่จำเป็นต้องเป็นหมายเลข F45AB24

F45AAF4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AB04	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	0
F45AB14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
F45AB24	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
F45AB34	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
F45AB44	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1

x → **F45AB04**

y → **F45AB24**

When you declare an integer and not assign it a value

```
int y;
```

F45AAF4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AB04	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	0
F45AB14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
F45AB24	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
F45AB34	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
F45AB44	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1

x

y

- When you don't initialize variables in C, their contents are unpredictable. This is what's known as "undefined behavior."
- The variables could start with random garbage values already in those memory locations.

- ค่าใน Address รหัสที่ F45AB24 จะมีค่า 10010000101011110110100000000011 หรือเท่ากับ -1867552765

ปัญหาจากอาทิตย์ที่แล้ว

ประกาศให้ตัวแปร = 0

```
#include <stdio.h>
int main() {
    int age, height, weight
    int cond1=0, cond2=0, cond3=0, cond4=0;
    int num_of_person = 50, sum_age = 0;
    float sum_height = 0.0, sum_weight = 0.0;
    for (int i = 0; i < num_of_person; i++) {
      scanf("%d %d %d", &age, &height, &weight);
      ...
```

Code

```
Age >= 20 and Height >= 160: 36
Age < 20 and Height <= 180 or Weight >= 60: 6
Age >= 30 and Weight >= 40 and Weight <= 80: 16
Age < 40 and Weight < 85 or Height <= 200: 13
Average Age: 57
Average Height: 190.56
Average Weight: 87.88
```

Output

ไม่ประกาศให้ตัวแปร = 0

```
#include <stdio.h>
int main() {
    int age, height, weight
    int cond1, cond2, cond3, cond4;
    int num_of_person = 50, sum_age;
    float sum_height, sum_weight;
    for (int i = 0; i < num_of_person; i++) {
      scanf("%d %d %d", &age, &height, &weight);
      ...
```

Code

```
Age >= 20 and Height >= 160: 1147568165
Age < 20 and Height <= 180 or Weight >= 60: -1913402690
Age >= 30 and Weight >= 40 and Weight <= 80: 17
Age < 40 and Weight < 85 or Height <= 200: 1804759613
Average Age: 36095253
Average Height: 190.56
Average Weight: 87.88
```

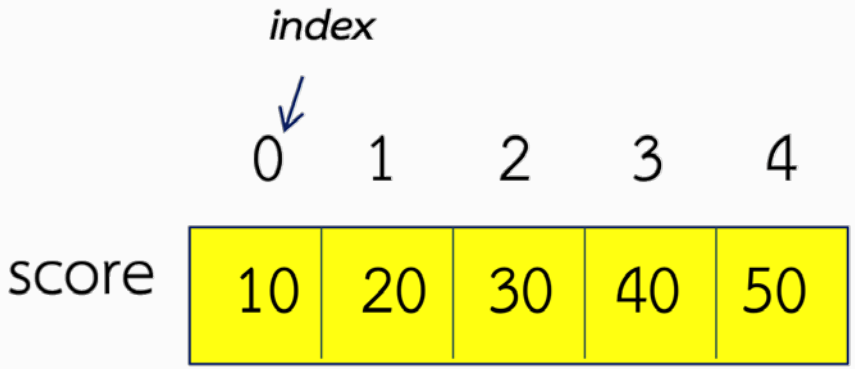
Output

Array

Lab06 - Array

Array

- **Array** เป็นโครงสร้างข้อมูลที่จัดเก็บข้อมูลเป็นลำดับแบบที่มีขนาดคงที่ของชนิดข้อมูลเดียวกัน โดยทำการจองพื้นที่ในหน่วยความจำต่อเนื่องกัน

