

# C Programming Part 2 - 1

By Tanaroeg O-Charoen

## Output

Output ในภาษา C นั้นมีหลายฟังก์ชัน และหลาย Stream ซึ่ง Stream ที่มากับภาษา C นั้นจะมี อยู่ 3 Stream นั่นก็คือ stdin stdout stderr ซึ่งการ output จะมีที่ใช้หลักๆก็คือ stdout และใช้ stderr ในการแสดง error ออกมา

เรามาเริ่มรู้จักฟังก์ชันในการปรั้นกันก่อน

### printf()

```
printf(const char *format, ...)
```

ฟังก์ชันนี้เป็นฟังก์ชันในการปรั้นแบบมีการ Format โดยจะใช้  `%[width][flags].[precistion length]`

โดยจะมี Flag สำหรับการปรั้นของแต่ละ Data type หลักๆอยู่ดังนี้

Type	Flags
Character	c
Signed Integer	d, i
Floating point	f
String of chaacters	s
Unsigned integer	u
Pointer	p

เพิ่มเติม ใส่  `0` ไว้ข้างหน้าเพื่อ width เพื่อให้ใส่ 0 ตรงพื้นที่ที่ว่างอยู่ (Zero fill)  
ใส่  `-` ไว้ข้างหน้าเพื่อ width เพื่อให้ตัวอักษรหรือตัวเลขมาชิดซ้ายแทน  
ใส่  `L` สำหรับปรั้นตัวเลข จำนวนเต็มที่มี  `long` นำหน้า ใส่  `L` สำหรับปรั้นตัวเลข ทศนิยมที่มี  `long` นำหน้า

### putchar()

`putchar(char c)` เป็นฟังก์ชันที่ใช้สำหรับปรั้นตัวอักษรที่เป็นประเภท character

### puts()

`puts(char *s)` เป็นฟังก์ชันที่ใช้ในการปรั้นข้อมูลประเภท String of character โดยฟังก์ชันนี้จะมีการ newline เมื่อพิมพ์เสร็จ

## Input

Input ของภาษา C นั้นจะรับผ่าน Stream ที่ชื่อว่า stdin ซึ่งในการรับข้อมูลก็จะมีฟังก์ชันในการรับข้อมูลดังนี้

### getchar()

`int getchar(void)` เป็นฟังก์ชันที่ใช้สำหรับการรับข้อมูลที่เป็นประเภท char จำนวน 1 ตัวอักษร โดยที่จะ return ascii code ของตัวอักษร ที่เราพิมพ์ลงไป ซึ่งเรามักจะใช้ฟังก์ชันนี้กับ  `putchar()`

```
#include <stdio.h>

int main() {
    char c = getchar();
    putchar(c);
}
```

### gets()

```
char* gets(char *str)
```

เป็นฟังก์ชันในการรับค่า string จากคีย์บอร์ดโดยจะรับจนกว่าจะเจอ  `newline ( Enter )` แล้วนำไปเก็บที่ตัวแปรใน parameter และ return ออกมา

```
#include <stdio.h>

int main() {
    char name[10];
    gets(name);
    puts(name);
}
```

**ข้อควรระวัง** หากใส่ข้อมูลเกินขนาดของ ตัวแปรที่เรากำหนดไว้จะทำให้เกิด Error ได้ เป็นฟังก์ชันที่ไม่มีความปลอดภัย

### fgets()

```
char *fgets(char *str, int n, FILE *stream)
```

เป็นฟังก์ชันที่รับค่าไปจนถึง  `newline` หรือ  `EOF (End-of-file)` คล้ายๆกับ  `gets()` แต่เราสามารถกำหนดขนาดของ input ได้และสามารถประยุกต์มาใช้กับไฟล์ได้ด้วยเช่นกันเพราะมี parameter ของ stream โดยในกรณีนี้เราจะใช้ stream เป็น stdin

```
#include <stdio.h>

int main() {
    char name[10];
    fgets(name, sizeof(name) -1, stdin); // ที่ลบ 1 ตรง sizeof เพราะเหลือที่ไว้ให้ null character
    puts(name);
}
```

ถ้าหากเราทดลอง ใส่ขนาดให้เกิน ขนาดของ input ในฟังก์ชันก็จะทำการตัด string ให้เท่ากับขนาดที่เรากำหนดไว้

### scanf()

```
int scanf(const char *format, ...)
```

`scanf()` นั้นจะเป็นฟังก์ชันที่ใช้สำหรับรับค่าของ input จากคีย์บอร์ด โดยที่เพิ่มเติมมาจากฟังก์ชันอื่นคือสามารถรับแบบ format คล้ายๆกับ  `printf()` แต่  `scanf()` นั้นจะรับไปจนถึง  `new line(enter)` หรือ  `tab` หรือ  `space` หรือ  `EOF(End-of-file)` และหลัง  `,` นั้น เราจะใส่ค่าเป็น address ของตัวแปร การรับข้อมูลที่ดีนั้นควรกำหนดขนาดของข้อมูลใน format ด้วย

```
#include <stdio.h>

int main() {
    char name[11];
    printf("Enter youname (10): ");
    scanf("%10s", name);

    printf("Your name: %s\n", name);
}
```

```
#include <stdio.h>

int main() {
    int x, y;
    // กรณีต้องการรับคู่อันดับ (x, y)
    printf("Enter (x, y): ");
    scanf("%d, %d", &x, &y);

    printf("x: %d\n", x);
    printf("y: %d\n", y);
}
```