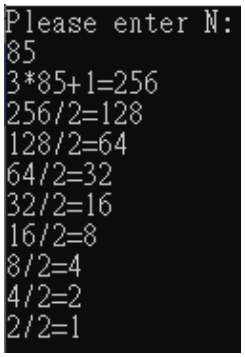
作業 6 函式與遞迴

6-1：函式基礎─3n+1 假說

3n+1 假說指的是任意取一數 n，如果 n 為奇數則讓 n 變成 3n+1，若 n 為偶數則讓 n 變成 n/2，周而復始直至 n 為 1 為止，該假說認為對任意數 n 都可以再不斷運算下得到最後的結果 1。下面以 85 為例子：



85→3*85+1=256→256/2=128→128/2=64→64/2=32→32/2=16→16/2=8→8/2=4→4/2=2→2/2=1

請撰寫一個函式：

- 當傳入值為奇數，回傳 3n+1
- 當傳入值為偶數，回傳 n/2

接著使用 while loop 不斷呼叫你寫的函式，直至回傳值為 1。

6-2：設計函式

請設計一個函式，其輸入值為一個陣列，輸出值為其最大值與最小值。即完成以下程式碼紅色部分。(不可修改黑色與藍色字部分)

- Hint：因為這裡需要回傳兩個值，需使用 Pass by reference (&)

```
#include <iostream>
#include <stdlib.h>
using namespace std;
void yourFunction(){

}
int main()
{
	int N;
	float m,n;
	cout << "Please enter the length of array" << endl;
	cin >> N;
	float *p = (float *) malloc(sizeof(float)*N);

	for(int i=0;i<N;i++)
		cin >> *(p+i);
	yourFunction(p,N,m,n);

	cout << "The maximum of your array is " << m << endl;
	cout << "The minimum of your array is " << n << endl;
	return 0;
}
```

```
Please enter the length of array
3
3 4 5
The maximum of your array is 5
The minimum of your array is 3
```

6-3：模組化程式

在之前的練習中，我們有寫過幫使用者算質因數的程式，但那時候我們沒有導入函式的概念，以至於 for 迴圈裏頭又有 for 迴圈，for 迴圈裏頭又有一大堆運算式。現在請你使用學到的函式概念，把質因數的兩個定義：判別是否為質數、判別是否為因數分別寫成一個函式，接著在 for 裡面直接呼叫這兩個函式，讓程式碼的可讀性與可維護性提高，也順帶感受一下函式的精隨。

回憶一下：該程式可以讓使用者不停地輸入一個數，而後判別該數是否為正數。但這次請你使用函式判別是否為質數與因數！

- 如果是正數便輸出該數的所有質因數後請使用者再行輸入。
- 但若輸入為負數，則輸出錯誤訊息並請使用者重新輸入。
- 但當使用者輸入 0 時結束程式。

```
Please enter an integer N (or 0 to leave) :
255
The prime factors of N are shown below:
3 is the prime factor of 255
5 is the prime factor of 255
17 is the prime factor of 255
-----------------------------
Please enter an integer N (or 0 to leave) :
-5
You enter a wrong N, please enter again !
-----------------------------
Please enter an integer N (or 0 to leave) :
0
-----------------------------
```

6-4：二分逼近法與遞迴

開根號、三角函數等運算往往需要耗費大量的時間與運算資源，然而在沒有公式輔助下，若需要的並不是完美的解答時，可以使用二分逼近法求出一個近式值。因此 6-4 的部分請你設計一個程式可以讓使用者輸入一個數 x 與可容許誤差 E，而後透過遞迴式利用二分逼近法求出$\sqrt{x}$的值，誤差範圍需在 E 以內。

Hint: https://zh.wikipedia.org/wiki/二分法_(數學)

```
Please enter a number(>1) and error :
5 0.000001
The square root is 2.23607
```

作業繳交方式：

請上傳至

http://www.lkm543.site/Course/Cpp/#Homework