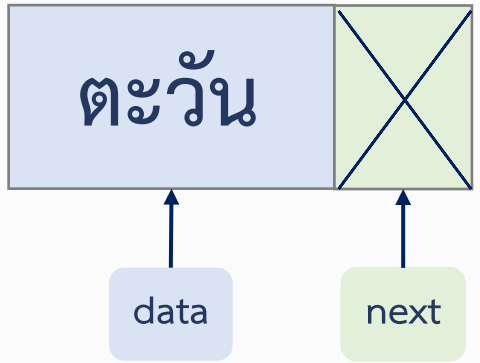


- Static Memory Allocation
  (การจัดสรรหน่วยความจำแบบคงที่)
  - เมื่อมีการประมวลผล เนื้อที่เหล่านี้ไม่สามารถขยายหรือลดลงได้
  - ไม่มีความยืดหยุ่น

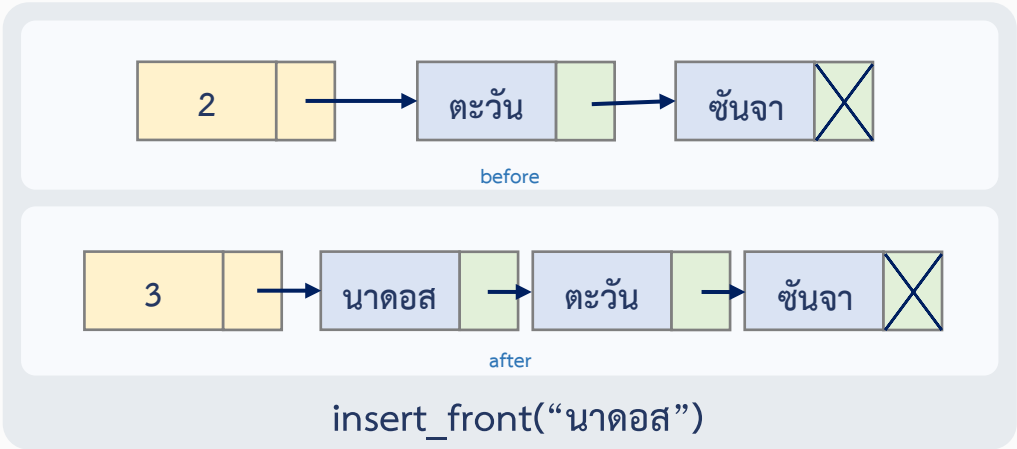
```
int score[] = {10, 20, 30, 40, 50};
score[1] = -20;
printf("%d", num[2]); // Output is 30
```

Linked List

- **Linked List** เป็นโครงสร้างข้อมูลที่องค์ประกอบต่างๆ จะถูกเก็บไว้ในโหนด แต่ละโหนดประกอบด้วยส่วนประกอบหลักสองส่วน
  - **Data:** The actual value stored in the node.
  - **Pointer:** A reference to the next node.

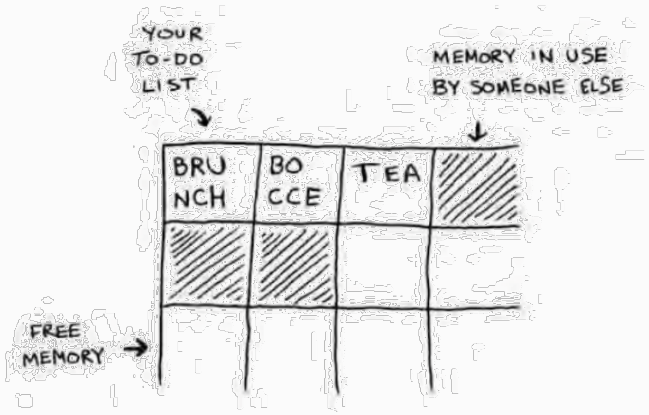


- Dynamic Memory Allocation **(การจัดสรรหน่วยความจำแบบไดนามิก)**
  - สามารถสร้างตัวแปรขึ้นได้ทุกครั้งที่ต้องการใช้ และสามารถทำลายลงเมื่อไม่ต้องการ
  - มีความยืดหยุ่น

