



# Assignment 1 (input/output)

## ✚ Assignment 1 ทดสอบคำสั่งรับข้อมูล / แสดงผล

- ให้นักเขียนโปรแกรมเพื่อทดสอบความเข้าใจการใช้คำสั่งในการจัดการตัวเลข และ ตัวอักษร
  - พิจารณาตัวอย่างโปรแกรมแล้วดัดแปลงเขียนใหม่ให้สามารถทดสอบความเข้าใจของตนเอง
  - ให้มีการเขียนวนรอบในการรับคำสั่งจนกว่าจะเลิกทำ เพื่อทดสอบความผิดพลาดที่อาจทำให้โปรแกรมหยุดทำงานด้วย

## ✚ ผลลัพธ์การเรียนรู้

- เข้าใจความถูกต้องในการใช้คำสั่งเกี่ยวกับการรับข้อมูลและแสดงผลของตัวแปรประเภทต่างๆ

## ✚ หัวข้อที่ต้องทดสอบ

- ความถูกต้องในใช้คำสั่งรับข้อมูล(scanf)/แสดงผล(printf) กับตัวแปรประเภทต่างๆ
- การใช้ตัวควบคุมการรับข้อมูล/แสดงผล ถ้าใช้ผิดมีผลหรือไม่อย่างไร
  - เช่น การใช้ %f กับตัวแปรจำนวนเต็ม การใช้ %d กับตัวแปรจำนวนจริง การใช้ %s เป็นต้น
- การป้อนข้อมูลเกินค่าที่ตัวแปรรับได้ เช่น
  - ป้อนตัวเลขเกินค่าที่รับได้ / ป้อนตัวอักษรเกินความยาวที่จองไว้
- การป้อนค่าผิดรูปแบบ เช่น ป้อนตัวอักษร ในคำสั่งที่รับตัวเลข

## ✚ Assignment รายงานที่ต้องส่ง

- ไฟล์แนบที่ 1 ตัวรายงานรวมแปลงเป็นฟอร์แมต pdf ประกอบด้วย source code ที่ออกแบบ หน้าจอผลการทดลอง (เลือกเฉพาะอันที่ต้องการอธิบาย) คำอธิบายผลของแต่ละหน้าจอ สรุปผลการทดลอง
- ไฟล์แนบที่ 2 เฉพาะ source code (ไฟล์นามสกุล.C) ที่ทดลองเขียน

# Assignment 1 (input/output)

เติมเครื่องหมาย ✓ ในช่องที่คำสั่งใช้งานได้

scanf	%d	%llu	%f	%lf	%Lf	%e	%c	%s
unsigned int	✓	✓		×	×	×		
int	✓				×	×		
long	✓				×	×		
long long		✓				×		
float			✓			×		
double				✓		×		
extend					✓	×		
char	✓					×	✓	
string						×		✓

printf	%d	%llu	%f	%lf	%Lf	%e	%c	%s
unsigned int	✓						✓	
int	✓						✓	
long	✓						✓	
long long	✓						✓	
float			✓	✓		✓		
double			✓	✓		✓		
extend			✓	✓	✓	✓		
char	✓	✓					✓	
string								✓

