

# 字元輸入與輸出

# getchar ( )      putchar ( )

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

```
int main(void)
```

```
{
```

```
    char ch;
```

```
    while ((ch = getchar()) != '@')
```

```
        putchar(ch);
```

```
    return 0;
```

```
}
```

範例 E08\_01.c



Hi, there. My email address



Hi, there. My email address

is mac@apple.com.

is mac



# getchar ( )      putchar ( )

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

```
int main(void)
```

```
{
```

```
    char ch;
```

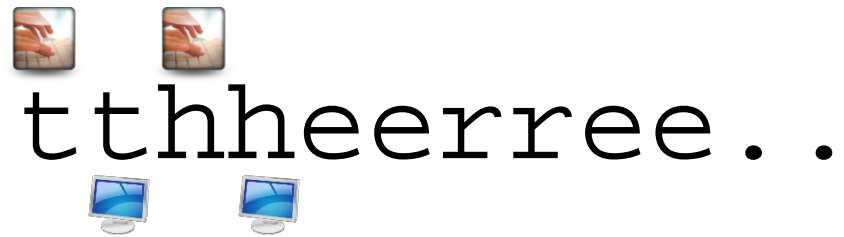
```
    while ((ch = getchar()) != '@')
```

```
        putchar(ch);
```

```
    return 0;
```

```
}
```

範例 E08\_01.c



# 緩衝區 (buffer)

- 鍵盤輸入的字元先被直接回應到螢幕上
  - (系統為了要讓我們知道究竟打了哪些字)
- 同時這些輸入的字元會被存在 buffer 中，等到使用者打了 Enter 才把 buffer 裡的字元送到程式

# 範例: 加密訊息

```
#include <stdio.h>
#define SPACE ' '
int main(void)
{
    char ch;
    ch = getchar();          /* read a character          */
    while (ch != '\n') {    /* while not end of line   */
        if (ch == SPACE)   /* leave the space         */
            putchar(ch);   /* character unchanged     */
        else
            putchar(ch + 1); /* change other characters */
        ch = getchar();     /* get next character       */
    }
    putchar(ch);           /* print the newline       */
    return 0;
}
```

輸出：

```
CALL ME HAL.
DBMM NF IBM/
```

# 跟字元處理有關的函數

<code>isalpha()</code>	判斷字元是否是英文字母
<code>islower()</code>	判斷字元是否是小寫
<code>isupper()</code>	判斷字元是否是大寫
<code>toupper()</code>	小寫轉大寫
<code>tolower()</code>	大寫轉小寫

# 大小寫互換

```
#include <stdio.h>
#include <ctype.h>
int main(void)
{
    char ch;
    while ((ch = getchar()) != '\n') {
        if (isalpha(ch)) /* if a letter, */
            putchar(ch + 1); /* change */
        else /* otherwise, */
            putchar(ch); /* print as is */
    }
    putchar(ch); /* print the newline */
    return 0;
}
```

# 檔案串流 stdin 和 stdout

- 對電腦來說，輸入輸出都是在對檔案做讀寫
- 作業系統角度
  - 把鍵盤的輸入寫到stdin
  - 把stdout的內容顯示在螢幕
- 程式的觀點
  - 從stdin這個檔案讀取資料(使用者輸入的資料)
  - 把printf之類的結果寫到stdout
- 檔案結束符號
  - EOF (end of file)
  - Ctrl+Z 或 Ctrl+D



# 重新導向

```
D:\code\> reflect.exe < reflect.c
```

```
D:\code\> reflect.exe > ten_words.txt
```

```
D:\code\> reflect.exe < reflect.c > reflect_copy.c
```

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

```
int main(void)
```

```
{
```

```
    int ch;
```

```
    while ((ch = getchar()) != EOF)
```

```
        putchar(ch);
```

```
    return 0;
```

```
}
```

# 處理 Buffered Input

```
while (getchar() != 'y')  
    printf("Well, then, is it %d?\n", ++guess);  
    printf("I knew I could do it!\n");
```

`\n', '\n'`

```
while (getchar() != 'y') {  
    printf("Well, then, is it %d?\n", ++guess);  
    while (getchar() != '\n')  
        continue;  
}
```