



#2.5 Input/Output

Output

Output ในภาษา C นั้นมีหลายฟังก์ชัน และหลาย Stream ซึ่ง Stream ที่มากับภาษา C นั้นจะมีอยู่ 3 Stream นั่นก็คือ stdin stdout stderr ซึ่งเราสามารถใช้ stdout และใช้ stderr ในการแสดง output ได้

**Stream คือสิ่งที่จะทำให้เรา interact กับ environment ได้ เช่น I/O Device, Keyboard เราสามารถรู้จักฟังก์ชันใน output กัน!

printf()

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

ฟังก์ชันนี้เป็นฟังก์ชันในการปรับแบบมีการ Format โดยจะใช้ `%[width].[precision]length[flags]`

โดยจะมี Flag สำหรับการปรับของแต่ละ Data type หลักๆอยู่ดังนี้

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

เพิ่มเติม

ใส่ `0` ไว้ข้างหน้าเพื่อ width เพื่อให้ใส่ 0 ตรงพื้นที่ที่ว่างอยู่ (Zero padding)

ใส่ `-` ไว้ข้างหน้าเพื่อ width เพื่อให้ตัวอักษรหรือตัวเลขมาชิดซ้ายแทน

ใส่ `l` สำหรับปρί้นตัวเลข จำนวนเต็มที่มี `long` นำหน้า

ใส่ `L` สำหรับปρί้นตัวเลข ทศนิยมที่มี `long` นำหน้า

putchar()

`putchar(char c)` เป็นฟังก์ชันที่ใช้สำหรับป้อนตัวอักษรที่เป็นประเภท character สู่ stdout

puts()

`puts(char *s)` เป็นฟังก์ชันที่ใช้ในการป้อนข้อมูลประเภท String of character โดยฟังก์ชันนี้ จะมีการ 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

```
#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]; printf("Enter your name:"); fgets(name, sizeof(name) -1, stdin); // ที่ลบ 1 ตรง sizeof เพราะเหลือที่ไว้ให้ null character puts(name); }
```

scanf()

อ่านข้อมูลจาก stdin และใช้ format specifier แบบเดียวกับ printf

```
#include <stdio.h> int main() { int num; printf("Enter an integer: "); scanf("%d", &num); printf("You entered: %d\n", num); return 0; }
```