

สัปดาห์ที่ 3 – ตัวแปรแถวลำดับ และวงวน

การเตรียมความพร้อมสู่ สอน. คอมพิวเตอร์ ค่าย 1 ระยะที่ 1

ห้องกุหลาบเพชร โรงเรียนสวนกุหลาบวิทยาลัย

ศิริระ ทรงพลโรจนกุล

ความเดิมตอนที่แล้ว

- รู้จัก ตัวดำเนินการในภาษาซี
- ใช้ if ได้

วันนี้ มีอะไร

- ตัวแปรแถวลำดับ
- วงวน

ตัวแปรแวดล้อม

ประกาศตัวแปรจำนวนเต็ม 3 ตัว ประกาศ ไปได้

```
int a,b,c;
```

ประกาศตัวแปรจำนวนเต็ม 5 ตัว ประกาศ ไปได้

```
int a,b,c,d,e;
```

ประกาศตัวแปรจำนวนเต็ม 10 ตัว ประกาศ ไปได้

```
int a,b,c,d,e,f,g,h,i,j,k;
```

ประกาศตัวแปรจำนวนเต็ม 20 ตัว ประกาศ ไปได้

```
int a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t;
```

จะดีเหรอ ??

ตัวแปรแถวลำดับ

ประเภท ชื่อตัวแปร[ขนาด];

ตัวอย่าง

```
int a[5]; //a มีขนาด 5 โดยมี a[0],a[1],a[2],a[3],a[4]
int x[15],y[101];
int z[5]={3,4,8,9,0};    // array z เก็บค่าตามที่กำหนด
int i[]={0,2,4,6,8,10};  // i ก็จะมีขนาด 6 โดยอัตโนมัติ
double p[5];
char s[3]={'a','A','e'};
char str[]="SUANKULARB";
```


เกี่ยวกับตัวแปรแถวลำดับ

```
int A[127];
```

- ตัวแปรแถวลำดับ มีชื่อภาษาอังกฤษว่า **array**
- เลขที่อยู่ใน [] ตอนประกาศตัวแปร คือ ขนาดของ อาร์เรย์
- เลขที่อยู่ใน [] เรียกว่า **index**
- Array มี **index** เริ่มที่ **0..n-1** (ในตัวอย่างนี้ ก็จะมี A[0],A[1],...,A[125],A[126])

ตัวอย่าง

```
int a[]={1,2,3,4,5};  
printf("%d\n",a[1]+a[2]);  
a[3]=a[1]+a[2];  
printf("%d\n",a[3]-a[0]);  
printf("%d\n",a[5]);  
printf("%d\n",a[-1]);  
printf("%d\n",a[1.0]);  
printf("%d\n",a);
```


วงวน

ต้องการหาผลบวก $1 - 100$ ทำไงดี

ใช้สูตร ก็ได้สินะ

งั้นหาผลรวมบนอาเรย์ ทำไงดี

$A[0]+A[1]+A[2]+A[3]+A[4]$ แค่นี้ก็เหนื่อยและ

วงวนช่วยได้

ประเภทของ วงวน ในภาษาซี

วงวน หรือ loop ในภาษาซี มี 3 แบบ

1. for loop
2. while loop
3. do-while loop

for loop

for(ค่าเริ่มต้น ; เงื่อนไขคงการทำซ้ำ ; การเพิ่มค่า)
คำสั่งที่จะให้ทำซ้ำ

การทำงานของ for loop

1. ทำตรงส่วนค่าเริ่มต้น
2. ตรวจสอบเงื่อนไข
3. ถ้าเป็นจริง → ทำตามคำสั่ง → ทำตามคำสั่งเพิ่มค่า → ทำข้อ 2 ซ้ำ
ถ้าเป็นเท็จ → ออกจาก loop

$$a[0] + a[1] + a[2]$$

```
→ int a[3]={1,2,7};  
→ int i,sum=0;  
→ for(i=0;i<3;i++)  
    → sum+=a[i];  
→
```

