

CP 111 Computer Programming (2/2554 Section B01-B02)

Handout 6

หัวข้อ:

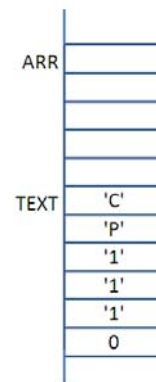
Pointers and Arrays (Chapter 16 textbook)

- Arrays (Chapter 16.3.1, 16.3.2)
- Strings in C (Chapter 16.3.4)
- Pointers (Chapter 16.2.1, 16.2.2, 16.2.4)
- Passing a reference using pointers (Chapter 16.2.3)
- Passing arrays as function arguments (Chapter 16.3.3)
- The Relationship between arrays and pointers in C (Chapter 16.3.5)

Arrays in C

1. ทบทวน Arrays (อาร์เรย์) in LC-3 assembly programming

```
.orig x3000
LEA R0,ARR
LDR R1,R0,#0
.....
ARR .blkw 5
TEXT .stringz "CP111"
.end
```



2. โดยทั่วไป อาร์เรย์ หมายถึง กลุ่มของข้อมูลที่มี data type ชนิดเดียวกันและถูกเก็บเรียงต่อกันในหน่วยความจำ

3. Array declaration

Syntax: <array element type> <array_name>[array_size];

เช่น

```
{
    int x;
    int ARR[5]; /* ประกาศตัวแปรอาร์เรย์ของ int ขนาด 5 element ชื่อ ARR */
    char text[6];
    double xx[4];
}
```

นอกจากนี้การประกาศตัวแปรแบบอาร์เรย์สามารถกำหนดค่าเริ่มต้น (initialization) ได้ดังตัวอย่างเช่น

```
{
    int ARR[5]={1,10,2,20,3};
    double xx[4]={1.1, 1.2, 1.3, -1000.12};
    int y[]={1, 3, 5, 7};
}
```

4. การอ้างถึงตัวแปรแบบอาร์เรย์

Syntax: <array_name>[<array_index>]

โดยที่

(4.1) <array_index> จะเป็นได้ทั้งตัวเลข (literal) จำนวนเต็มที่ไม่มากกว่าหรือเท่ากับ 0 หรืออาจจะเป็น expression ที่คำนวณค่าออกมาได้เป็น int

(4.2) ค่า <array_index> จะมีค่าตั้งแต่ 0, 1, 2, ..., N-1 โดยที่ N=array_size

(4.3) <array_index> = 0 หมายถึงการอ้างค่า element แรก และถ้าเท่ากับ array_size-1 หมายถึงการอ้างค่า element สุดท้าย

ตัวอย่าง

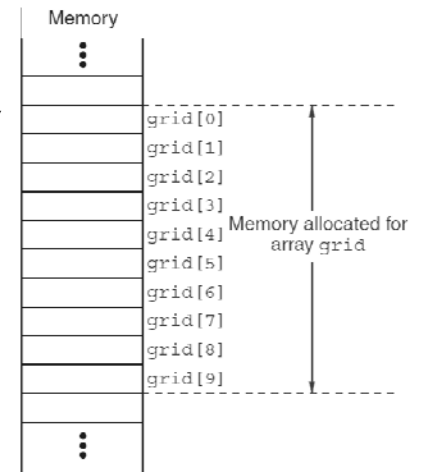
```
{
int i;
int grid[10]; /* ถ้ามองในหน่วยความจำแล้วอาร์เรย์ grid จะแสดงดังรูปด้านข้าง */

/* นำค่าใน element แรก (index=0) และค่าใน element สุดท้าย (index=9)
   มาบวกกันแล้วนำมาเก็บที่ element ที่สอง (index=1) */
grid[1]=grid[0]+grid[9];

grid[6]=grid[3]+1;

i=3; /* ค่า array index สามารถเป็น expression ที่คืนค่า int ได้ */
grid[i+2]=grid[i]+grid[i-2];
grid[i]=grid[i*2]+grid[i/2]+grid[i*i];

/* แต่ระวังอย่าให้เกินขอบเขตของ array ซึ่ง Compiler จะตรวจไม่เจอ และจะเกิด RUN-TIME ERROR */
i=9;
grid[i]=grid[i+1]; /* i+1 เท่ากับ 10 จะเกินขอบเขต 0, 1, ..9 */
}
```