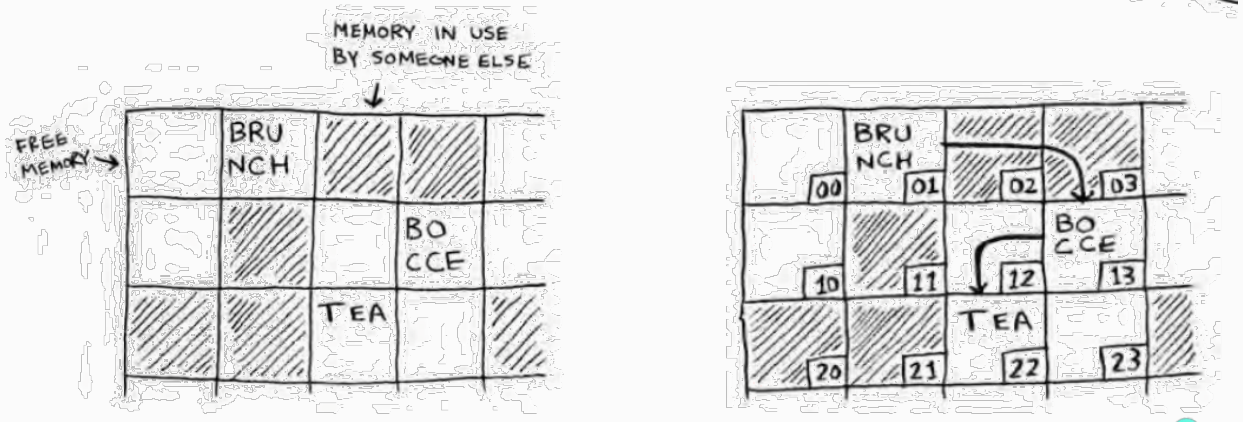


Array VS Linked List

คุณสมบัติ	Array	Linked List
ขนาด	คงที่	ไดนามิก (สามารถขยายหรือลดได้)
การจอง/จัดสรรหน่วยความจำ	ต่อเนื่อง	ไม่ต่อเนื่อง
การเข้าถึง	โดยตรง (ใช้ index)	ต่อเนื่อง (ใช้ pointer)



Array



Linked List

For simplicity

F45AAF4	1	1	1	0	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1
F45AB04	1	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	0
F45AB14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
F45AB24	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	1
F45AB34	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1
F45AB44	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	0	1	1

1000	-445285113
1004	-95889994
1008	2
1012	-1867552765
1016	65070089
1022	-725163749

Linked List

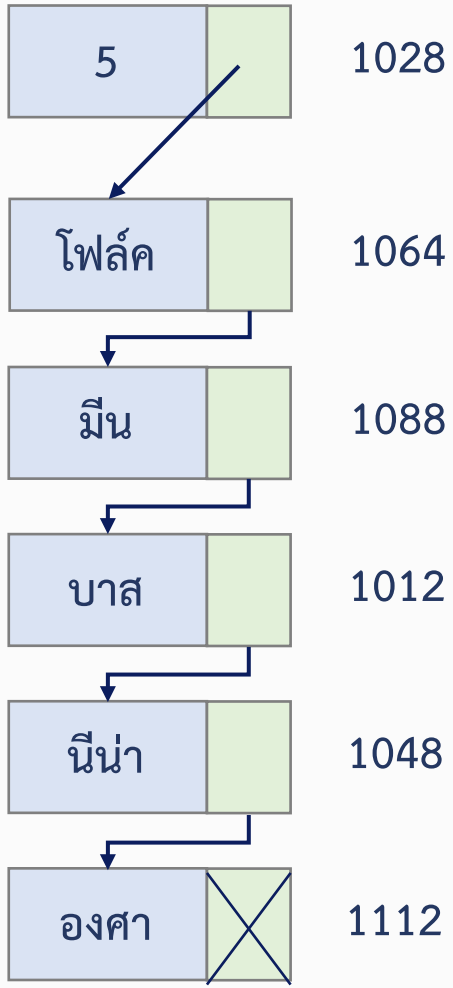
1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	data: บาส, next: 1048
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	count: 5, head: 1064
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	data: นีน่า, next: 1112
1052	. . . . .
1056	. . . . .

data

1060	. . . . .
1064	data: โฟล์ค, next: 1088
1068	. . . . .
1072	. . . . .
1076	. . . . .
1080	. . . . .
1084	. . . . .
1088	data: มีน, next: 1012
1092	. . . . .
1096	. . . . .
1100	. . . . .
1104	. . . . .
1108	. . . . .
1112	data: องศา, next: null
1116	. . . . .

