# Exercise4-1
# (Input/Output)

## Q1

Implement FileCopyer that receives arbitrary number of input filename and one output filename. This class must contain copy(File srcFile, File destFile) method that concatinates contents of srcFile to the end of destFile and you have to use it. Note that, as shown in the following example, the last filename should be the output filename. If the output file already exists, you can overwrite it.

Definision of copy method

```
public void copy(File srcFile, File destFile) throws Exception {
  // some impelmentation...
}
```

Execution example

```
%cat src1.txt
aaa
bbb
ccc

%cat src2.txt
ddd
eee
fff

%java FileCopyer src1.txt src2.txt output.txt
%cat output.txt
aaa
bbb
ccc
ddd
```

```
eee
fff
```

## Q2

Implement BinaryInput that receives filename as String in the constructor, and contains the following methods. Also implement Q42Main that uses BinaryInput.

- readChar: read 1 byte from the file and return it as char.
- readShort: read 2 byte from the file and return it as short.
- readInt: read 4 byte from the file and return it as int.

If there is no data in the file, you will throw RuntimeException. Also you can define other methods if needed. Note that, the length of the file is **undefined**, meaning that you **cannot** read the entire data on the memory (because of the too big data that cannot be stored on memory).

A sample file (test.dat)

```
abcdefghijklmnopqrstuvwxyz
```

An example of Q42Main and result(1)

```
public class Q42Main {
    public static void main(String[] args) {
        BinaryInput input = new BinaryInput("test.dat");
        for(int i = 0; i < 26; i++) {
            System.out.println(input.readChar());
        }
    }
}

%java Q42Main
a
b
c
d
e
f
g
h
i
j
k
l
m
n
```

```
o
p
q
r
s
t
u
v
w
x
y
z
```

An example of Q42Main and result(2)

```
public class Q42Main {
    public static void main(String[] args) {
        BinaryInput input = new BinaryInput("test.dat");
        for(int i = 0; i < 13; i++) {
            System.out.println(input.readShort());
        }
    }
}

%java Q42Main
24930
25444
25958
26472
26986
27500
28014
28528
29042
29556
30070
30584
31098
```

An example of Q42Main and result(3)

```
public class Q42Main {
    public static void main(String[] args) {
        BinaryInput input = new BinaryInput("test.dat");
        for(int i = 0; i < 6; i++) {
            System.out.println(input.readInt());
        }
    }
}

%java Q42Main
1633837924
1701209960
1768581996
1835954032
1903326068
1970698104
```

# Q3

Implement TinyDataInputStream and TinyDataOutputStream. TinyDataInputStream contains readChar and readInt methods:readChar read data as char type and readInt read data as int type. In this case, char is 2 byte (standard in Java language) different from **Q2**. On the other hand, TinyDataOutputStream contains writeChar and writeInt methods that are opposit of readChar and readInt respectively. Both classes should contain close() method to close the stream.

You can confirm the behavior by using Q43Main as follows. Note that, you **cannot** use DataInputStream/DataOutputStream that Java contains, and you can define endian (e.g., big or little) as you like.

An example of Q43Main and result

```
public class Q43Main {

    public static void main(String[] args) {
        try {
            FileOutputStream out = new FileOutputStream("output.dat");
            TinyDataOutputStream tout = new TinyDataOutputStream(out);
            tout.writeChar(’a’);
            tout.writeInt(10);
            tout.writeChar(’b’);
            tout.writeInt(20);
            tout.writeChar(’c’);
            tout.writeInt(30);
            tout.close();

            FileInputStream in = new FileInputStream("output.dat");
            TinyDataInputStream tin = new TinyDataInputStream(in);
            System.out.println(tin.readChar());
            System.out.println(tin.readInt());
            System.out.println(tin.readChar());
            System.out.println(tin.readInt());
            System.out.println(tin.readChar());
            System.out.println(tin.readInt());
            tin.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

%java Q43Main
a
10
```

```
b
20
c
30
```