5. โจทย์ตัวอย่าง: จงระบุผลลัพธ์ของโปรแกรมข้างล่าง เมื่อข้อมูลที่ป้อนเข้ามาคือ 1 2 3 4 5 ตามลำดับ

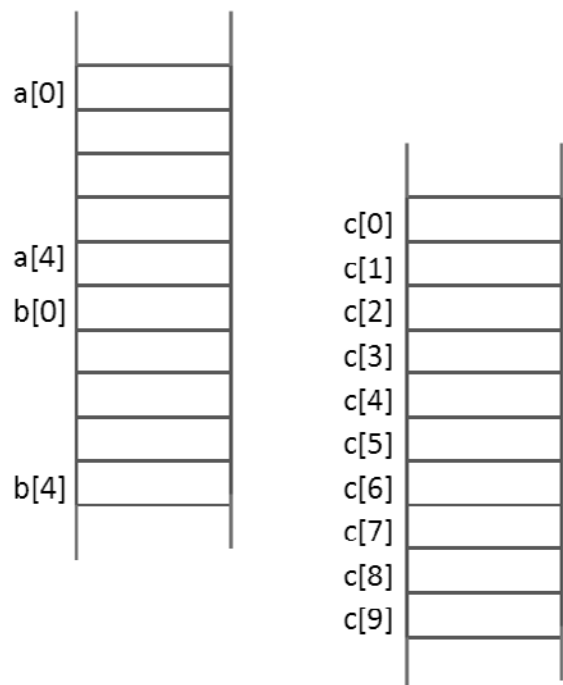
```
#include <stdio.h>
int main()
{
int i;
int a[5]={5,4,3,2,1};
int b[5],c[10];

for(i=0;i<5;i++)
scanf("%d",&b[i])

for(i=0;i<10;i++)
{
if(i%2==0)
c[i]=a[i];
else
c[i]=b[i]+c[i-1];
}

for(i=0;i<10;i++)
printf("%d ",c[i])

return 0;
}
```



6. โจทย์ตัวอย่าง: จงเขียนโปรแกรมเพื่อคำนวณหาค่าเฉลี่ย (Average) และค่าเบี่ยงเบนมาตรฐาน (Standard deviation) ของข้อมูลจำนวนเต็มบวก โดยข้อมูลนำเข้าจะเป็นจำนวนเต็มบวกหรือศูนย์บรรทัดละตัว และสิ้นสุดด้วยค่าลบ ตัวอย่างของข้อมูลนำเข้า เช่น 8 12 5 18 2 -1

เมื่อสิ้นสุดข้อมูลนำเข้าให้แสดงค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานออกสู่หน้าจอที่ละบรรทัด โดยให้แสดงเป็นเลขจุดทศนิยม 2 ตำแหน่ง และกำหนดให้ข้อมูลที่เข้ามาจะไม่เกิน 100 ตัว และกำหนดให้ ค่าเบี่ยงเบนมาตรฐาน (Standard deviation) ของข้อมูล $x_1, x_2, \dots, x_N$ คำนวณโดยใช้สูตร $STD = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$ โดยที่ $\bar{x}$ คือค่าเฉลี่ย

```
#include .....

#include .....
int main()
{
    .....

    int i,N,data,sum;
    double .....

    .....

    .....

    while(1) /* ลูปแรก รับค่า -> เก็บลง array -> คำนวณ sum of xi -> คำนวณ xbar (Mean) */
    {
        scanf("%d",&data);

        if(.....)
        {
            .....
        }

        .....

        .....

    }

    .....

    ..... /* ลูปแรก sum of (xi-xbar)^2 -> คำนวณ STD */

    for(.....)
    {
        .....

        .....

    }

    .....

    printf("%.2f\n",mean);
    printf("%.2f\n",std);
    return 0;
}
```

Strings in C

1. String ในภาษา C จะหมายถึงอาร์เรย์ของข้อมูลชนิด char โดยที่แต่ละ element ของอาร์เรย์จะเก็บรหัสแอสกีของตัวอักษร และจุดสิ้นสุดของ string จะระบุโดย NULL character (ASCII มาค่าเท่ากับ 0) จึงนิยมเรียกว่า NULL-terminated character