นี่คือการสมมุติ ขนาดของข้อมูลจริงไม่ใช่ตามตัวอย่างสมมุติ ***

```
data = []
```



Linked List

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	. . . . .
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	count: 5, head: null
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	. . . . .
1052	. . . . .
1056	. . . . .

data

1060	. . . . .
1064	. . . . .
1068	. . . . .
1072	. . . . .
1076	. . . . .
1080	. . . . .
1084	. . . . .
1088	. . . . .
1092	. . . . .
1096	. . . . .
1100	. . . . .
1104	. . . . .
1108	. . . . .
1112	. . . . .
1116	. . . . .

นี่คือการสมมุติ ขนาดของข้อมูลจริงไม่ใช่ตามตัวอย่างสมมุติ ***


```
data = []
```

สร้าง List ขึ้นมา

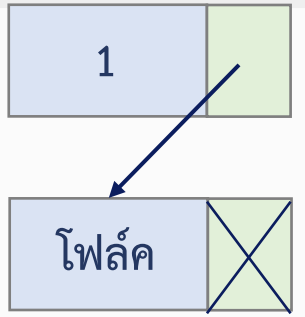
Linked List

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	. . . . .
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	count: 5, head: 1064
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	. . . . .
1052	. . . . .
1056	. . . . .

data

1060	. . . . .
1064	data: โฟล์ค, next: null
1068	. . . . .
1072	. . . . .
1076	. . . . .
1080	. . . . .
1084	. . . . .
1088	. . . . .
1092	. . . . .
1096	. . . . .
1100	. . . . .
1104	. . . . .
1108	. . . . .
1112	. . . . .
1116	. . . . .

นี่คือการสมมุติ ขนาดของข้อมูลจริงไม่ใช่ตามตัวอย่างสมมุติ ***



1028 (Head Node)

1064

มีน 1088

บาส 1012

นีน่า 1048

องศา 1112

```
data.append("โฟล์ค")
```

เพิ่มชื่อ “โฟล์ค”

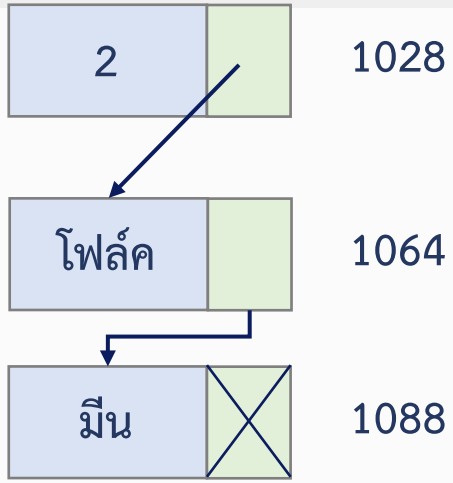
Linked List

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	. . . . .
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	count: 5, head: 1064
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	. . . . .
1052	. . . . .
1056	. . . . .

data

1060	. . . . .
1064	data: โฟล์ค, next: 1088
1068	. . . . .
1072	. . . . .
1076	. . . . .
1080	. . . . .
1084	. . . . .
1088	data: มีน, next: null
1092	. . . . .
1096	. . . . .
1100	. . . . .
1104	. . . . .
1108	. . . . .
1112	. . . . .
1116	. . . . .

นี่คือการสมมุติ ขนาดของข้อมูลจริงไม่ใช่ตามตัวอย่างสมมุติ ***




```
data.append("มีน")
```

เพิ่มชื่อ “มีน”

