Task 3. Mobile Subjects

Available marks: 21

Smartphones are designed to allow users to select an option quickly from a fairly short list. When the list is very long, say over a hundred items, it's more efficient to introduce an intermediate selection to avoid tedious scrolling. The intermediate selection often derives from alphabetic ranges representing the first letter of contiguous sublists, such as A–J (all items starting with A, B, C, ..., I, J), K–P and R–Z. It works best if there are few ranges (say up to 12 or so) and the sublists represented have roughly the same number of items. In this scheme no initial letter can be split, for example, all the Cs must be in the same list.

The UNSW timetable app (yet to be implemented) is an example. It will allow users to look up class times based on the full course (subject) code. Courses are grouped into 148 4-letter subject areas, such as ACCT for Accounting, COMP for Computing and SENG for Software Engineering. Your task is to write a program that can analyse a list of subject areas or other items in alphabetic order, and produce a specified number of sublist ranges to cover them all. The ranges must be chosen to minimise the difference between the number of items in the longest sublist and the number of items in the shortest list.

In addition, the following rules must be followed:

- The list of ranges must be in alphabetic order and must cover all items.
- Each range is either two different letters, in order and separated by a dash, or a single letter (so the singleton range M–M should be just M).
- A range must not start or end with a letter for which there are no matching items. For example, if no item starts with Q, the ranges L–Q and Q–T would both be invalid. However, not all letters included in the middle of a range need have items.

Data Format

The first line of input contains the number of items, a space, and the required number of ranges (between 1 and the number of items). Then follow the items, one per line, in alphabetic order. They may be in upper and/or lower case, but ranges should always use capitals. For example, to divide the names of the months into 4 ranges, the data looks like this:

```
12 4
April
August
December
February
January
July
June
March
May
November
October
September
```

The program should display on each output line one range and the number of items it represents. On the last line show the difference in lengths between the longest and shortest list.

For example, one of four solutions for this data is

```
A-F 4
J 3
M-N 3
O-S 2
Difference: 2
```

If there are multiple solutions (all with the same difference) any one will do.

Test Data

There are three data files to test your program with, besides the sample above (`task3sample.dat`).

File	Contains	Size	Number of ranges
`task3a.dat`	UNSW subject areas	148	7
`task3b.dat`	Names of some animals	181	6
`task3c.dat`	Complete list of IATA 3-letter airport codes	2799	9

Marking Scheme

7 marks for each of the three test cases, for a total of 21. Of the seven, 3 marks will be deducted if the ranges are not optimal, and a further mark if any of the other rules is violated.