

สัปดาห์ที่ 1 – ปฐมบท

การเตรียมความพร้อมสู่ สอน. คอมพิวเตอร์ ค่าย 1 ระยะที่ 1

ห้องกุหลาบเพชร โรงเรียนสวนกุหลาบวิทยาลัย

ศิริระ ทรงพลโรจนกุล

วันนี้ มีอะไร

- ดูตัวอย่างต่างๆ ที่เกิดจากการเขียนโปรแกรม
- ทำไมต้องเรียนภาษาซี
- แนะนำโปรแกรมบรรณาธิการ
- รู้จักหน้าต่างของโปรแกรมภาษาซี
- การแสดงผล
- ตัวแปร
- การรับข้อมูล

ตัวอย่างโปรแกรมต่างๆ

ทำไมต้อง ภาษาซี

- เหมาะกับผู้เริ่มต้น
- เข้าใจง่าย
- ทำงานเร็ว
- เป็นแกนหลัก ของการเขียนโปรแกรม
- ใช้ใน สอน.

เตรียมตัวเขียนโปรแกรม

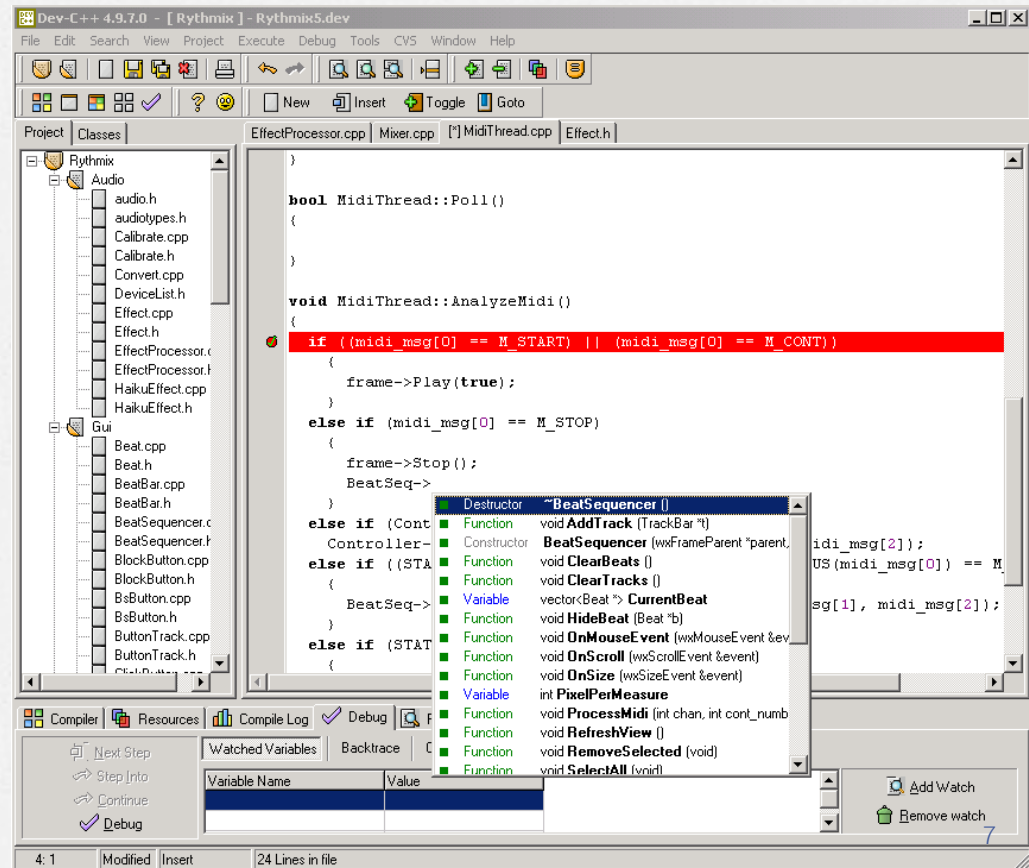
โปรแกรมบรรณาธิการ

- มีไว้เขียนโปรแกรม
- มีให้เลือกใช้มากมายหลายตัว ทั้งเสียเงิน และไม่เสียเงินขึ้นอยู่กับการใช้งาน
- โปรแกรมบรรณาธิการที่แนะนำ คือ