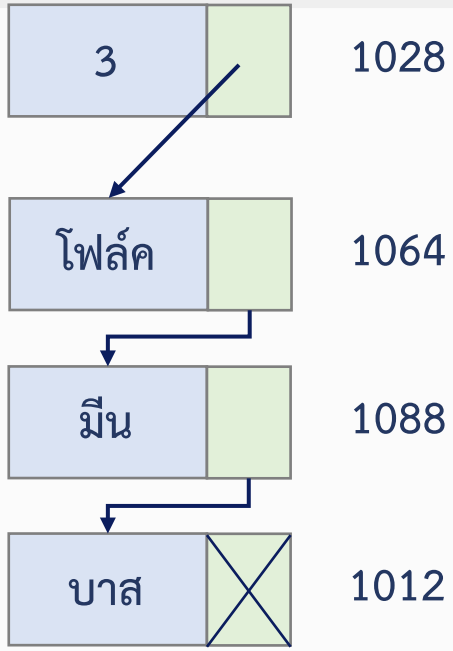
Linked List

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	data: บาส, next: null
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	count: 5, head: 1064
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	. . . . .
1052	. . . . .
1056	. . . . .

data

1060	. . . . .
1064	data: โฟล์ค, next: 1088
1068	. . . . .
1072	. . . . .
1076	. . . . .
1080	. . . . .
1084	. . . . .
1088	data: มีน, next: 1012
1092	. . . . .
1096	. . . . .
1100	. . . . .
1104	. . . . .
1108	. . . . .
1112	. . . . .
1116	. . . . .

นี่คือการสมมุติ ขนาดของข้อมูลจริงไม่ใช่ตามตัวอย่างสมมุติ ***




```
data.append("บาส")
```

เพิ่มชื่อ “บาส”

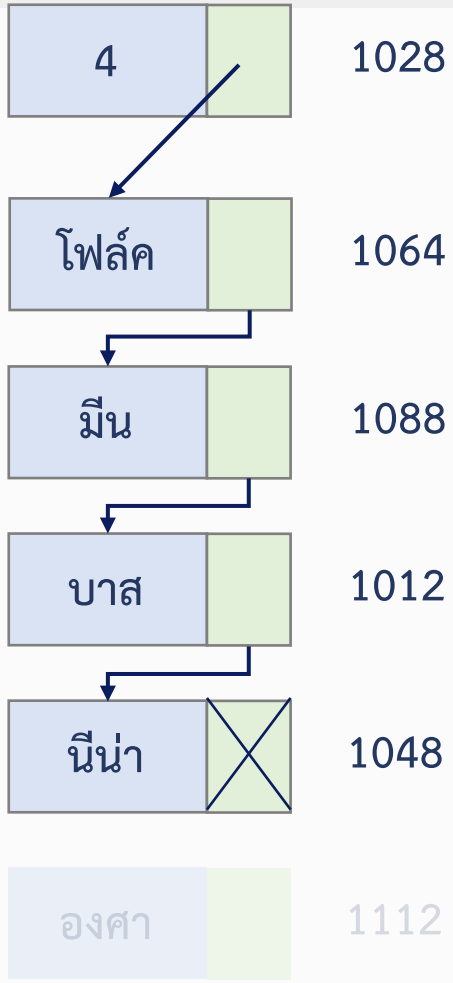
Linked List

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	data: บาส, next: 1048
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	count: 5, head: 1064
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	data: นีน่า, next: null
1052	. . . . .
1056	. . . . .

data

1060	. . . . .
1064	data: โฟล์ค, next: 1088
1068	. . . . .
1072	. . . . .
1076	. . . . .
1080	. . . . .
1084	. . . . .
1088	data: มีน, next: 1012
1092	. . . . .
1096	. . . . .
1100	. . . . .
1104	. . . . .
1108	. . . . .
1112	. . . . .
1116	. . . . .

นี่คือการสมมุติ ขนาดของข้อมูลจริงไม่ใช่ตามตัวอย่างสมมุติ ***




```
data.append("นีน่า")
```

เพิ่มชื่อ “นีน่า”