~~0~~

i

~~100~~

sum

การทำงานของวงวน

1. ตั้งค่าตั้งต้น (i=0)
2. ตรวจสอบเงื่อนไข (i<3 ? → T)
3. ทำคำสั่ง (sum+=a[i])
4. เพิ่มค่า (i++ → i=1)
5. ตรวจสอบเงื่อนไข (i<3 ? → T)
6. ทำคำสั่ง (sum+=a[i])
7. เพิ่มค่า (i++ → i=2)
8. ตรวจสอบเงื่อนไข (i<3 ? → T)
9. ทำคำสั่ง (sum+=a[i])
10. เพิ่มค่า (i++ → i=3)
11. ตรวจสอบเงื่อนไข (i<3 ? → F)
12. จบ loop

for loop

```
int i,sum=0,n=10;  
for(i=1;i<=n;i++)  
    sum+=i;
```

สามารถประกาศตัวแปรใน loop
แบบนี้ได้ แต่ตัวแปรที่ประกาศตรงลูป
จะใช้ได้แค่ ภายในลูป เท่านั้น

```
int sum=0;  
for(int i=1;i<=10;i++)  
    sum+=i;
```


ตัวอย่าง

```
int sum=0;  
for(int i=1;i<=10;i++)  
    sum+=i;
```

```
int sum=0,j=0;  
for(int i=1;i<=100;j++)  
    sum+=i;
```

```
int sum=0;  
for(int i=1;i<=10;++i)  
    sum+=i;
```

```
int sum=0,j=0;  
for(int i=1,false;j++)  
    sum+=i;
```

```
int sum=0;  
for(int i=1;i<=10;i+=2)  
    sum+=i;
```

```
int sum=0,i=1;  
for(;i<=10;i++)  
    sum+=i;
```

```
int sum=0;  
for(int i=1;i<=10 && i%2!=0;i++)  
    sum+=i;
```

เกี่ยวกับ loop -- 1

```
for( A ; B ; C )  
    sum+=i;
```

A: initial ค่าใส่ได้มากกว่าตัวเดียว

for(int a=0;.. / for(i=0,j=0;..

B: condition เป็นประโยคเงื่อนไข แบบที่ใช้ใน if

a>0&&b<0 / a / true

C: increase,decrease ใส่ได้มากกว่าหนึ่งอย่างเช่นกัน

i++,++j / i*=2,j++,k-

ทุกอัน สามารถละไว้ได้


```
int s1=0,s2=0;
int i,j,k;
for(i=1,j=1; i<5 ;i++,j*=2)
    s1+=i;
    s2+=j;
```

s1 = ?

s2 = ?

```
int s1=0,s2=0;
int i,j,k;
for(i=1,j=1; i<5 ;i++,j*=2)
    s1+=i;
    s2+=j;
```

s1 = 10

s2 = 16 // อย่าลืม วงเล็บปีกกา ถ้าอยากให้ทำทั้งกลุ่ม


```
int s1=0,s2=0;
int i,j,k;
for(i=1,j=1; i<5 ;i++,j*=2) {
    s1+=i;
    s2+=j;
}
s1 = 10
s2 = 15
```

ตัวอย่างที่ 1 - square

input – N ($N > 2$)

output – สี่เหลี่ยมขนาด $N \times N$

$N=3$

* * *

* *

* * *

$N=4$

* * * *

* * *

* * *

* * * *

ตัวควบคุมวงวน

นอกจากการให้หลุดลูป โดยธรรมชาติ (เงื่อนไขเป็นเท็จ) ยังสามารถควบคุมด้วยวิธีคำสั่งอื่นได้ คือ

`continue;`

`break;`

continue

เจอคำสั่งนี้ **ขึ้นลูปใหม่** ทันที

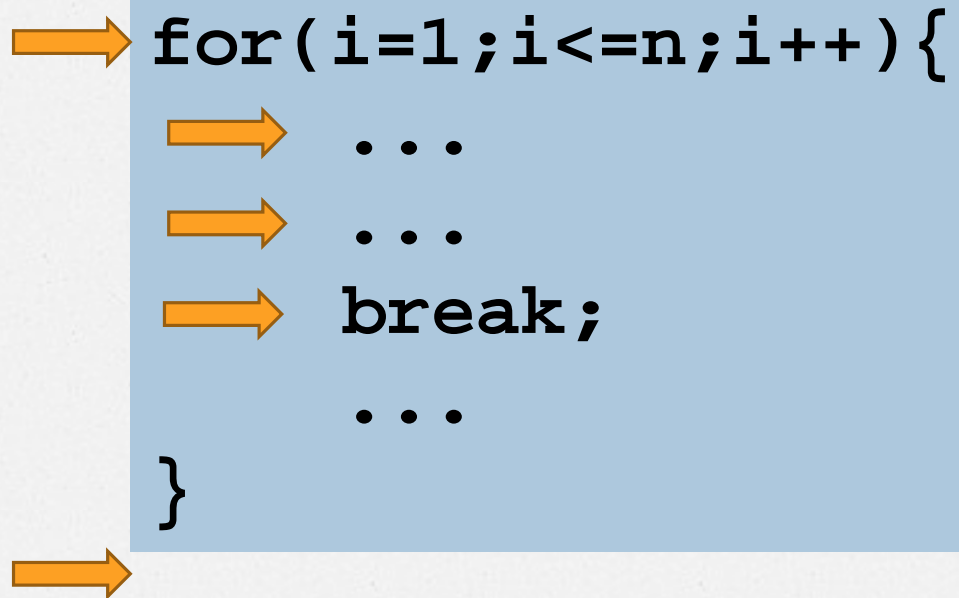
```
→ for( i=1; i<=n; i++ ) {  
    → ...  
    → ...  
    → continue;  
    ...  
}
```