# Assignment 1 (input/output)

- คำสั่งรับข้อมูล scanf
- คำสั่งแสดงผล printf
- ชนิดของตัวแปร และตัวควบคุม(format specifier)
  - ตัวเลขจำนวนเต็ม
    - %d** **int** (integer) 16 bits ( $-2^{15} .. 2^{15}-1 \Rightarrow -32768 .. 32767$ ) ปัจจุบันเปลี่ยนเป็น 32 bits
    - %d** **long** / long int (long integer) 32 bits ( $-2^{31} .. 2^{31}-1$ )
    - %llu** **long long** (64 bits integer) 64 bits ( $-2^{63} .. 2^{63}-1$ )
  - ตัวเลขจำนวนเต็มแบบไม่มีบิตเครื่องหมาย (8/16/32/64 bits) และตัวควบคุม %d, %ld, %llu
    - %d** **char** 8 bits (0..255) สามารถใช้ร่วมกับตัวอักษร ASCII ได้
    - %d** **unsigned int** จำนวนเต็มไม่คิดบิตเครื่องหมาย(-)ขนาด 16 bits (0 .. 65535)
  - ตัวเลขจำนวนจริง (32/64/128 bits IEEE-754) และตัวควบคุม %f, %lf, %Lf, %e
    - %f** **float** 32 bits single-precision
    - %lf** **double** 64 bits double-precision
    - %Lf** **long double** 128 bits extend-precision
    - %e** ใช้เฉพาะการแสดงผลเลขจำนวนจริงในรูปแบบของ scientific notation  $aeb \Rightarrow a \times 10^b$
  - ตัวแปรเก็บตัวอักษร(ASCII)
    - %c** **char** 8-bits ASCII character
    - %s** **char []** Array of character / string

# ตัวอย่างโปรแกรม Assign1.C

เปิดโปรแกรมด้วย CodeBlock (codeblocks-20.03mingw-setup.exe) หรือใช้ตัวที่ถนัด

- File -> New -> Source File
- พิมพ์โปรแกรมตามที่กำหนด

<pre> 1 #include &lt;stdio.h&gt; 2 int main() 3 { float a; 4   char ch; 5   do { 6     a = 0; 7     printf("\nAssignment 1 Program Test specific format of variable\n"); 8     printf("Enter value of a : "); 9     scanf("%f",&amp;a); 10    printf("print format of a\n"); 11    printf("int = %d\nfloat = %f\ndouble = %lf\nexponent = %e\n", a, a, a, a); 12    printf("Enter y to run again or n to exit."); 13    scan(" %c",&amp;ch); 14  } while (ch == 'y'); 15  printf("\nEnd Program\n"); 16  return 0; 17 }</pre>	<p>กรณี จองตัวแปรเป็นจำนวนจริง double แต่ทดลองอ่านด้วย %f ซึ่งสรุปได้ว่า ไม่สามารถใช้ %f กับตัวแปรที่เป็น double ได้</p> <p>ทดสอบการแสดงผลตัวแปรด้วย format ต่างๆกัน \n คือการขึ้นบรรทัดใหม่</p> <p>มีเว้นวรรคหน้า " %c"</p>	<p>ค่า a ที่แสดงผลด้วย %lf</p> <p>ทดสอบโปรแกรมโดยป้อนเลข 1 Enter value of a : 1Enter]</p> <p>int = 0 float = 0.000000 double = 0.000000 exponent = 1.000000e+000</p> <p>Enter y to run again or n to exit. □</p> <p>ค่า a ที่แสดงด้วย %e</p> <p>หยุดรอให้ป้อน y เพื่อทำงานต่อ จากคำสั่ง scan(" %c",&amp;ch);</p>
---	--	--

- Save ไฟล์ตั้งชื่อ โดยใช้นามสกุล .c
- Run โปรแกรมโดยใช้ Build & Run

ถ้าใช้ a เป็น int และใช้ %d (บรรทัดที่ 9)  
Enter value of a : 1Enter]

int = 1  
float = 0.000000  
double = 0.000000  
exponent = 4.940656e-324  
Enter y to run again or n to exit. □

ถ้าใช้ a เป็น char และใช้ %d (บรรทัดที่ 9)  
Enter value of a : 258Enter]

int = 2  
float = 0.000000  
double = 0.000000  
exponent = 9.881313e-324  
Enter y to run again or n to exit. □

โปรแกรมทันทีโดยไม่หยุดรอ

ถ้าใช้ a เป็น char และใช้ %d (บรรทัดที่ 9)  
Enter value of a : AEnter]

int = 0  
float = 0.000000  
double = 0.000000  
exponent = 0.000000e-000  
Enter y to run again or n to exit.  
End Program