โปรแกรมบรรณาธิการ

DEV C++

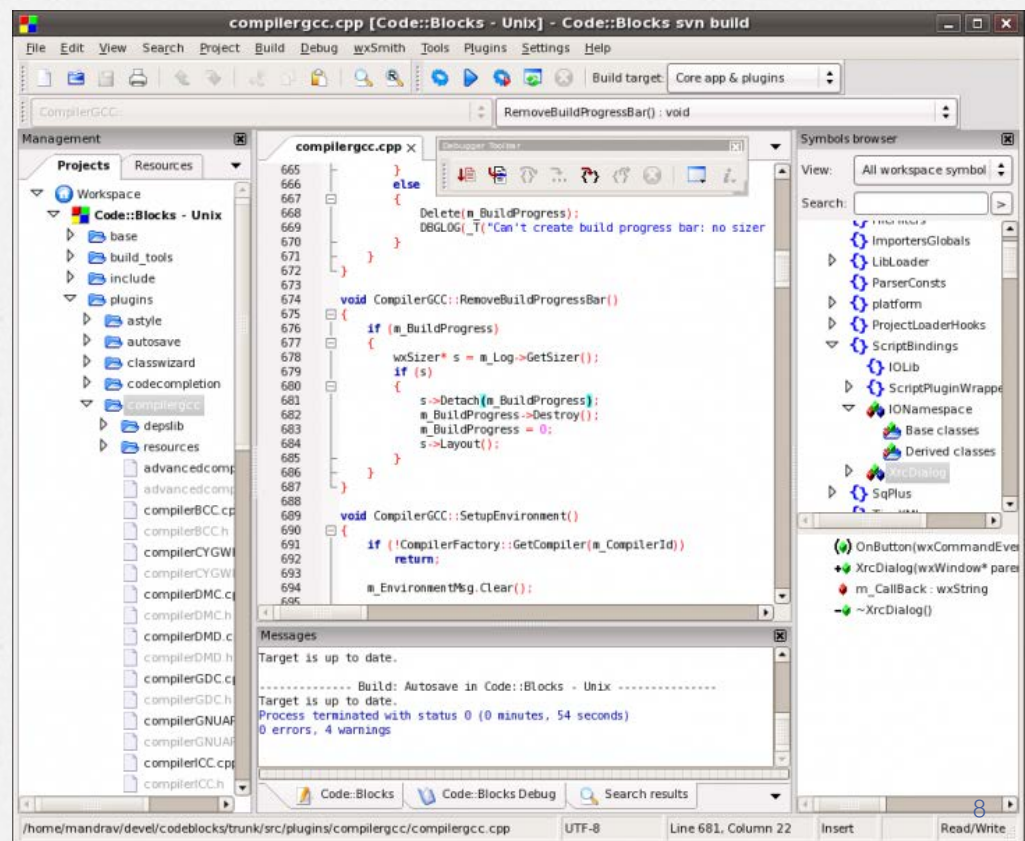
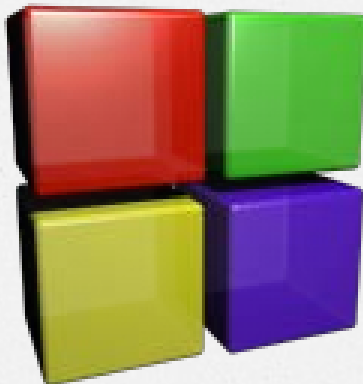
4.9.9.2 (Sep 2007)