ปัญหาที่ 1 - โปรแกรมปริศนา

```
int sum=0,i;  
for(i=1;i<=100;i++){  
    if(i%2)  
        continue;  
    else  
        sum+=i;  
}  
printf("%d\n",sum);
```

break

เจอคำสั่งนี้ **ออกจากรูป** ทันที



```
for( i=1; i<=n; i++ ) {  
    ...  
    ...  
    break;  
    ...  
}
```


ตัวอย่างที่ 2 - X

input – N ($N > 2$ and N is odd)

output – กราฟขนาด N

N=3

```
*  *  
  
 *  
  
*  *
```

N=5

```
*      *  
  
  *  *  
  
    *  
  
  *  *  
  
*      *
```

ตัวอย่างที่ 3 - diamond

input – N ($N > 2$ and N is odd)

output – สี่เหลี่ยมข้าวหลามตัดขนาด N

N=3

```
  *  
 * *  
  *
```

N=5

```
    *  
  * *  
 *   *  
*     *  
 *   *  
  * *  
    *
```


while loop

**while(เงื่อนไขคงการทำซ้ำ)
คำสั่งที่จะให้ทำซ้ำ**

การทำงานของ while loop

1. ตรวจสอบเงื่อนไข
2. ถ้าเป็นจริง → ทำตามคำสั่ง → ทำซ้ำ 2 ซ้ำ
ถ้าเป็นเท็จ → ออกจาก loop

จะใช้ while อย่าลืมเงื่อนไขเพิ่มค่า มิเช่นนั้นจะเกิด inf. loop

do-while loop

do

คำสั่งที่จะให้ทำซ้ำ

while(เงื่อนไขคงการทำซ้ำ) ;

การทำงานของ do-while loop

1. ทำคำสั่งก่อน
2. ตรวจสอบเงื่อนไข
3. ถ้าเป็นจริง → ทำข้อ 1 ซ้ำ
ถ้าเป็นเท็จ → ออกจาก loop

Loop แบบต่างๆ

```
int i,sum=0;  
for(i=1;i<=10;i++)  
    sum+=i;
```

```
int i=1,sum=0;  
do  
{  
    sum+=i;  
    i++;  
}  
while(i<=10);
```

```
int i=1,sum=0;  
while(i<=10)  
{  
    sum+=i;  
    i++;  
}
```

Loop อีกชนิด

```
int i=1,sum=0;  
for(i=1;i<=10;i++);  
    sum+=i;
```

```
for(;;);    while(1);
```

```
while(0) printf("x");
```

```
int i=1,sum=0;  
for(i=10;i;i--)  
    sum+=i;
```

```
do  
{  
    printf("x\n");  
}while(false);
```

```
int i=1;  
while(i<=5)  
    printf("x");  
    i++;
```

```
int i;  
for(i=0xABCD;i;i>>=1)  
    printf("x");
```


ตัวอย่างที่ 4 - max

หาค่ามากที่สุดใน อาร์เรย์

ตัวอย่าง

input	5
	9 2 12 4 6
output	12

ตัวอย่างที่ 5 – gen_id

สร้างเลขบัตรประจำตัวประชาชน ที่ถูกต้องตามกฎหมาย

```
k=0;  
for(i=0;i<12;i++)  
    k+=((str[i]-'0')*(13-i));  
k%=11;  
k=11-k;  
k%=10;  
printf("%s%d\n",str,k);
```


?? คำถาม ??

แบบฝึกหัด

- skgrader : แฟคทอเรียล (fac)
- skgrader : ผลบวกต่อเนื่อง (consum)
- skgrader : ต้นคริสต์มาสภาค 2 (chist2)
- skgrader : ผลต่างมากที่สุด (maxdif)
- skgrader : หารลงตัว (div)

จบสัปดาห์ที่ 3

สวัสดี