Q1

Class files submitted separately. The Java Byte Code for the two programs is given below. I came across an issue where read gets stuck in an infinite loop due to System.in.read returning -1 when the programs are run from within the scala console. The programs worked as expected when run from the standard console.

```
Idc "Fib"
                        ; "Fib"
 invokestatic fib3/fib3/writes(Ljava/lang/String;)V
 invokestatic fib3/fib3/read()I
 istore 0
                        ; n
 ldc 0
 istore 1
                        ; minus1
 ldc 1
 istore 2
                        ; minus2
Loop_begin_2:
 iload 0
                        ; n
 ldc 0
 if_icmple Loop_end_3
 iload 2
                        ; minus2
 istore 3
                        ; temp
 iload 1
                        ; minus1
 iload 2
                        ; minus2
 iadd
 istore 2
                        ; minus2
 iload 3
                        ; temp
 istore 1
                        ; minus1
 iload 0
                        ; n
 ldc 1
 isub
 istore 0
                        ; n
 goto Loop begin 2
Loop end 3:
 Idc "Result"
                        ; "Result"
 invokestatic fib3/fib3/writes(Ljava/lang/String;)V
 iload 2
                        ; minus2
 invokestatic fib3/fib3/write(I)V
```

```
Idc "Fact"
                         ; "Fact"
 invokestatic fact/fact/writes(Ljava/lang/String;)V
 invokestatic fact/fact/read()I
 istore 0
                         ; n
 ldc 1
 istore 1
                         ; result
Loop_begin_5:
 iload 0
                         ; n
 if_icmple Loop_end_6
 iload 1
                         ; result
 iload 0
                         ; n
```

```
imul
 istore 1
                        ; result
 iload 0
                        ; n
 ldc 1
 isub
 istore 0
                        ; n
 goto Loop begin 5
Loop end 6:
 Idc "Result"
                        ; "Result"
 invokestatic fact/fact/writes(Ljava/lang/String;)V
                        ; result
 invokestatic fact/fact/write(I)V
Q2
Java Byte Code for the example for loop:
ldc 2
 istore 0
                        ; i
Loop_begin_3:
 iload 0
                        ; i
 ldc 4
 ldc 1
 iadd
 if_icmpge Loop_end_4
                        ; i
 invokestatic forTest/forTest/write(I)V
 iload 0
                        ; i
 ldc 1
 iadd
 istore 0
                        ; i
 goto Loop_begin_3
Loop_end_4:
Q3
```

The program prints the numbers 1 to 10. This is because the variable 'i' has a global scope within the program so the inner loops increments 'i' to 10, which also meets the condition for exiting the outer loop.

The Java Byte Code for the program:

```
ldc 1
 istore 0
                        ; i
Loop_begin_9:
 iload 0
                        ; i
 ldc 10
 ldc 1
 iadd
 if_icmpge Loop_end_10
 ldc 1
 istore 0
                        ; i
Loop_begin_12:
 iload 0
                        ; i
 ldc 10
```

```
ldc 1
 iadd
 if_icmpge Loop_end_13
 iload 0
            ; i
 invokestatic for2/for2/write(I)V
 iload 0
                    ; i
 ldc 1
 iadd
 istore 0
                     ; i
 goto Loop_begin_12
Loop_end_13:
 iload 0
                     ; i
 ldc 1
 iadd
 istore 0
                     ; i
 goto Loop_begin_9
Loop_end_10:
```