

## Bonus: # and ## Operators in C

```
#include <stdio.h>
#define mkstr(s) #s
int main(void)
{
    printf(mkstr(geeksforgeeks));
    return 0;
}
```

"#"名稱為stringize operator, 可以在marco中將後面接著的文字用雙引號框起來

例如#Welcome

相當於 "Welcome"

```
#include <stdio.h>
#define concat(a, b) a##b
int main(void)
{
    int xy = 30;
    printf("%d", concat(x, y));
    return 0;
}
```

"##"稱作Token-pasting operator, 可以在marco中將兩個變數名字連在一起

## 範例 1

有時候我們會有變數名稱很類似的東西

像是token1,token2,token3 ...

如果使用這兩個operators, 可以簡化輸出類似變數的步驟

```
#include <stdio.h>
#define paster( n ) printf("token" #n " = %d\n", token##n)
int main()
{
    int token1 = 5, token2 = 3;
    paster(1);
    paster(2);
}
```

用 #n 將帶入值轉成字串，再用token##n

把帶入值加上token前綴變成我們的變數

就可以印出我們要的變數

輸出結果如下圖

```
lapor@DESKTOP-7NVPA71:/mnt/e/CSIE-Programming-II/thinking$ ./A
token1 = 5
token2 = 3
```

## 範例 2

Debug時很多人喜歡拿printf去看變數的值

有時候我們要印很多變數時，會覺得這很煩

printf("var1 = %d,var2 = %d\n",var1,var2);

```
#include <stdio.h>
#define debug(n) printf(#n "= %d\n", n)
int main()
{
    int var1 = 32, anotherVar = -100000009;
    debug(var1);
    debug(anotherVar);
}
```

類似上面的範例 1，我們將代入的變數名稱轉成字串

再跟值一起輸出，就可以達到只要打一次變數就可以印出變數名跟值的效果了