โปรแกรมบรรณาธิการ

Codeblock

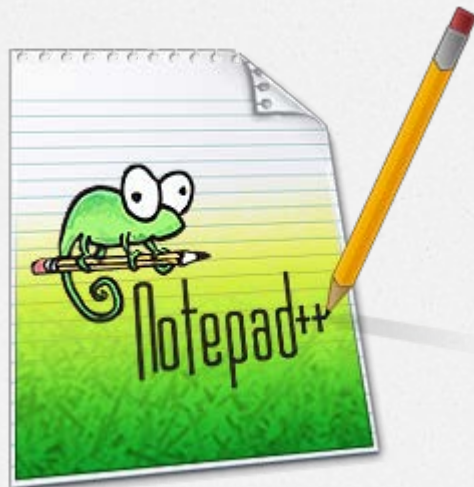
10.05 (May 2010)



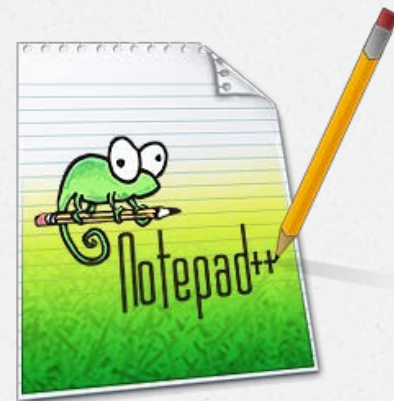
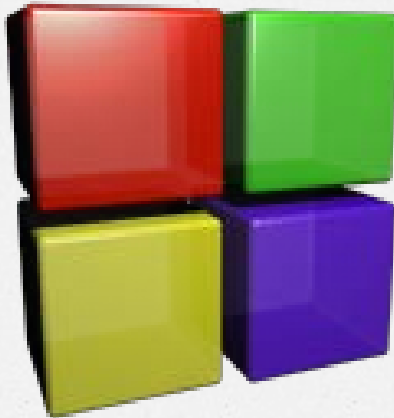
โปรแกรมบรรณาธิการ

Notepad++

6.1.3 (May 2012)

A screenshot of the Notepad++ application window. The title bar shows the file path: *D:\source\npp.5.9.2.src\notepad4ever.cpp - Notepad++. The window contains a C++ source file named notepad4ever.cpp. The code is as follows:

```
1  #include <GPL.h>
2  #include <free_software.h>
3
4  void notepad4ever ()
5  {
6      while (true)
7      {
8          Notepad++;
9      }
10 }
```



หน้าตาโปรแกรมภาษาซี

ส่วนต่างๆ ของโปรแกรม

```
1  /*
2  TASK: mod
3  LANG: C++
4  */
5  #include<stdio.h>
6  #include<stdlib.h>
7  bool remain[40];
8  int main()
9  {
10     int i,j,k;
11     for(i=1;i<=15;i++)
12     {
13         scanf("%d",&j);
14         remain[j%37]=1;
15     }
16     k=0;
17     for(i=0;i<37;i++)
18         k+=remain[i];
19     printf("1\n",k);
20     return 0;
21 }
22
```

ส่วนหัว

ส่วนตัว

ส่วนหัวของโปรแกรม

- ขึ้นด้วยตัว #
- ในขั้นนี้ ให้รู้จัก สองตัวคือ
 - #include<ชื่อเฮดเดอร์>
 - #define ตัวแปร ค่า

#include

- o ใช้ดึงแฟ้มเฮดเดอร์ ที่เก็บฟังก์ชันต่างๆ มาใช้งาน
- o เฮดเดอร์ที่สำคัญ ในการเขียนโปรแกรม คือ
 - o `stdio.h` → standard input / output
 - o `stdlib.h` → standard library
 - o `math.h` → math function
 - o `string.h` → string

#define

- ใช้กำหนด ค่าคงที่ต่างๆ เช่น
 - #define VAT 0.07
 - #define YEAR 2555
- ใช้กำหนด ฟังก์ชัน
 - จะกล่าวถึง ในโอกาสต่อไป

ส่วนตัวโปรแกรม

- เป็นส่วนที่สำคัญที่สุด ถ้าไม่มีโปรแกรมรันไม่ได้
- ประกอบด้วยหลายๆ ฟังก์ชัน เข้าด้วยกัน
- ฟังก์ชันที่สำคัญที่สุดคือ `int main()`