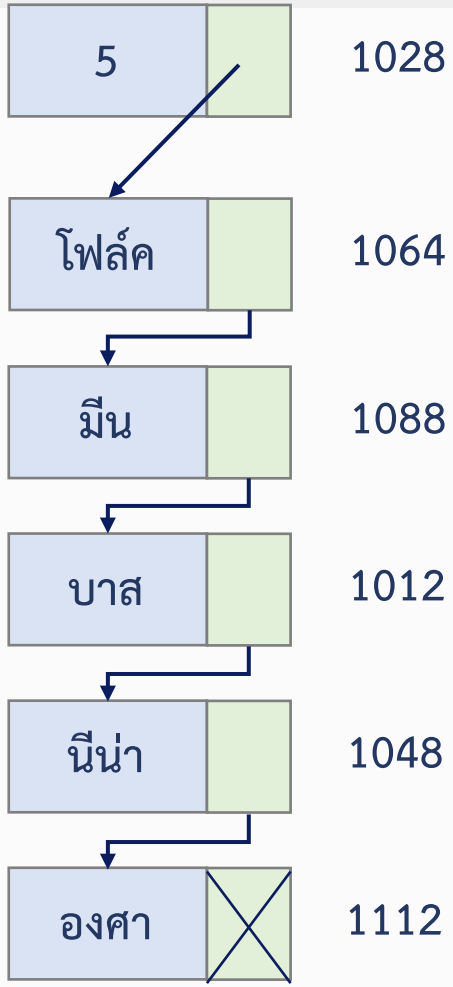
Linked List

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	data: บาส, next: 1048
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	count: 5, head: 1064
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	data: นีน่า, next: 1112
1052	. . . . .
1056	. . . . .

data

1060	. . . . .
1064	data: โฟล์ค, next: 1088
1068	. . . . .
1072	. . . . .
1076	. . . . .
1080	. . . . .
1084	. . . . .
1088	data: มีน, next: 1012
1092	. . . . .
1096	. . . . .
1100	. . . . .
1104	. . . . .
1108	. . . . .
1112	data: องศา, next: null
1116	. . . . .

นี่คือการสมมุติ ขนาดของข้อมูลจริงไม่ใช่ตามตัวอย่างสมมุติ ***




```
data.append("องศา")
```

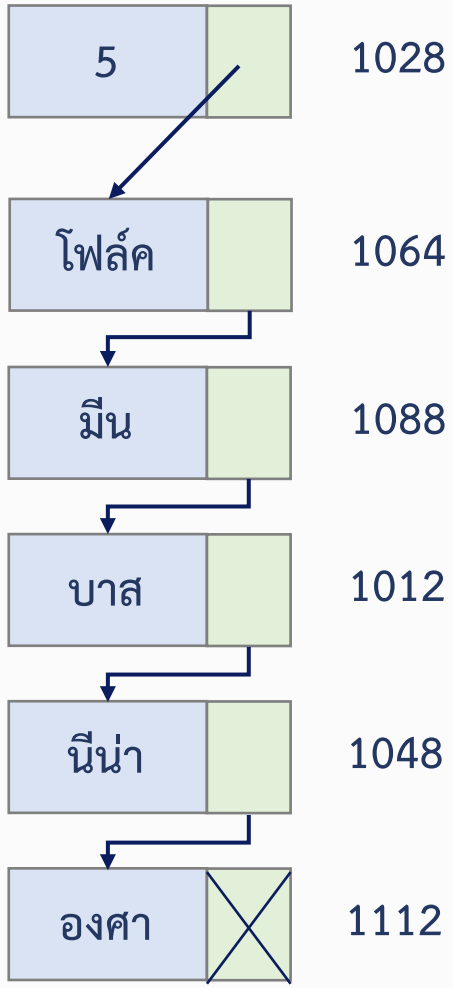
เพิ่มชื่อ “องศา”

Linked List

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	data: บาส, next: 1048
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	count: 5, head: 1064
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	data: นีน่า, next: 1112
1052	. . . . .
1056	. . . . .

