Student Name and Surname:

Problem 1 (0.5 points)

SECTION 1. Create a function called "CalculateGrade". This function should operate as follows: (0.25 points)

- The system will verify that the data item (grade) entered by the user is between 0 and 10. If it is not, it will ask for another grade.
- If the grade is less than 5, the system will print "Fail" on the screen. If the grade is greater than 5 but less than 7, the system will display "Pass". If the grade is from 7 to less than 9, it will be "Good". If the grade is 9 or higher, it will be "Excellent".

Examples:

```
Example1.
 CalculateGrade()
[1] "Insert a grade: "
1: 2
Read 1 item
[1] "Fail"
```

```
Example2.
CalculateGrade()
 [1] "Insert a grade: "
1: 12
Read 1 item
[1] "Insert a grade:"
1: 6
Read 1 item
[1] "Pass"
```

SECTION 2. Modify the CalculateGrade function created in Section 1 so that, apart from the above functionality, it searches the number of student grades by parameter that the user is required. This function will be called: CalculateGradesParameter. (0.25 points)

```
> CalculateGradesParameter(2)
[1] "Insert a grade:"
1: 1
Read 1 item
[1] "Fail"
[1] "Insert a grade: "
1: 7
```

```
Read 1 item
[1] "Good "
```

```
> CalculateGradesParameter(3)
[1] "Insert a grade: "
1: 10
Read 1 item
[1] "Excellent "
[1] "Insert a grade: "
1: 1
Read 1 item
[1] "Fail"
[1] "Insert a grade: "
1: 11
Read 1 item
[1] "Insert a grade:"
1: 6
Read 1 item
[1] "Pass"
```

Problem 2 (1 point)

Section1. Define a function called "StudentGrades" defining a data structure to describe the table below; insert the names of the rows and columns and the values contained in that table. After building the table, modify the Omar's grade for French to 5. Display the table on screen. (0.25 points)

Table 1: Data

	English	French
Javier	8	9
Ana	1	4
Pepe	3	7
Rosa	2	8
Juan	6	3
Alba	8	4
Omar	5	4

Section2. Define a function called "CountStudentPasses". The function should add three rows to Table1 without creating a new matrix. It should also count the number of students that have: 1) passed both subjects; 2) passed English but not French; 3) passed French but not English; 4) not passed either subject. Additionally, it should display the data on screen. (0.5 points)

The shaded rows should be added without creating a new array:

Table2. Table 1 with new rows

	English	French
Javier	8	9
Ana	1	4
Pepe	3	7
Rosa	2	8
Juan	6	3
Alba	8	4
Omar	5	5
Alberto	2	8
Ana	10	3
Pedro	7	4

```
> CountStudentPasses ()
          English French
Javier     6     9
Ana        1     4
Pepe       3     7
Rosa       2     8
Juan      6     3
Alba       8     4
Omar       5     5
Alberto    2     8
Ana       10     3
Pedro      7     4
[1] "Number of students passing English and French:
 [1] 2
[1] "Number of students passing English approved and failing French:
 [1] 4
[1] " Number of students failing English and passing French:
 [1] 3
[1] " Number of students failing both English and French:
[1] 1
```

SECTION 3. Define a function called "CountStudents". The function should count students who have higher grades than the English and French grades passed by a parameter. It should also display: the number of students; the number of students whose grades are greater than or equal to the specified grades; the number of students that have lower than the specified grades; and their percentages in both cases. (0.25 points)

Example:

```
> CountStudents(3,7)
         English French
```

```
Javier     6    9
Ana        1    4
Pepe       3    7
Rosa       2    8
Juan       6    3
Alba       8    4
Omar       5    5
Alberto    2    8
Ana       10    3
Pedro      7    4
[1] "Number of students: "
[1] 10
[1] "Number of students that have higher than the specified grades is: "
[1] 2
[1] "And the percentage is (%): "
[1] 20
[1] " Number of students whose grades are not higher than the specified grades is: "
[1] 8
[1] "And the percentage is (%): "
[1] 80
```