

# 105學年 資料結構TA課

迴圈(Loop)x遞迴(Recursion)



# 遞迴 (Recursion)

- 遞迴？  
指函數本身又可以呼叫自己的副程式。
- 用於處理性質一樣的問題。
- 簡潔的程式碼。
- 用於解決大問題（將大問題拆解成小問題）  
$$\text{sum}(10) = 1+2+ \dots +10$$
$$\text{sum}(10) = \text{sum}(9)+10$$

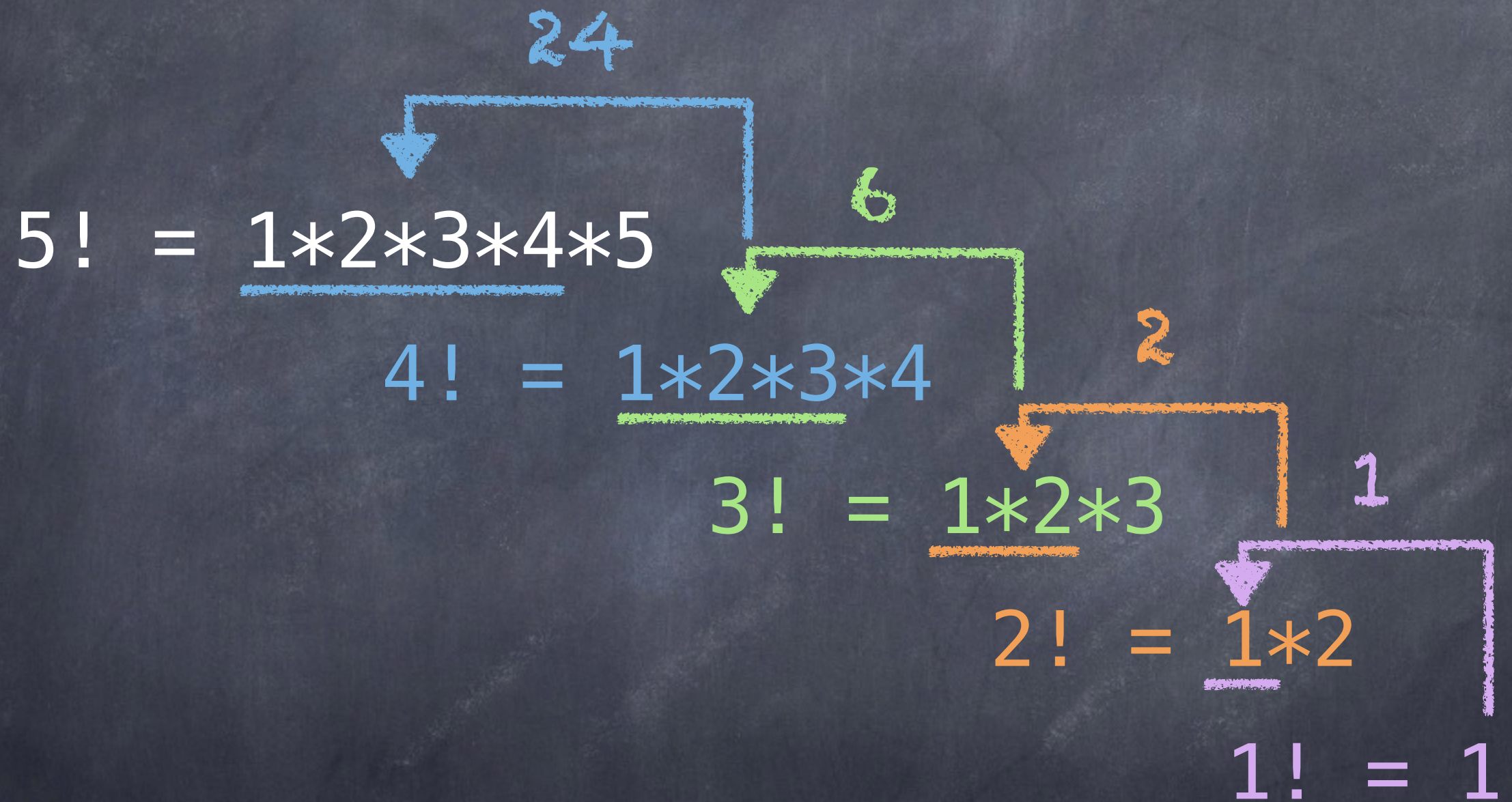


# 迴圈與遞迴

	遞迴	迴圈
優點	<ul style="list-style-type: none"><li>● 程式簡潔</li><li>● 節省記憶體空間</li><li>● 表達能力較高</li><li>● 區域變數與暫存變數較少</li></ul>	<ul style="list-style-type: none"><li>● 較節省執行的時間</li><li>● 不需額外堆疊空間</li></ul>
缺點	<ul style="list-style-type: none"><li>● 執行時間參數的存取較費時</li><li>● 需要額外堆疊空間支援</li></ul>	<ul style="list-style-type: none"><li>● 程式較長</li><li>● 浪費記憶體空間</li><li>● 表達能力較弱</li><li>● 區域變數與暫存變數較多</li></ul>



# N!





# N!

## 遞迴演算法

```
int N(int n)
{
    if(n == 1) return 1;
    return n*N(n-1);
}

int main()
{
    cout << N(5);
    return 0;
}
```

## 非遞迴演算法

```
int main()
{
    int N=1;
    for(int i=1; i<=5; i++)
    {
        N=N*i;
    }
    cout << N;
    return 0;
}
```



作業時間...

