

Lab Assignment 4**Lab Grading Policy: Attendance 40%, Score 60%**

In case you have difficulty in finishing the exercises on time, you should upload them by **Thursday noon** with a penalty of 20% on your score. No late submission is permitted after that. We will in general post the reference solutions **by Friday**.

Exercise 1 (20%): (a) Write two functions: one named `average` and the other named `standardDev`. Both functions take two parameters: the first one is an array parameter of an array of `int` and the second one is a `size_t` that refers to the size of the array. The `average` function will return a `double` that is the average of these integers. The `standardDev` function will return a `double` that is the standard deviation of these integers. Initialize an array in the main program, such as {3, 10, 40, 5, 12, 7, 22, 39}:

```
The result:
Average = 17.25
Standard deviation = 13.9261
```

(b) Repeat (a) but now write the `average` and `standardDev` functions with a single parameter: the `vector` of `int`. In the main function, allow users to input as many positive integers as one wishes and use -1 to signify the end of inputs. Pass the input vector into the `average` and `standardDev` functions and print the outcomes. A sample run looks like:

```
Input positive integers: 3 10 40 5 12 7 22 39 -1
The result:
Average = 17.25
Standard deviation = 13.9261
```

Exercise 2 (20%): Write a program that uses a (a) **loop** and (b) **recursion** in a function `factorial()` to find the factorial with the following signatures, respectively: for (a):

```
void factorial(int n, int& result );
```

and for (b):

```
int factorial(const int n);
```

The following is the `main()` function you CANNOT change:

```
int main()
{
    int n, result;
    cout<< "Please input n: ";
    cin >> n;
```

```
        factorial(n, result);  
        // or for (b):  
        // result = factorial(n);  
        cout << "n! is " << result << endl;  
  
    return 0;  
}
```

Sample runs look like:

```
Please input n: 5  
n! is 120
```

```
Please input n: 3  
n! is 6
```

Exercise 3 (20%): Write a collection of functions with the name `lab43 (...)` to test the function overload. **Note:** In C++, string literals (strings in a pair of double quotes without storing it in a variable) are immutable (they are constants you cannot change), and are stored in a read-only memory location.

The following is the test program, with a `main()` you cannot change.

```
int main(){  
    lab43();  
  
    int i = 2;  
    lab43(i);  
    cout << "main: i++ = " << i << endl;  
  
    string s = "abcde";  
    lab43(s);  
    lab43("fghij");  
  
    return 0;  
}
```

You can test your function overload by commenting some of the code out, one by one, if you like. The complete output is as follows:

```
no arg version!  
lab43: i++ = 3  
main: i++ = 3  
lab43(s): abcde  
lab43(s): fghij
```

Exercise 4 (self challenge): Write a program that uses a **recursive** function `gcd()` to find the greatest common divisor (最大公因數). The function has the following signature:

```
int gcd(int a, int b);
```

The following is the `main()` function you CANNOT change:

```
int main(){
    int x, y;
    cout<< "Please input the two integer numbers that you want to find
the GCD of: ";
    cin >> x >> y;

    cout << "The GCD of " << x << " and " << y << " is " << gcd(x, y) <<
endl;
    return 0;
}
```

Sample runs look like:

```
Please input the two integer numbers that you want to find the GCD of: 259 111
The GCD of 259 and 111 is 37
```

```
Please input the two integer numbers that you want to find the GCD of: 10 5
The GCD of 10 and 5 is 5
```