Q1– Translating Text Messages (10 Marks)

Instant messaging (IM) and texting on portable devices has resulted in a set of common abbreviations useful for brief messages. However, some individuals may not understand these abbreviations.

Write a program that reads a one-line text message containing common abbreviations and translates the message into English using a set of translations stored in a file. For example, if the user enters the text message

y r u l8

the program should print

why are you late

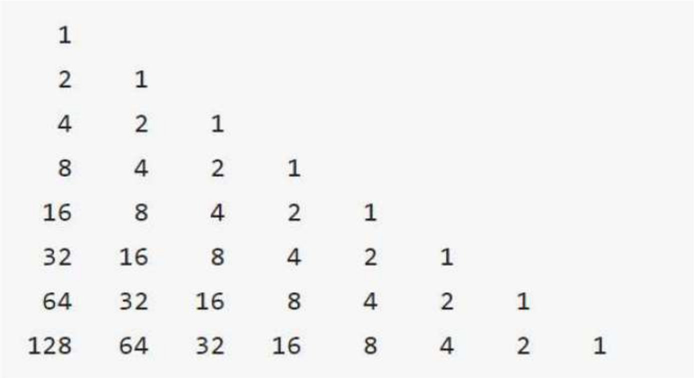
As a simplification, you can assume that are no punctuation marks.

Proceed as follows:

- **a.** Build a dictionary with abbreviations as keys and associated texts as values.
  Read the provided text file **abbreviations.txt**, each line of which contains an abbreviation and the associated text it. (You should display the list lines to see what a string of this list looks like.).
- **b.** Translate a message.
  Split the message into “words”. If a “word” is in the dictionary, replace it by the associated text. Otherwise simply copy the “word” to the translation.

Q2. Pattern (10 Marks)

Write a program in python to print the following pattern. You need to enter the number of rows from a user using input () function.



```
  1
  2    1
  4    2    1
  8    4    2    1
 16    8    4    2    1
 32   16    8    4    2    1
 64   32   16    8    4    2    1
128   64   32   16    8    4    2    1
```

Q3 (20 Marks)

a. Write a Python program that uses a while loop to ask the user to input 5 numbers. The program should then enter each of these numbers into a list.

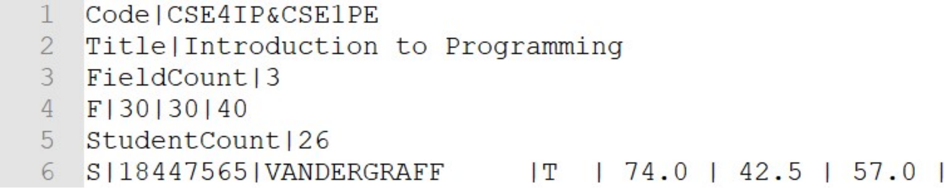
```
For example:
Enter a Number: 1
Enter a Number: 9
Enter a Number: 5
Enter a Number: 10
Enter a number: 11
[1, 9, 5, 10, 11]
```

b. Modify your program such that it outputs the list containing the square of each entered number.

```
For example:
Enter a Number: 1
Enter a Number: 3
Enter a Number: 6
Enter a Number: 2
Enter a number: 3
[1, 9, 36, 4, 9]
```

Q4 (60 marks)

Write a program to process the marks of the students enrolled in CSE4IP&CSE1PE. The marks are stored in a text file called **subject.txt** that has the format shown below.



```
Code|CSE4IP&CSE1PE
Title|Introduction to Programming
FieldCount|3
F|30|30|40
StudentCount|26
S|18447565|VANDERGRAFF     |T  | 74.0 | 42.5 | 57.0 |
```

- The first line identifies the subject code.
- The second line identifies the name of the subject.
- The third line, beginning with FieldCount, identifies the number of assessment components used in the subject.
- The fourth line, beginning with F, contains the percentage weight for each assessment component (e.g. the first component in the example is worth 20/100, the second 10/100 and the third 70/100).
- The fifth line, beginning with StudentCount, identifies how many students' marks the file contains.
- All the remaining lines each identify a student (and start with an S). A student's line has their student number, their family name, their initials, and then a mark for each assessment component. All component marks are out of 100. The tokens in each line are separated by the vertical bar ('|') symbol.

a. Write a program to read the file and store the data in various variables. The data about the students are to be stored in a list of tuples. For example, data of the first student are stored as

(18447565, VANDERGRAFF, T, [74.0, 42.5, 57.0])

where the marks are stored in a list. For the purpose of inspection, display the subject's general information, and the students' details. A student's detail can be displayed as a tuple, each student on one line.

b. Write a program to read the file and find the final mark for each student. The final mark can be rounded to the nearest integer. After that open a new file called **finalmarks.txt** to save the output. For example, data of the first student are stored as

Name final mark

c. Write a program to read the file and find min, max and average of the marks for each assessment. For example:

Assignment 1 Min Max Average
Assignment 2 Min Max Average
Final Min Max Average