การเขียน

```
int main() {
```

เขียนโค้ดที่นี่



```
}
```


เริ่มต้นเขียนโปรแกรม

การแสดงผล

คำสั่งการแสดงผลพื้นฐาน

#include<stdio.h>

printf(“**สิ่งที่**จะแสดง”);

ต้องมีเครื่องหมาย วรรคตอน
semicolon

ตัวอย่างที่ 1 : SUANKULARB

พิมพ์คำว่า SUANKULARB ออกมาทางหน้าจอ

คำถาม ?

หากต้องการให้พิมพ์ เป็น

SUANKULARB

127

ต้องแก้ไขโค้ดในโปรแกรมอย่างไร

อักขระหลีก (Escape Characters)

- คือ อักขระ ที่ไม่สามารถแสดงผลได้ด้วยการพิมพ์ปกติ จะต้องมียุ่บว backslash ‘\’
- อักขระหลีก ที่ต้องรู้
 - \n → ขึ้นบรรทัดใหม่
 - \t → tab (เว้น 8 อักขระ)
 - \\, \', \” → พิมพ์ตัว \ ‘ “
 - \0 → NULL
 - \a → ส่งเสียง

ตัวอย่างที่ 2 : show that

ให้เขียนโปรแกรม ที่พิมพ์ข้อความ ดังนี้

'\n' is a New line character.

skgrader : star

พิมพ์

*

* *

* * *

ทางหน้าจอ

การรับข้อมูล

ประเภทของข้อมูล

ในภาษาซี ข้อมูล แบ่งออกหลักๆ เป็น 3 ประเภท คือ

- จำนวนเต็ม
- จำนวนจริง (ทศนิยม)
- อักขระ

สำหรับข้อมูลประเภทที่ต่างกัน ก็ต้องมีภาชนะจัดเก็บที่ต่างกัน เหมือนกับอาหาร กับ ภาชนะที่ใส่ โดยในภาษาซี เรียกภาชนะที่ใส่นี้ว่า **ตัวแปร**

จำนวนเต็ม

type	ขนาดตัวแปร	ช่วงการเก็บข้อมูล
short	16 bits	-32,768 ~ 32,767
int	32 bits	-2,147,483,648 ~ 2,147,483,647
long long	64 bits	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807

8 bits = 1 byte

จำนวนเต็ม : 2's complement

ตัวอย่าง จำนวนเต็ม 8 บิต

0	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

↑
sign bit

0 เป็น +

1 เป็น -

ตัวอย่างนี้ มีค่าเป็น

$$1 \times 2^0 + 1 \times 2^2 + 1 \times 2^4 = 21$$

จำนวนเต็ม : 2's complement

การเปลี่ยน sign bit เป็น 1 แบบนี้ ไม่ใช่เป็น -21

1	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

จำนวนเต็มลบ : 2's complement

ถ้าต้องการเปลี่ยนเลข **บวก** เป็น **ลบ** ให้ กลับบิตทุกตัว แล้ว บวก 1

0	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---



1	1	1	0	1	0	1	0
---	---	---	---	---	---	---	---



1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

2's complement

สังเกตว่า

- $-21 + 21 = 0$ (ใน 8 บิต ที่เราสนใจ เป็น 0 ทุกตัว)
- ถ้าใช้หลักคิดนี้ จะทำให้ $0 = -0$ (ลองคิดดู)
- จึงกำหนดให้ $10000000 = -2^n$ (n =จำนวนบิต)
- ดังนั้น ขอบเขตของค่า จะเป็น

จำนวนเต็ม

type	ขนาดตัวแปร	ช่วงการเก็บข้อมูล
short	16 bits	$-2^{15} \sim 2^{15}-1$ ประมาณ สามหมื่น
int	32 bits	$-2^{31} \sim 2^{31}-1$ ประมาณ สองพันล้าน
long long	64 bits	$-2^{63} \sim 2^{63}-1$ ประมาณ เลขสิบล้านล้าน

