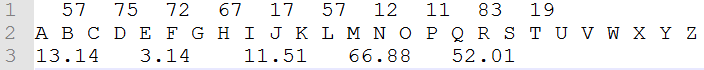
1. 新建一个输出文本文件 **data.txt**（以绝对路径和相对路径方式均可），并向该文件中写入以下内容：

* **第一行**：写入10个0~99的随机整数（每个整数占4个输出域宽）
* **第二行**：写入A~Z 的26个字符（字符间以空格字符’ ’进行分割）
* **第三行**：写入5个任意的浮点数（浮点数间以字符’\t’进行分割）



1. 新建一个输出二进制文件 **binary.bin**（以绝对路径和相对路径方式均可），并向该文件中写入以下5个Complex类的对象：

// Complex 类定义如下

class Complex{

public:

Complex(double \_real = 0.0, double \_image = 0.0): real(\_real), image(\_image)

{ }

void set(double \_real, double \_image){

real = \_real;

image = \_image;

}

double getReal() const{

return real;

}

double getImage() const{

return image;

}

void print() const{

cout<<real<”+”<<image<<”i”<<endl;

}

protected:

double real;

double image;

};

Complex 类对象如下：

Complex cp1(1.0, 2.0);

Complex cp2(1.1, 2.5);

Complex cp3(2.0, 3.0);

Complex cp4(4.0, 5.5);

Complex cp5(6.6, 8.8);

1. 打开输入文本文件 **data.txt**，读取该文件中的数据并输出显示到屏幕上，显示格式如下：

* **第一行**：输出写入的10个随机整数（每个整数占4个输出域宽）
* **第二行**：输出写入的A~Z 的26个字符（字符间以空格字符’ ’进行分割）
* **第三行**：输出写入的5个浮点数（浮点数间以字符’\t’进行分割）

1. 打开输入二进制文件 **binary.bin**，读取该文件中的5个Complex类对象的数据，并调用Complex类的成员函数print将其输出显示到屏幕上。