(c) Create a list with the name `shortestNames` which contains all shortest names and then display the list. Shortest names are those names with the smallest number of characters.

(d) Create a list with the name `longestNames` which contains all longest names and then display the list. Longest names are those names with the highest number of characters.

(e) Prompt the user to enter name of a girl and then display how many names matches with the entered name. The name should not be case sensitive. If the specified name does not exist, display the message:

```
Entered name …. does not exist in the list
```

and the program should terminate. If the user presses ENTER key without entering the name, then the program should display error message `Sorry, please specify the name` and the user should re-enter the name until a name has been entered. Note that all names in the list have small letters except the first letter which is capitalized.

Test your code with following (enter the name as specified) (i) gRace (ii) Kaboya (iii) EmMA (v) eliza (iv) ENTER key (without specifying anything)

(f) Prompt user to enter a letter `'a'..'z' or 'A'..'Z'`. The letter should not be case sensitive. Note that there is a possibility of user entering more than one character and the program should be able to handle that. If the specified letter is not in the range, display the message: `Entered letter .. is not in the range, try again` and user should re-enter the letter. If there are names which starts with that letter, display the names in descending order in one line (separate the names with a space), otherwise, display the message: `No name starts with the letter ….` and the program terminates. If user enteres more than one character, the program should consider only the first character of the word entered.

Test your code with following entries `(i) 'j' (ii) 'b' (iii) 'mEA' (v) 'WEA' (iv) ? (v) 'Q' (vi) ENTER key (without specifying anything)`

Question 02

The excel file `stdsscores.xls` (provided) contains scores (out of 100) for Undergraduate students at CoICT in 2 tests and UE for a certain course. Two tests (Test1 and Test2) contributes to Continuous Assessment (CA). Weights for Test1 and Test2 in CA are 45% and 55% respectively. Final Mark (FM) out of 100 is calculated as: FM=CA/50+UE/50 where both CA and UE are out of 100. CA and FM should be in 1 decimal place.

Grades ranges are:

```
A: 70≤ FM ≤ 100;  B+: 60 ≤ FM < 70;  B: 50 ≤ FM < 60
C: 40 ≤ FM < 50; D: 35 ≤ FM < 40; E:0 ≤ FM <35
```

Pass mark is grade C.

Remark is either P or F where P stands for Pass and F for Fail.

Using excel formulas:

(a) Fill columns CA, FM, Grade and Remark.

At the bottom of excel file:

(b) Display the minimum, maximum and average for Test1, Test2, CA, UE and FM. Averages should be in 2 decimal places.

(c) Display how many students have passed and how many have failed.

(d) Display statistical summary of Grades as shown in the excel file.

(e) Create a copy of the worksheet (within the same workbook) and call it `sorted_FM`. Sort `sorted_FM` in `descending` order of FM.

Question 03

The file `stdsscores_file.py` (provided) contains a list (with the name `stdsscores`) of scores mentioned in Question 2 above. Each row contains a data for each student in the format:

```
[SN, Test1, Test2, UE]
```

Write a program to perform following services. You need to **import** the list in your workspace. Answers should be obtained programmatically **NOT** by manual means:

(a) Display how many students in the list.

(b) Create a copy of `stdsscores` list and call it `fscores.` Display the last 5 records from `fscores.`

Use `fscores` to answer following questions:

(c) Calculate `CA`, `FM` (in 1 decimal place), `Grade` and `Remark` for each student and update `fscores`. That means each row of `fscores` will have following format in specified order: `[SN, Test1, Test2, UE, CA, FM, Grade, Remark]` and display the

first five records. Compare your results for `CA`, `FM`, `Grade` and `Remark` with those obtained using excel.

(d) Display minimum, maximum, and average for Test1, Test2, CA, UE and FM. Averages should be in 2 decimal places. Display the values in neat tabular form as follows:

	Test1	Test2	CA	UE	FM
Minimum	x	x	x	x	x
Maximum	x	x	x	x	x
Average	x	x	x	x	x

(e) Display record(s) of student(s) who scored lowest mark and also student(s) who scored the highest mark in the FM. Note that there is a possibility of having more than one in each category.

(f) Create a dictionary called `gradesSummary` which contains number of students for each grade. Sample of `gradesSummary` is (`in sorted order of grades`)

`gradesSummary={'A':...,'B':...., 'B+':...,'C':...,'D':....,'E':...}.`

Compare your answer with that from Excel and comment in the pdf file.

(g) Display grade(s) with the highest number of students and grades with lowest number of students. State also number of students in those grades. Sample display:

`Grade(s) with the lowest number of students are:……. and number of students is: ……..`

`Grade(s) with the highest number of students are:……. and number of students is: ……..`

(h) Display record of a student whose S/N is specified from the Keyboard. If the serial number specified is not in the allowable range, display error message:

`Wrong entry! Valid ranges are … up to ….. try again.`

(i) It has been found that one student was forgotten in the list and has to be added at the end of `fscores.` His scores are: `Test1=45.6, Test2=76.4, UE=65.5.` Add that record to `fscores` and display the last records from `fscores` and check if the record corresponds to added student.

(j) Display the best top 5 students based on FM in the neat tabular form as follows (where x stands for values.

SN	FM
x	x
x	x
…..	……
x	x

Tip: Create a new dictionary containing Serial numbers as keys and FM as values and call it `fmDict`. Using `fmDict`, create a 2 dimension list `fmList2D` containing [`FMvalue,SN`] pairs. Then form another list `fmList2DSorted` in descending order. Display the first 5 records from `fmList2DSorted` but in the form `SN FMvalue`.

Note: You are free NOT to use given tip. Sample display is as follows. Compare your answer with that of Question 2(e).

```
SN      FM
197     92.7
229     91.8
113     89.1
…..     ……
```

=== END ===