2. Input/Output: การใช้ printf() และ scanf() กับสตริง จะใช้ conversion specific %s

(2.1) printf("%s", array_name (array of type char));

(2.2) scanf("%s", array_name (array of type char));

%s ที่ป้อนเป็นอาร์กิวเมนต์ของฟังก์ชัน scanf() จะอ่าน string of char จากคีย์บอร์ดจนกว่าจะเจอ **Whitespace character** ซึ่งได้แก่ SPACE (ASCII = ' ' หรือ 0x20), TAB (ASCII = '\t' หรือ 0x09), Newline (ASCII = '\n') Carriage Return (ASCII=0x0D), VERTICAL TAB (ASCII = 0x0B) , Form Feed (ASCII = 0x0C)

(2.3) gets() เนื่องจาก scanf("%s") จะอ่านสตริงจนกว่าจะเจอ whitespace ทำให้ไม่สามารถอ่านสตริงอินพุตที่มี space ภายในได้ จึงนิยมใช้ ฟังก์ชัน gets() ซึ่งเป็น standard library function ในการรับสตริงอินพุตจากคีย์บอร์ด แทนรูปแบบ function call: gets(array_name);

```
char word[10],input[10];
char str[10]="Hi"; /* initialization with 'H','i',0 */
word[0]='H';
word[1]='e';
word[2]='l';
word[3]='l';
word[4]='l';
word[5]='o';
word[6]=0;
printf("%s",word); /* แสดง Hello ที่หน้าจอ */
scanf("%s",input); /* สมมติป้อน Pradit แล้วกด Enter อาร์เรย์ input จะเก็บ 'P','r',...'t', 0 */
printf("%s!!!,%s",str,input); /* แสดง Hi!!!, Pradit */
gets(input); /* ถ้าป้อน My name is Pradit. แล้ว gets จะรับทั้งสตริงเข้ามาได้ */
printf("%s\n",input);
```

4. การเขียนโปรแกรมจัดการกับสตริงจะเหมือนกับการจัดการกับอาร์เรย์ แต่เงื่อนไขในการหยุดลูปมักจะตรวจสอบว่าเจอ NULL character (ASCII =0) หรือยัง

5. โจทย์ตัวอย่าง: จงระบุผลลัพธ์ของโปรแกรมข้างล่างเมื่อป้อนสตริง "ab12c04" แล้วกด Enter

```
#include <stdio.h>
int main()
{ char input[100];
  int i;
  gets(input);
  i=0;
  while(input[i]!=0)
  {
    if(input[i]>='0' && input[i]<='9')
    {
      printf("%c",input[i]);
    }
    i++;
  }
  printf("\n%d\n",i);
  return(0);
}
```

6. โจทย์ตัวอย่าง: จงเขียนโปรแกรมที่รับอินพุตเป็นสตริง 1 ข้อความ หลังจากนั้นให้แปลงตัวอักษรในสตริง ที่เป็นตัวอักษรใหญ่ ('A' - 'Z') ให้กลายเป็นตัวอักษรตัวเล็ก ('a' - 'z') โดยที่ตัวอื่นๆ ไม่มีการเปลี่ยนแปลง หลังจากนั้นให้แสดงสตริงที่แปลงแล้วออกสู่หน้าจอ เช่น ถ้าสตริง Ab0+cD1= ถูกป้อนเข้ามา หลังจากแปลงสตริงแล้วจะแสดง ab0+cd1=

```
#include <stdio.h>
int main()
{
    .....
    int i;

    .....

    for(..... )
    {
        if(.....)
        {
            .....
        }
    }
    printf(.....);

    return(0);
}
```

7. โจทย์ตัวอย่าง: จงเขียนโปรแกรมที่รับอินพุตเป็นสตริง 1 ข้อความ หลังจากนั้นให้แปลงเป็นสตริงใหม่ที่ตัวอักษรนั้นจะเรียงสลับจากท้ายไปหน้า แล้วแสดงออกสู่หน้าจอ เช่น ถ้าป้อน "Hi, my name is C." โปรแกรมจะแสดง ".C si eman ym ,iH"

```
#include <stdio.h>
int main()
{
    char instr,outstr;
    int i,last,k;

    gets(instr); /* แปลง instr -> ostr โดยสลับตัวอักษรจากท้ายไปหน้า */
    /* หา array index ของตัวอักษรท้ายสุด ใน instr เก็บไว้ที่ last */
    .....

    while(..... )
    {
        .....
    }
    last=.....
    i=0;
    for(k=last; ..... )
    {
        .....
        .....
    }

    .....
    printf("%s",ostr);
    return(0);
}
```

