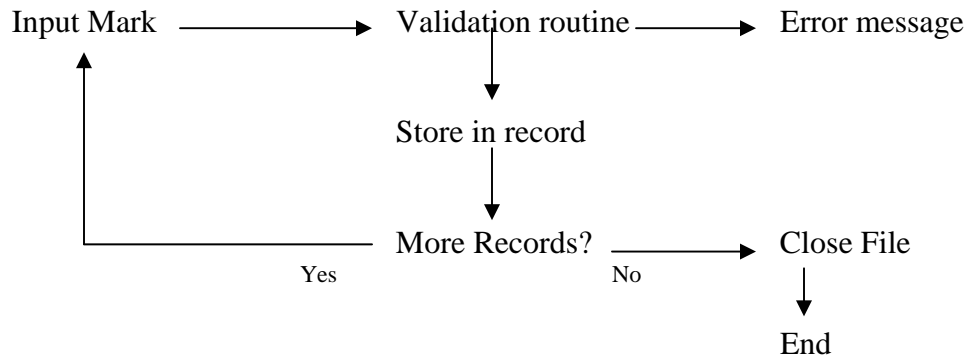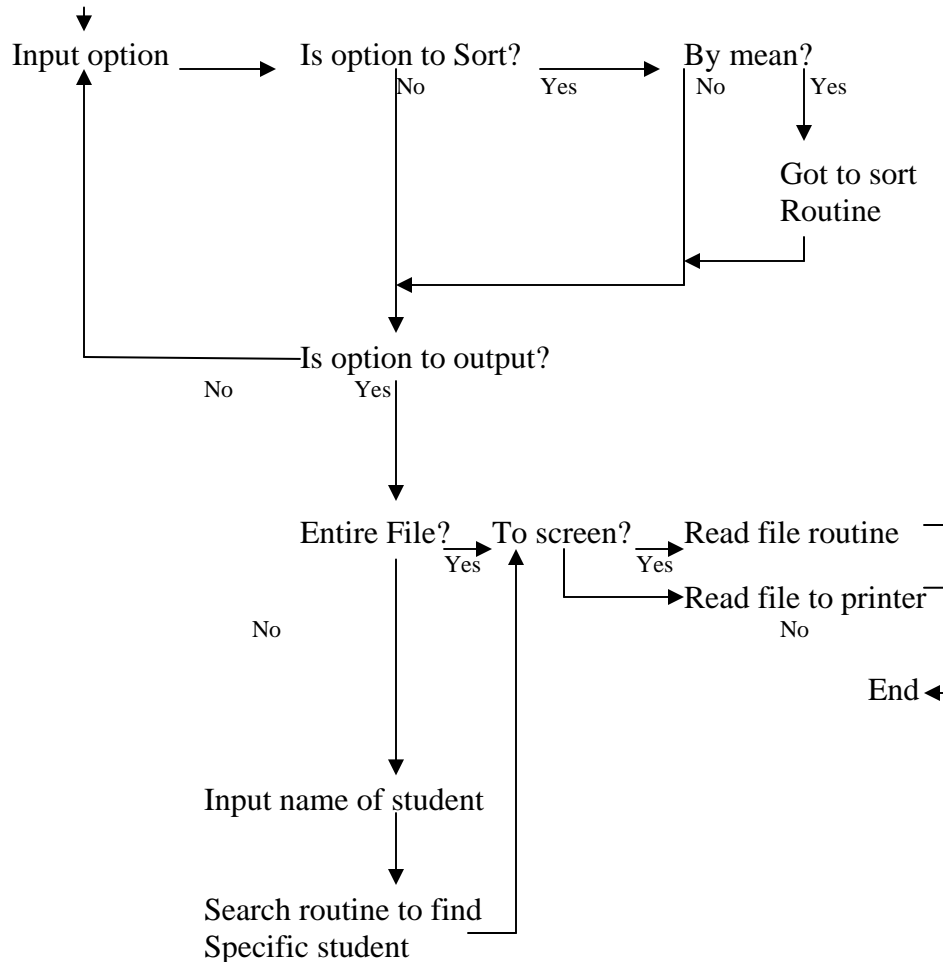## Part (b)   Program Design

