

Homework 2 part 2

班級	學號	姓名	日期
四機械四乙	B10831020	吳宇昕	10/1/2022

Problem 2 宣告函式計算表面積

[source code](#)

```
double getSurfaceArea(double _radius, double _height)
{
    const double pi = 3.14159;
    return 2*pi*_radius*_height + 2*pow(_radius, 2)*pi;
}
```

main() 與終端機輸出

```
10 int main()
11 {
12     double radius, height;
13     std::cout << "Student B10831020" << std::endl;
14     std::cout << "Calculate the surface area of a cylinder" << std::endl;
15     std::cout << "Please enter the radius of the cylinder \t";
16     std::cin >> radius;
17     std::cout << "Please enter the height of the cylinder \t";
18     std::cin >> height;
19     std::cout << "\tThe surface area of this cylinder is " << getSurfaceArea(radius, height);
20     system("pause");
21     return 0;
22 }
```

PS D:\NTUST_Not_Sync\EngineeringProgramming\code\HW2\ > .\HW2-problem2.exe
Student B10831020
Calculate the surface area of a cylinder
Please enter the radius of the cylinder 34.2
Please enter the height of the cylinder 46.2
The surface area of this cylinder is 17276.7
Press any key to continue . . .

善用函式可以大幅增進程式的可讀性，把許多與整體應用程式邏輯脈絡無關的內容移至main()函式之外。一個函式應該只負責一項很鮮明的工作，不應該試圖讓同一函式身兼多職。把不同功能寫在不同函式裡，讓函式看起來意義更明顯，也更方便命名。

Problem 5 函式自我呼叫、迭代

[source code](#)

終端機輸出

```
1 #include <iostream>
2
3 uint64_t getFibNumber(int);
4
5 int main()
6 {
7     int index;
8     std::cout << "Enter a 0-based index of a Fib number\t";
9     std::cin >> index;
10    std::cout << "The Fib number is " << getFibNumber(index) << std::endl;
11    system("pause");
12 }
13
14 uint64_t getFibNumber(int _index)
15 {
16     if(_index < 2)
17         return _index;
18     else
19         return getFibNumber(_index-1) + getFibNumber(_index-2);
20 }
```

PS D:\NTUST_Not_Sync\EngineeringProgramming\code\HW2\CODE> .\HW2-problem5.exe
Enter the index of a Fib number 5
The Fib number is 5
Press any key to continue . . .
PS D:\NTUST_Not_Sync\EngineeringProgramming\code\HW2\CODE> .\HW2-problem5.exe
Enter the index of a Fib number 10
The Fib number is 55
Press any key to continue . . .
PS D:\NTUST_Not_Sync\EngineeringProgramming\code\HW2\CODE> .\HW2-problem5.exe
Enter the index of a Fib number 50
The Fib number is 12586269025
Press any key to continue . . .

- 回傳值使用`uint_64t`減緩memory overflow

尋找Fibonacci number的函式執行時間隨輸入index增加而指數上升，要找第50個Fibonacci number時明顯感受到電腦需要花相當久的時間才能找到。若C++函式沒有自我呼叫的功能，找Fibonacci number的函式恐怕會複雜得多，甚至有點不知道怎麼寫。

曾經聽說寫微電腦的程式應該避免使用迭代的技巧，以免它們計算資源不足，記憶體耗盡。若有辦法，之後我會更注意。

Problem 8

source code

終端機輸出

```
code > HW2 > CODE > C++ HW2-problem8.cpp > main()
1  #include <iostream>
2
3  void modifyArr(int _arr[], int count)
4  {
5      for(int i=0; i<count; i++){
6          _arr[i] = 0;
7      }
8  }
9
10 void displayArr(int _arr[], int count)
11 {
12     for(int i=0; i<count; i++)
13         std::cout << _arr[i] << std::endl;
14 }
15
16 int main()
17 {
18     int myInts[] = {32, 35, 67, 99, 3, 66};
19     modifyArr(myInts, 6);
20     displayArr(myInts, 6);
21     system("Pause");
22     return 0;
23 }
```

```
PS D:\NTUST_Not_Sync\EngineeringProgramming\code\HW2\CODER> .\HW2-problem8.exe
0
0
0
0
0
0
Press any key to continue . . .
```

將陣列傳入`modifyArr()`與`displayArr`兩函式，以陣列本身以及其大小作為函式的引數。一般的陣列跟現在c++的standard array可以像這樣將其傳入一函式，並修改其值，vector卻不行。由於vector只要增長，其記憶體位置就可能改變，不像array的記憶體位置是固定的，要在函式中修改vector必須將其pass by reference。