โจทย์ปัญหา (เพิ่มเติม)

1. กำหนดให้ข้อมูลนำเข้าเป็นจำนวนเต็มบวกในช่วง 0-99 เข้ามาบรรทัดละตัว และสิ้นสุดด้วยค่าลบ จงเขียนโปรแกรมเพื่อ นับว่าค่าใดในบรรดาค่า 0-99 ของข้อมูลนำเข้าถูกป้อนเข้ามามากที่สุด โดยให้แสดงค่าตัวเลข และจำนวนครั้งที่เจอออกสู่ หน้าจอที่ละบรรทัด เช่น ถ้าป้อน 41 12 31 12 31 12 31 12 12 41 31 -10000 โปรแกรมจะแสดงผลลัพธ์เป็นค่า 12 และ ค่า 5 (มีข้อมูล 12 อยู่มากที่สุดทั้งหมด 5 ตัว) ให้สมมติว่าจะไม่มีข้อมูลที่ถูกป้อนเข้ามามีจำนวนครั้งที่มากที่สุดมีค่าเดียวเสมอ

```
#include <stdio.h>
int main()
{
.....
int x,i,max_i,max_count;

    for(i=0;.....)
        ..... /*เคลียร์ค่า data_count[i]=0 i=0, ...,99*/

    while(1) /* วนลูป รับค่า จนกว่าเจอค่าลบ ถ้าข้อมูลคือ x (0-99) ให้เพิ่มค่า data_count[x] ไป 1 */
    {   scanf("%d",&x);
        if(.....)
        {
            .....
        }
        data_count[.....]=.....
    }
    /* วนลูปตรวจสอบว่า data_count[..] ตัวใดที่มีค่ามากที่สุด ถ้าคือ data_count[i] หมายความว่า i ถูกป้อนเข้ามามากที่สุด */
    max_count=.....

    for(i=0;.....)
    {
        if(.....)
        {
            max_count=.....
            max_i=.....
        }
    }
    printf("%d\n%d\n",max_i,max_count);
    return(0);
}
```

2. Exercise 16.1 textbook: Pig Latin จงเขียนโปรแกรมที่รับสตริงที่เป็นคำภาษาอังกฤษเข้ามา 1 คำหลังจากนั้นให้แปลงข้อความนี้ให้เป็นข้อความแบบ Pig Latin ซึ่งการแปลงทำได้โดยลบตัวอักษรตัวแรกของข้อความออกไป แล้วนำไปต่อท้าย หลังจากนั้นให้ต่อท้ายด้วย “ay” เช่น ถ้าอินพุตคือข้อความ “Hello” หลังจากแปลงเป็น Pig latin แล้วจะได้ “elloHay” นอกจากนี้ให้สมมติว่าความยาวของคำที่ป้อนเข้ามาจะไม่เกิน 50 ตัวอักษร

3. Exercise 16.2 textbook: จงเขียนโปรแกรมที่รับข้อมูลจำนวนเต็มจากคีย์บอร์ดไปเรื่อยๆ จนกว่าจะเจอข้อมูลที่ซ้ำกับตัวเลขที่เคยป้อนก่อนหน้านี้ หลังจากนั้นให้แสดงผลรวมของข้อมูลที่ได้อ้อนเข้ามาโดยไม่นับตัวสุดท้ายที่ซ้ำ เช่น ถ้าข้อมูลที่ป้อนเข้ามาคือ 5 45 0 -6 45 ผลลัพธ์ที่แสดงที่หน้าจอคือ 44 ซึ่งเท่ากับ $5+45+0+(-6)$ นอกจากนี้ให้สมมติว่าข้อมูลที่ป้อนเข้ามาจะไม่เกิน 100 ตัว ก็จะเจอตัวซ้ำ