

常用 printf() 和 scanf() 格式總整理

printf("%d", n);

FORMAT	ARGUMENT TYPE	
%d, %i	int	十進位表示法
%u	unsigned int	
%x	unsigned int	十六進位表示法
%#x	unsigned int	十六進位表示法 數字前面會自動加上 0x
%f	double	(傳給 printf() 的參數型別如果是 float，其實會直接被升級成 double)
%Lf	long double	
%e, %E	double	科學記號表示法
%c	int	(顯示出對應的字元, 傳參數的時候 char 會先被轉成 int)
%s	char *	用來顯示字串
%p	void *	用來顯示指標
%g, %G	double	選擇 %f 或 %e 兩者之中較簡短的表示法

scanf("%d", &n);

FORMAT	ARGUMENT TYPE	
%d	int *	傳整數變數的位址 例如 &n 會將讀取到的整數值存入 n
%ld	long *	
%lld	long long *	
%u	unsigned int *	
%f	float *	讀取 float
%lf	double *	讀取 double
%Lf	long double *	讀取 long double
%c	char *	讀取固定數量的字元 例如 %3c
%s	char *	讀取用來字串 (至空格出現為止)
%n	int *	搭配 %s 使用 讀取到的字元數會存入參數中 假設 char a[100]; int len; 則 scanf("%s%n", a, &len); 之後 len 會 得到讀到的字元數量

另外我們還可以在整數型別前面加上 unsigned，用來表示這個整數變數不包含負數的部份

解決 without C99 C11mode

Project >> Build Option >>

分別在Debug跟Release中的"其他選項"手動加入 **-std=c99**

Debugger

確認 build option裡面有沒有勾選 produce debugging symbols

open debugging windows(介面左上角)

Demo2

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

```
int main(void)
```

```
{
```

```
int x; // integer 32-bit
```

```
double y; // 64-bit double precision floating point number
```

```
x = 789;
```

```
y = 3.1415;
```

```
// formatted output
```

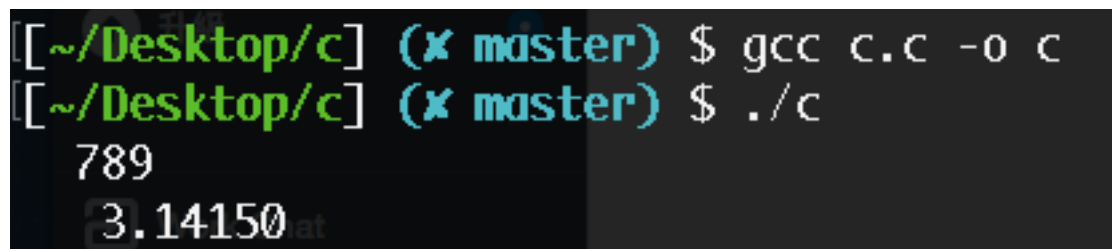
```
printf("%5d\n", x); // decimal
```

```
printf("%10.5f\n", y); // floating point
```

```
return 0;
```

```
}
```

Output:



```
[~/Desktop/c] (x master) $ gcc c.c -o c
[~/Desktop/c] (x master) $ ./c
  789
 3.14150
```

HW1-1

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

```
int main(void)
{
    int a;
    int n;
    int d;

    scanf("%d %d %d", &a, &n, &d);
    int nth = a+(n-1)*d;

    int sum;
    if(n>0 && n<1000 && -1000<a<1000 && -1000<d<1000)
    {
        if(n%2!=0)
        {
            sum=a*n + d*n*(n-1)/2;
        }
        else
        {
            sum=a*n + d*n*(n-2)/2 + n/2*d;
        }
        printf("%d %d", nth, sum);
    }
    else return 0;
}
```

HW1-2

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

```
int main(void)
{
    int n, x, y, z, b, x_butnot_y;
    scanf("%d %d %d %d", &n, &x, &y, &z);
    if(0<n<1000 && 0<x<1000&& 0<y<1000 && 0<z<1000)
    {
        b = z+x+y-n;
        x_butnot_y = x-b;
        printf("%d %d", b, x_butnot_y);
    }
}
```

HW1-3

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

int main()
{
    //input is string
    char input[7];
    scanf("%s", &input);
    //printf("%s\n", input);
    //convert number1 to double
    double number1;
    number1 = atof(input);
    //printf("%f\n", number1);

    //reverse number
    int temp = input[0];
    input[0] = input[2];
    input[2] = temp;

    temp = input[4];
    input[4] = input[6];
    input[6] = temp;

    //printf("%s\n", input);

    //convert number2 to double
    double number2;
    number2 = atof(input);
    //printf("%f\n", number2);

    double output;
    output = number1 + number2;
    printf("%.3lf", output);

    return 0;
}
```