data

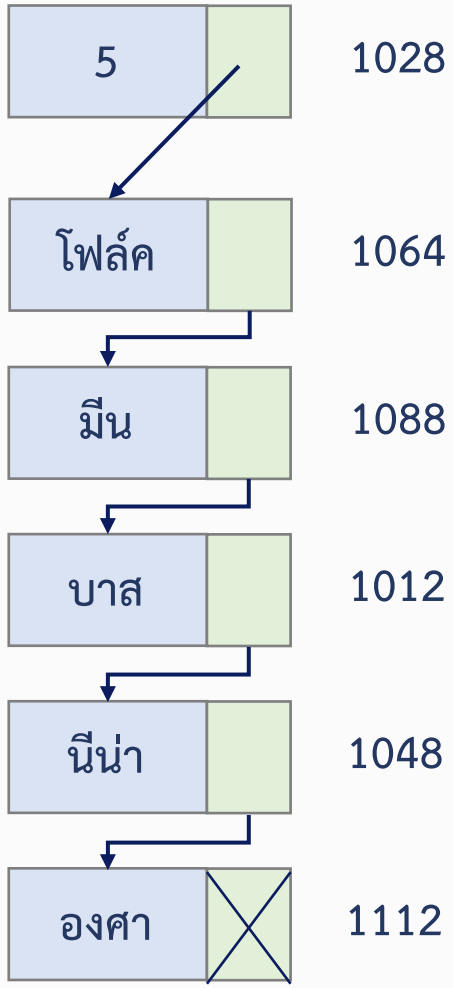
1060	. . . . .
1064	data: โฟล์ค, next: 1088
1068	. . . . .
1072	. . . . .
1076	. . . . .
1080	. . . . .
1084	. . . . .
1088	data: มีน, next: 1012
1092	. . . . .
1096	. . . . .
1100	. . . . .
1104	. . . . .
1108	. . . . .
1112	data: องศา, next: null
1116	. . . . .



นี่คือการสมมุติ ขนาดของข้อมูลจริงไม่ใช่ตามตัวอย่างสมมุติ ***

Linked List

- ถ้าข้อมูลเยอะๆ เข้าถึงข้อมูลจะช้ามากๆ
- สามารถขยายตัวเองได้เรื่อยๆ

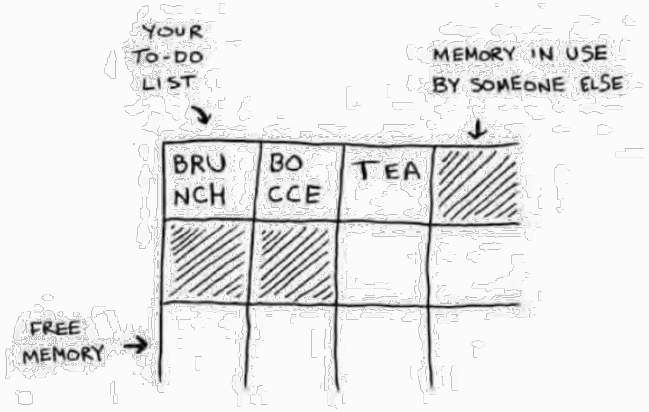


Array

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	66070199
1016	66070200
1020	66070207
1024	66070210
1028	66070212
1032	0
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	. . . . .
1052	. . . . .
1056	. . . . .

num

- Array เป็นโครงสร้างข้อมูลที่จัดเก็บข้อมูลเป็นลำดับแบบที่มีขนาดคงที่ของชนิดข้อมูลเดียวกัน
  - จองพื้นที่ในหน่วยความจำต่อเนื่องกัน
  - เนื้อที่เหล่านี้ไม่สามารถขยายหรือลดลงได้
  - ไม่มีความยืดหยุ่น



Array: How it works?

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	. . . . .
1016	. . . . .
1020	. . . . .
1024	. . . . .
1028	. . . . .
1032	. . . . .
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	. . . . .
1052	. . . . .
1056	. . . . .

1

เมื่อเราสั่งจอง Array, โปรแกรมจะคำนวณ**ขนาดข้อมูลทั้งหมด**ที่ต้องการจะเก็บและจะไปขอพื้นที่ใน memory จาก OS

- **ขนาดข้อมูลทั้งหมด**
  = ขนาดของชนิดตัวแปรที่ต้องการเก็บ x จำนวนสมาชิกใน Array

คุณฉงนต้องการเก็บรหัสนักศึกษา 6 คนเป็นประเภท int ทั้งหมด 6 ตัว (int มีขนาด 4 byte) แต่ว่าคุณฉงนใส่รหัสนักศึกษาไปแค่ 5 คน

```
int num[6] = {66070199, 66070200, 66070207, 66070210, 66070212};
```

= sizeof(int) x 6

= 4 x 6

= 24

ขนาดของ Array ที่คุณฉงนต้องการจะเก็บคือ 24 bytes

Array: How it works?