The first stage in this part requires the student to find out in detail from the end user what needs to be done in order to be successful. In my case I have talked to Mr. Jotas and found out that I need to store the names of the students in the class (there will be a maximum of 20 at any one time. In addition, for each student, the system needs to store a number of homework marks. Mr. Jotas expects the system to be able to work out the mean mark for each student and to output, to screen or printer, the values for a specific student or for the whole set in either alphabetic order or in descending order of their mean mark. All the detail is now in place. The next stage is to make some sense of it so that I can write the program. The most sensible way of doing this is to draw a flow diagram showing how all the items fit together and in what order they should be done.

Input:

Input Mark ⟶ Validation routine ⟶ Error message

Store in record

More Records?   ⟶ Close File

Yes   No

End

Output:

Display option screen

Input option ⟶ Is option to Sort? — No / Yes ⟶ By mean? — No / Yes

Got to sort Routine

Is option to output? — No / Yes

Entire File? — Yes / No ⟶ To screen? — Yes / No ⟶ Read file routine

Read file to printer

End

Input name of student

Search routine to find Specific student

I would ideally need to produce flow diagrams for my modules which are going to be the creation of, the searching of and the sorting of the file. Note that these diagrams do not need to be produced on the computer. Far better is hand drawn on paper using a pencil and a ruler. Notice also that there are no pretty boxes around the stages. They are not necessary, use them if they help you to see what is going on but they do not need to be done, but what the examiner is interested in seeing is whether you can understand the logical steps that are necessary to solve the problem and the order in which they should be carried out.

I have now specified what is to be stored in the file and I have drawn diagrams to show the logic of my proposed solution. The next thing that I need to do is to address what the input and output screens will look like. There is no problem here because I was careful to make sure that Mr. Jotas was only interested in the details being output, not the form of the output. So, I only need to describe the screen with my simple text statements for the input and the output being in tabular form or in a single row of text if the selection is one student only.

I need to specify the validation procedure for inputs to my file. These are going to be a simple range check and a character check on the marks added to the file. This will be the only input to the file apart from the names of the students which can be validated by a character check.

The syllabus says that I have to specify the hardware that I need to use. Keep it simple. I need: a processor, a keyboard for the teacher to input details of students and to select outputs, a screen (can be low resolution and monochrome if necessary to keep costs down) to see the outputs from the system, a hard disk drive to act as storage for the file and software, a floppy disk drive to allow the teacher to make a copy of the file for safe keeping (he specified this in part a), and a printer to print hard copy outputs.

Finally, I need to describe how I am going to solve the problem. I will have an initial screen that will allow four choices. The first will be to input new data to the file, the second will be to request a sort of the data, the third will be a calculation of the means for the students and the last option will be to ask for an output of the data. Each one of the four options will lead to an individual module so that I can solve the problem in a modular fashion.

Notice that the requirements for this section are not complex. Approach the different tasks one at a time, do not try to do the whole thing in one go. The requirements of this section should take no more than 5 sides of A4. Think of half a side for each of the file description and the hardware requirements, a side for each of the input/output screen designs and the written description of how the problem will be solved, and finally 2 sides for the diagrammatic description of the solution. These are only rough figures and don't worry if you produce more than this, but the point is that it is not required and if it takes more than this you are either providing a lot of information which is not necessary or your problem solution is far more complex than it needs to be.

Evidence required
- there needs to be a file to store data as this is one of the requirements of the programming, this file needs to be detailed here. Ensure that you have included details of name, size, data types and the data that will be held.
- a diagrammatic representation of your proposed solution, this will probably be a flow diagram to illustrate the logic of your solution
- simple diagrams of what the input and output screens will look like. These should be drawn rather than being printouts as, like the rest of this section, they should be done before the programming is attempted.
- the hardware requirements for the system. Don't go into too much detail, keep it simple. Nobody wants to know about the type of mother board or the size of the memory (after all I'm only going to be using a file with 20 students in it!)
- a written description of how you intend to solve the problem. Keep it short, no more than a side of A4.