

Lab-session 7

- 1) Consider the table given below and suppose that you enter the data on this table into the range A1:D6 of your Excel worksheet:

Student's name	CW1	CW2	Exam
Kadiri	45	60	50
Lai	67	78	60
Phull	89	67	44
Ramnath	23	50	45
Sanchez	90	75	76

The first column of the table contains the names of various second year students doing a particular course. The second column contains the coursework marks obtained by each student for their first coursework. The third column contains the marks obtained by each student for their second coursework. Finally, the fourth column contains the mark obtained by each student in the final exam. The final mark that each student is going to obtain for the module is given by the following formula:

$$\text{Finalmark} = \frac{1}{100} (10 \times (\text{CW1} + \text{CW2}) + 80 \times \text{Exam}) \quad (0.1)$$

Write two user defined functions:

- i) The first function should be called **fmark** and given the name of a student in the list it would compute the final mark by means of the formula (0.1). Define all variable types and employ the worksheet function **VLOOKUP**.
- ii) The second function should be called **howgood** and take again as an input the name of a certain student. The function **howgood** should make use of the function **fmark** defined in i) and employ a **IF...ELSEIF** structure. Depending on the name of the student, the function should return one of the following values:
 - If the value of **fmark** for the student is lower than 40 it should return the message: “the student will have to re-sit the exam”
 - If the value of **fmark** for the student is between 40 and 54 it should return the message: “this is an average student”
 - If the value of **fmark** for the student is above 54 and under 74 it should return the message: “this is a good student”
 - If the value of **fmark** for the student is bigger or equal to 74 it should return the message: “this is a very good student”
- iii) Compute **fmark**(“Lai”) and **howgood**(“Sanchez”).

- 2) Recall the function `bmi` from Lecture 8, which computes the body mass index given the weight in Kilograms and the height in meters.
- i) Produce a table which labels columns by heights from 1,30 meters to 2,00 meters in steps of 10 cm and rows by weights from 50 Kg to 100 Kg in steps of 5 Kg. In each intersection of the table compute the corresponding body mass index. (Use extensively the `autofill` function to produce this table!)
 - ii) Produce two tables in which columns and rows are labelled the same way as in a). In each intersection of the table compute now the meaning of the body mass index, i.e. “normal weight, “overweight” etc (a quick way of doing this is by using the functions `male(bmi)` and `female(bmi)` defined in the last lecture). Make one table for male and one for female.
 - iii) Write down the code of a user defined function that returns the body mass index of a person, given the height and weight as input. The function should extract the body mass index from the table of part i) by using `VLookup` or `HLookup` structures (there are several ways of writing the code). It should work so that if a weight or height are entered which are not in the table, it would select the data for the weight or height that are closest and smaller or equal to the one given as input.