2

```
int num[6] = {66070199, 66070200, 66070207, 66070210, 66070212};
```

เรารู้แล้วว่าขนาดของ array นั้นมีขนาด 24 bytes

- step ต่อมาตัวโปรแกรมก็จะขอพื้นที่ว่างใน memory จาก OS (Operating System) มา 24 bytes
- OS เจอพื้นที่ว่างแล้ว, OS เลยส่งเป็น memory address กลับมาให้
- ในตัวอย่างจะเป็นตำแหน่งที่ 1012 นั่นก็แปลว่าตั้งแต่ตำแหน่งที่ 1012 ถึงตำแหน่งต่อไปอีก 24 bytes ก็จะเป็นพื้นที่ของ array num ถึงตำแหน่ง 1035
  - 1012+24-1 = 1035
    - สาเหตุที่ต้อง -1 เพราะว่าเราต้องนับ 1012 ไปด้วย

Array: How it works?

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012	66070199
1016	66070200
1020	66070207
1024	66070210
1028	66070212
1032	0
1036	. . . . .
1040	. . . . .
1044	. . . . .
1048	. . . . .
1052	. . . . .
1056	. . . . .

num

3

```
int num[6] = {66070199, 66070200, 66070207, 66070210, 66070212};
```

เราได้จองตำแหน่งที่ 1012-1035 ไว้เก็บ array num แล้ว ที่นี้ตัวโปรแกรมก็จะเขียนข้อมูลตัวเลขเข้าไปทีละตัว

- 1012-1015 (num[0]) = 66070199
- 1016-1019 (num[1]) = 66070200
- 1020-1023 (num[2]) = 66070207
- 1024-1027 (num[3]) = 66070210
- 1028-1031 (num[4]) = 66070212
- 1032-1035 (num[5]) = 0
  - ถ้าเราไม่ได้ประกาศค่าเอาไว้, default จะเป็น 0

Array: Accessing Data

1000	. . . . .
1004	. . . . .
1008	. . . . .
1012 num[0]	66070199
1016 num[1]	66070200
1020 num[2]	66070207
1024 num[3]	66070210
1028 num[4]	66070212
1032 num[5]	0
1036	. . . . .
1040	. . . . .
1044	. . . . .

num

```
int num[6] = {66070199, 66070200, 66070207, 66070210, 66070212};
```

ถ้าคุณฉงนต้องการ **ดูข้อมูลใน array** ในโปรแกรมจะเอาค่า address ของข้อมูลตัวแรกไปบวกกับผลลัพธ์ของขนาดของข้อมูลคูณกับลำดับของข้อมูล

ตำแหน่งของข้อมูลตามลำดับที่ n

= ตำแหน่งของตัวแรกสุด + (ขนาดของประเภทข้อมูล x ลำดับ)

ถ้าคุณฉงนต้องการดูข้อมูล num[2]

= &num + (sizeof(int) * 2)

= 1012 + (4 * 2)

= 1012 + 8

= 1020 = 65070207

The reason why index start from zero is that index is used as an offset. Suppose we have an array Arr = [1,2,3,4,5] Arr[0] actually means that first element is 0 element away from the memory location where the arr points as elements in an array are stored in a contiguous manner.

Accessing Data

ถ้าคุณฉงนต้องการเข้าถึงข้อมูลตำแหน่งที่ 100,000

Linked List

```
loc = 100000;
i = 0;
while (i++<loc)
    node = node->next;
printf("%d", node->data);
```

- วิ่งไปเรื่อยๆ จนกว่าจะถึงตัวที่ 100,000
- ช้า

Array

```
loc = 100000;
add = &num + (sizeof(int) * loc);
printf("%d", *add);
```

- คำนวณค่าของตำแหน่งข้อมูลได้เลย
- รู้ผลทันที

Question???