ถ้าเรารู้ว่า จะเก็บแต่ค่าบวก \rightarrow ไม่สนใจเครื่องหมาย = ค่าที่เก็บได้ก็เพิ่มขึ้น

จำนวนเต็ม

type	ขนาดตัวแปร	ช่วงการเก็บข้อมูล
unsigned short	16 bits	$0 \sim 2^{16}-1$
unsigned int	32 bits	$0 \sim 2^{32}-1$
unsigned long long	64 bits	$0 \sim 2^{64}-1$

จำนวนจริง

type	ขนาดตัวแปร	ช่วงการเก็บข้อมูล
float	32 bits	3.4E +/- 38 (7 digits)
double	64 bits	1.7E +/- 308 (15 digits)
long double	128 bits	

อักขระ

type	ขนาดตัวแปร	ช่วงการเก็บข้อมูล
char	8 bits	-128 ~ 127
unsigned char	8 bits	0 ~ 255

ในภาษาซี เก็บตัวอักขระ โดยใช้ ASCII code (1 byte)

จาวาใช้ UNI code (4 bytes)

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

ASCII code ที่ต้องจำ

0 → 48 / A → 65 / a → 97

การประกาศ ตัวแปร

ประเภท ชื่อตัวแปร;

ตัวอย่าง

```
int sk=127;
```

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

```
double r,pi=3.14;
```

```
double x;
```

$x = y$;

เป็นการเอา ค่าทางนิพจน์ทางขวา ของ = มาใส่ในตัวแปรทางซ้าย

การประกาศตัวแปร

ตัวแปร **char** มีการ กำหนดค่า ที่ต่างออกไป

สมมติ จะให้ **ch** เก็บ อักขระ **1** ต้องประกาศ

```
char ch='1';    // correct
```

ถ้าประกาศ

```
char ch=1;  // incorrect
```

การประกาศอย่างด้านล่าง ไม่ใช่ทำไม่ได้ แต่เพียงแค่ได้สิ่งที่ไม่ตรงกับที่เรา
อยากได้ เท่านั้นเอง

การประกาศตัวแปร

```
char ch1='A';  
char ch2=65;           // ch2='A'  
char ch3=ch2+32        // ch3='a'  
char ch4=ch1+32        // ch4='a'  
int x=ch1-'0'           // x=17  
int y='s'-'A'           // y=?  
char ch5=y              // ch5=?
```

int กับ **char** ก็คือจำนวนเต็ม ทั้งคู่ ต่างกันที่จุดประสงค์ที่ใช้

การประกาศตัวแปร

โดยปกติ เราก็ไม่ได้มีแค่ อักขระเดี่ยวๆ มันมาเป็นสายอักขระ (string)
การประกาศตัวแปร **char** เดี่ยวๆ หลายๆ ตัวมาเก็บ ก็ยุ่งยากเกิน
หากต้องการเก็บสายอักขระ ให้ประกาศ

char ชื่อตัวแปร[ขนาดของสตริง];

ตัวอย่าง

```
char str[]="SUANKULARB";  
char s[127];
```

สังเกตว่า สตริง ใช้ “ ”
อักขระ ใช้ ‘ ’

สายอักขระ

idx	0	1	2	3	4	5	6	7	8	9	10
str	S	U	A	N	K	U	L	A	R	B	\0

สังเกต

สตริง จะจบด้วย ‘\0’ ดังนั้น หากอยากเก็บสตริงยาว 10 ก็ต้องประกาศ มากกว่านั้นอย่างน้อย 1 (โดยทั่วไปแนะนำให้เผื่อไว้ 5) สตริง ก็คือ อักขระมาต่อๆ กัน สามารถเรียกทีละช่องได้ เช่น

str[0] , str[5]

