

ปฏิบัติการครั้งที่ 3

ฟังก์ชันและการส่งผ่านค่า



ฟังก์ชันมาตรฐาน (Library Functions)

▶ โปรโตไทป์ของฟังก์ชัน

- ประกาศไว้หน้แฟ้มข้อมูลที่มีนามสกุล เป็น .h

`int getchar (void) ;`

- เรียกการประกาศนี้โดยใส่

`#include <ชื่อแฟ้ม.h>`

`#include <stdio.h>`

- ▶ การเรียกใช้ฟังก์ชัน : ใช้กับรูปแบบโปรโตไทป์ของฟังก์ชันนั้นๆ เช่น

`ch = getchar () ;`

- ▶ การกำหนดฟังก์ชัน : มีพร้อมในตัวโปรแกรม

ฟังก์ชันมาตรฐานสำหรับการประมวลผลอักขระ

▶ ฟังก์ชันสำหรับรับอักขระ

- `int getchar(void); /*read a character from stdin*/`

▶ ฟังก์ชันสำหรับส่งอักขระ

- `int putchar(int); /* write a character to stdout */`

- ▶ ฟังก์ชัน/แมคโคร ใน `ctype.h` : ฟังก์ชันหรือแมคโครที่ช่วยอำนวยความสะดวกในการทำงานโปรแกรม เช่น ฟังก์ชันตรวจสอบว่าตัวอักขระนั้นเป็นตัวพิมพ์ใหญ่หรือไม่

การใช้ฟังก์ชันมาตรฐานรับอักขระ `getchar()`

- ▶ เพื่อทำการอ่านทีละตัวอักขระ จากบัฟเฟอร์ของแป้นพิมพ์ (Keyboard buffer) โดยมีการส่งต่อให้ทีละตัวอักขระทีละอัน

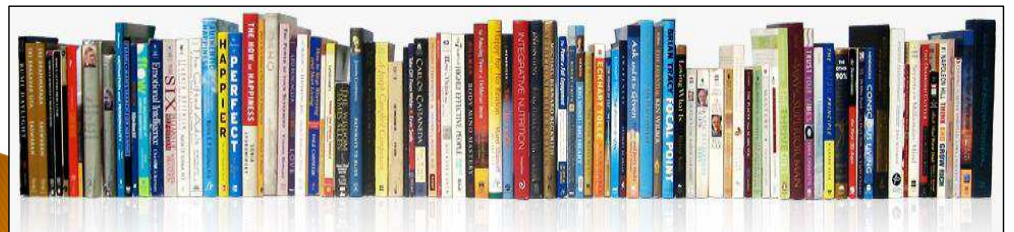
- ▶ ลักษณะการทำงานที่ปรากฏเสมือนเป็นการรับข้อความที่ส่งบรรทัด

- ▶ Include file : `<stdio.h>`

- ▶ Prototype : `int getchar (void) ;`

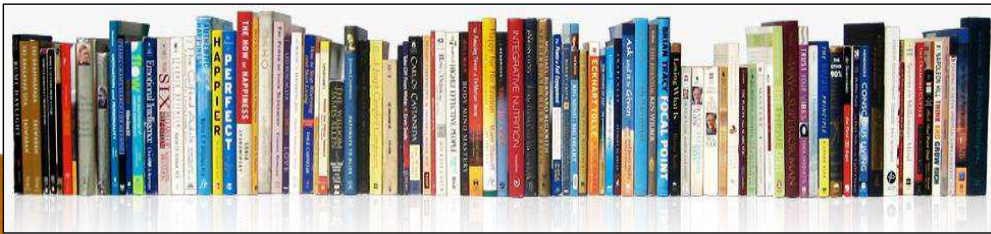
- ▶ Argument : ไม่มี หรือไม่ต้องระบุ

- ▶ Return : อักขระที่อ่านได้ ถ้าอ่านไม่ได้ จะให้ค่าเป็น EOF(-1)



การใช้ฟังก์ชันมาตรฐานสำหรับอักขระ putchar()

- ▶ เพื่อทำการแสดงอักขระการอุปการณีสถาปนศาสตร์
- ▶ Include file : <stdio.h>
- ▶ Prototype : int putchar (int);
- ▶ Argument : รหัส ASCII ของอักขระที่ต้องการแสดง
- ▶ Return : อักขระที่แสดงหรือปรากฏ
ในกรณีที่แสดงไม่ได้ จะให้ป้อนค่า EOF



