

Lab 1

Lab 1 对应 lecture notes 的 Lecture 1 (课程简介)，训练目标是能够正确安装和使用 C++ 集成开发环境 (IDE)，并利用 IDE 编写简单的 C++ 程序。

Problem 1.

(a) 参考附件“VS2019 下载安装说明”, “codeblocks 下载安装说明”，或者互联网上的教程，在自己的电脑上安装 Visual Studio 2019 或者 codeblocks，或者其他 C++ 的 IDE

(b) 参考附件“visual studio 运行程序演示”，“codeblocks 运行程序演示”，或者互联网上的教程，在安装好的 C++ IDE 上编写、运行程序，在控制台输出“hello world!”

Problem 2.

(a) 编写程序，使用 C++ 中的 cout，输出下面图形：

```
*
***
*****
*****
*****
***
*
```

(b) 编写程序，从键盘输入任意 1 个字符，使用该字符输出上面的图形。

例子：

输入：

@

输出：

```
@
@@
@@@
@@@@@
@@@
@@
@
```

Problem 3.

编写程序，完成以下功能：从键盘输入 2 个整数（用空格隔开），分别实现加、减、乘、除和取模（求余数）运算，分行输出结果。

例子：

输入：

3 5

输出：

8

-2
15
0
3

提示：C++中的乘法运算符是*，除法是/，取模是%

Problem 4.

编写程序，完成以下功能：从键盘输入 1 个 3 位数的正整数，反序输出该正整数。

例子：

输入：

123

输出：

321

Problem 5.

编写程序，完成以下功能：从键盘输入 1 个浮点数，代表圆的半径，计算并输出圆的面积，结果保留 3 位小数（圆周率取 3.14159）。

例子：

输入：

3.2

输出：

32.170

提示：(1) 在程序最前面增加 `#include <iomanip>` (2) 对于变量 x，保留 2 位小数可以通过 `cout<<fixed<< setprecision(2)<<x;` 来实现

Problem 6.

编写程序，完成以下功能：从键盘输入 3 个字符（空格隔开），在同一行反向输出这 3 个字符，用空格隔开。

例子：

输入：

A B C

输出：

C B A

Problem 7.

(a) 编写程序，完成以下功能：从键盘输入 1 个不包含空格的字符串，输出这个字符串的长度（即包含字符的个数）。

例子：

输入：

hello

输出：

5

提示：(1) 在程序开头加上 `#include <string>` (2) 对于一个 string 类型的变量 s，`s.size()` 可以得到字符串的长度

(b) 编写程序，完成以下功能：从键盘输入 1 个包含空格的字符串，输出这个字符串的长度（包含空格在内）。

例子：

输入：

hello world

输出：

11

提示：对于一个 string 类型的变量 s，cin>>s 并不能读入包含空格的字符串，需要用 getline(cin, s)来实现

(C) 编写程序，完成以下功能：从键盘分两行分别输入 1 个字符串（可能包含空格）和一个整数 N（整数小于字符串的长度），输出字符串中的第 N 个字符（位置从 0 开始计）

例子：

输入：

Hello

1

输出：

e

提示：(1) 输入的不要把字符串和整数放在同一行，否则整数也被认为是字符串的一部分了，正确的做法是：第一行先输入字符串，回车，第二行再输入整数，回车；(2) 对于一个 string 类型的变量 s，s[N]可以获得 s 的第 N 个字符；

(d) 学习下面程序，掌握更多字符串的操作。

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main() {
```

```
    string s = "hello"; //定义字符串变量 s
```

```
    string s2 = "world"; //定义字符串变量 s2
```

```
    int i = 1;
```

```
    cout<<s[i]<<endl; //输入字符串 s 的第 i 个字符（位置从 0 开始计）
```

```
    s[i] = 'D'; //将字符串 s 的第 i 个字符替换为'D'
```

```
    cout<<s.substr(1,2)<<endl; //输出字符串 s 从位置 1 到位置 2 的子串
```

```
    cout<<s.size()<<endl; //输出 s 的长度
```

```
    cout<<s+s2<<endl; //输出字符串 s 和 s2 连接后组成的新字符串
```

```
    cout<<s.append(s2)<<endl; //将 s2 连接到 s 后面，并输出新的 s
```

```
    cout<<s.insert(1,"abc",3)<<endl; //在字符串 s 的位置 1 处插入字符串"abc"的前 3 个字符
```

```
    cout<<s.replace(1,3,"defg")<<endl; //将字符串 s 从位置 1 到 3 的子串，替换为 defg
```

```
    cout<<s.erase(1,3)<<endl; //将字符串 s 从位置 1 到 3 的子串删除
```

```
    return 0;
```

```
}
```

程序输出结果：

e

DI

5

hDlIoworld

hDlIoworld

habcDlIoworld

hdefgDlIoworld

hgDlIoworld