สังเกตเห็นว่ามันเริ่มที่ ช่อง 0 ไม่ใช่ช่อง 1

คำสั่งการรับข้อมูล

```
#include<stdio.h>
```

```
scanf(“รูปแบบการสแกน”,&var1,&var2,...,&varN);
```


การรับข้อมูลเข้าตัวแปร

ต้องมี กลวิธี เฉพาะ ในการรับข้อมูลเข้าตัวแปร เหมือนประหนึ่งว่า จะใส่ข้อมูลลงภาชนะ อย่างไร ต้องใช้ให้ถูกแบบ

ตัวอย่าง 3 : $a+b$

รับจำนวนเต็มสองตัว a และ b

ให้หาค่าของ $a+b$

ตัวอย่าง รับเลข 2 3 ต้องแสดงผล 5 ออกมา

จำนวนเต็ม

type	format	หมายเหตุ
short	%hd	
int	%d	รับข้อมูลเป็นจำนวนเต็มฐานสิบ
long long	%I64d %lld	window linux

type	format	หมายเหตุ
unsigned short	%hu	
unsigned int	%u	
unsigned long long	%I64u %llu	window linux

จำนวนเต็ม

คอมพิวเตอร์ เก็บข้อมูลเป็น เลขฐานสอง

สังเกตว่า จำนวนเต็ม ใช้ %d → decimal (base10)

ต้องการรับข้อมูลเป็นเลข ฐานแปด

%o → octal (base8)

ฐานสิบหก (0123456789ABCDEF)

%x , %X → hexadecimal (base16)

จำนวนเต็ม

ถ้ามี 0 นำหน้า เป็นเลขฐาน 8

ถ้ามี 0x นำหน้า เป็นเลขฐาน 16

เช่น

15

015 \rightarrow 13

0x15 \rightarrow 21

เรามี %i เพื่อรองรับตัวเลขเหล่านี้ในการรับเข้า (ปกตินิยมใช้ %d มากกว่า)

จำนวนจริง

type	format	หมายเหตุ
float	%f	
double	%lf	
long double	%Lf	ใช้ไม่ได้ใน dev-c

จำนวนจริง

$$A \times 10^N, 0 \leq A < 10 \rightarrow A \in \mathbb{R}$$

$$1.7 \times 10^3 \rightarrow 1.7\text{E}3$$

ใช้ %e, %E, %g, %G ในการรับ

อักขระ

type	format	หมายเหตุ
char	%c	
char[]	%s	string

char สามารถใช้ tag ของจำนวนเต็มได้

รูปแบบทั่วไปของ format scanf

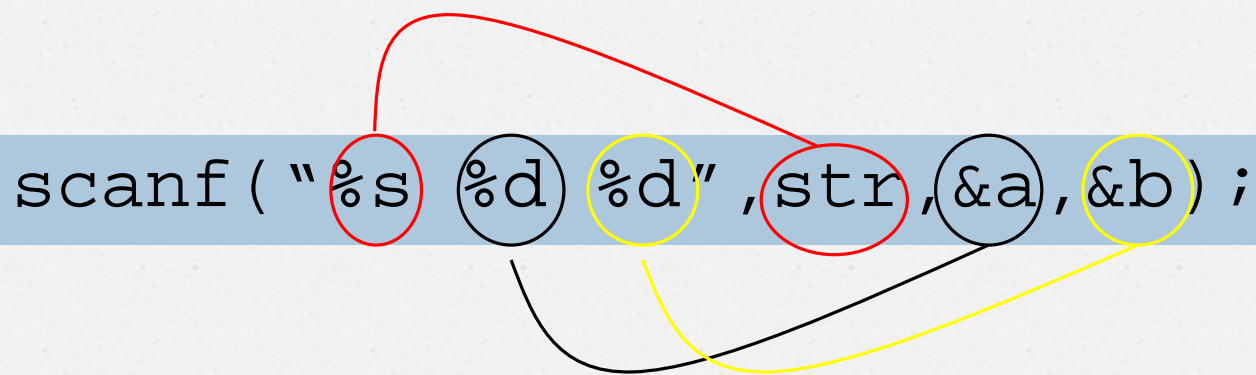
`%[*][width][modifiers]type`

[*] คือ รับมาแต่ไม่ได้เก็บเข้าตัวแปรใด

[width] คือ ขนาดของข้อมูล(จำนวนตัว) ที่รับ

[modifiers] คือ ขนาดของตัวแปร เช่น ปกติ สั้น(h) ยาว(l)