ตัวอย่างการใช้ฟังก์ชันสำหรับการประมวลผลอักขระ

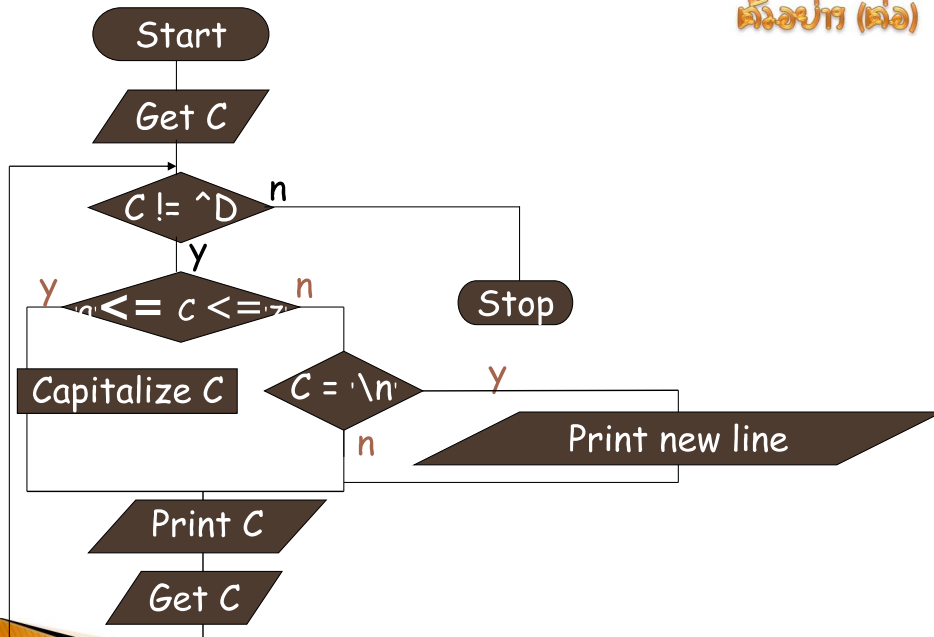
จงเขียนโปรแกรมเพื่อรับอักขระ แล้วแสดงผลของอักขระที่รับ โดยแสดงเป็นตัวพิมพ์ใหญ่ทั้งหมด พร้อมทั้งแทรกบรรทัดว่าง หากเจอการขึ้นบรรทัดใหม่ หยุดการป้อนข้อมูลด้วยการกด Control-D

เช่น รับข้อมูลเป็น I'll be a champion! <กด enter>
ผลลัพธ์ได้เป็น I'LL BE A CHAMPION!

<เว้น 1 บรรทัด>

<cursor รอรับการป้อนข้อมูล>

ตัวอย่าง (ต่อ)



ตัวอย่างการใช้ฟังก์ชันสำหรับการประมวลผลอักขระ

```

#include <stdio.h>
#define ControlD -1
#define NEWLINE '\n'
int main ( ) {
    int c;

    printf("Please enter text, press Ctrl+d to stop..\n");
    /* get characters, stop at end of file */
    while ((c = getchar()) != ControlD) {
        if ('a' <= c && c <= 'z')
            c = c + 'A' - 'a'; /*Capitalize the character c */
        else
            if (c == NEWLINE)
                putchar(NEWLINE); /*write the newline character*/
            putchar(c); /* output the character*/
    } /* End of while loop */
    return 0;
}
    
```

The macro in ctype.h

Macro

isalpha(c)
isupper(c)
islower(c)
isdigit(c)
isalnum(c)
isxdigit(c)
isspace(c)
ispunct(c)
iscntrl(c)
isascii(c)

Nonzero is returned if :
c is a letter
c is an uppercase letter
c is a lowercase letter
c is a digit
c is a letter or digit
c is a hexadecimal digit
c is a white space character
c is a punctuation character
c is a control character
c is an ASCII code

Function / Macro effect

toupper(c) *change c to uppercase*
tolower(c) *change c to lowercase*
toascii(c) *change c to ASCII code*

ตัวอย่างการใช้ฟังก์ชันสำหรับการประมวลผลอักขระ

```
#include <stdio.h>
#include <ctype.h>
#define ControlD -1
#define NEWLINE '\n'
int main ( ) {
    int c;
    while ((c = getchar()) != ControlD) {
        if (islower(c))
            c = toupper(c); /*Capitalize the character c */
        else
            if (c == NEWLINE)
                putchar(NEWLINE); /*write the newline character*/
            putchar(c); /* output the character*/
    }
    return 0;
}
```

ฟังก์ชันที่พัฒนาโดยโปรแกรมเมอร์

- ▶ การกำหนดฟังก์ชัน (Function definition)
- ▶ การเรียกใช้ฟังก์ชัน (Function invocation)
- ▶ การประกาศโครงสร้างหรือรูปแบบการเรียกใช้ฟังก์ชัน (Function prototype)

ลำดับการวาง

1. Function prototype

2. Function invocation

3. Function definition

1. Function definition

2. Function invocation

การออกแบบบัพฟังก์ชัน



- ▶ ส่วนต่อประสานของฟังก์ชัน →

- ส่วนหัวของฟังก์ชัน
- ประโยคโหม่งของฟังก์ชัน



การกำหนดฟังก์ชัน

Function header

ชนิดข้อมูลที่ส่งกลับ ชื่อฟังก์ชัน (การประกาศ พารามิเตอร์)

```
{  
    การประกาศตัวแปรภายในฟังก์ชัน;  
    คำสั่ง ;  
    return ค่าที่ต้องการส่งกลับ ;
```

Function body

```
}    int min (int x, int y) {  
        return (x<y? x : y);  
    }
```