type คือ ประเภทของตัวแปร



```
scanf ( "%s %d %d" , str , &a , &b ) ;
```

การสแกน จะรับค่าเก็บเข้าตัวแปร เรียงตามลำดับเข้าไป

ค่าที่ใส่ จะต้องสอดคล้องกับ **format** ที่กำหนด
มิเช่นนั้นจะเกิดข้อผิดพลาดขึ้น

สังเกตว่า รับ **string** ไม่ต้องใส่ **&**
การรับตัวแปรอื่น ต้องใส่ **&**

skgrader : อ่าน เขียน

รับจำนวนเต็ม a b c

ให้พิมพ์ค่าของ

a b c

c b a

ออกมา

printf กับ format

format	type	ตัวอย่าง
%d	integer	127
%lf	double	3.1412
%c	char	A
%s	string	Suankularb
%u	unsigned	609

% อื่นๆ ที่ไม่ได้เขียน ที่ใช้กับ `scanf` ได้ ก็ใช้กับ `printf` ได้

รูปแบบทั่วไปของ `format printf`

`%[flags][width][.precision][modifiers]type`

[flags] คือ + - 0 # หรือ ไม่มี

[width] คือ ขนาดของข้อมูลที่จะพิมพ์

[.precision] คือ จำนวนตำแหน่งทศนิยม

[modifiers] คือ ขนาดของสิ่งพิมพ์ เช่น ปกติ สั้น(h) ยาว(l)

type คือ ประเภทที่จะพิมพ์

ตัวอย่าง

`%5d` → จอง 5 ช่อง พิมพ์เลขชิดขวา

`%05d` → ??

`%-5d` → ??

`%+5d` → ??

`%lf` → ??

`%.4lf` → ??

ทดลองดู ตามอัธยาศัย


```
printf ("%c %c\n", 'a', 65);  
printf ("%d %I64d\n", 2555, 123456760911LL);  
printf ("%10d\n", 2555);  
printf ("%010d\n", 2555);  
printf ("%d %x %o %#x %#ox\n", 100, 100, 100, 100,  
100);  
printf ("%4.2f %+0e %E \n", 3.1416, 3.1416, 3.1416);  
printf ("Width trick: %*d \n", 5, 10);  
printf ("%s_\n", "SUANKULARB");
```

a A

2555 123456760911

2555

0000002555

100 64 144 0x64 0144x

3.14 +3e+000 3.141600E+000

Width trick: 10

SUANKULARB_

ตัวอย่าง 4 : $a+b$ อีกครั้ง

รับ จำนวนเต็ม สองตัว a กับ b

แต่ a เป็นเลขฐานแปด b เป็นเลขฐานสิบ

ให้หา $a+b$ ฐานสิบหก ว่าได้เท่าใด

ตัวอย่าง

INPUT : 15 15

OUTPUT: 1C

skgrader : chocolate

จำนวนเต็มสี่ตัว a_1, b_1, a_2, b_2 แต่ละตัวคั่นด้วยช่องว่างหนึ่งช่อง ($1 \leq a_1, b_1, a_2, b_2 \leq 1000$) โดยที่ a_i แสดงขนาดบรรจุและ b_i ระบุจำนวนกล่องที่มีสำหรับขนาดบรรจุ a_i

หาจำนวนช็อกโกแลตที่มี

?? คำถาม ??

แบบฝึกหัด

- o skgrader : ตัณคริสตมาส
- o skgrader : กล่อง
- o skgrader : วงกลม (ใช้ `M_PI` แทนค่า `pi` ใน header `math.h`)

จบสัปดาห์ที่ 1

สวัสดี