Exercise 1: Most Frequent Letter *n*-gram [50 marks]

Problem Statement

A **letter *n*-gram** is a sequence of *n* letters. Letter *n*-grams with *n*=1 are called **unigrams** (or we normally just call them letters, *e.g.*, "p", "a" and "c"), while letter *n*-grams with *n*=2 are called **bigrams** (*e.g.,* "ae", "qb" and "ff").

Counting the frequencies of letter *n*-grams is very useful in cryptology. For example, if in a piece of cryptic English text, two letters frequently appear together, it is very likely that they correspond to "th" because "th" is the most frequently encountered bigram in English text.

In this exercise, you are to write a program **ngram.c** to find the most frequent letter unigram or bigram in an English text and report its frequency. The text comprises English words in lower case with no punctuation. All neighbouring words are separated by exactly one space.

For example, if the given text is "a friend in need is a friend indeed", the most frequent letter unigram is "e" which appears 6 times, while the most frequent letter bigram is "nd" which appears 3 times.

Do take note that for two letters to form a bigram, they must appear next to each other without any other letters or spaces in between them. In the previous example, the letters "a" and "f" never appears without any letters or spaces in between, therefore, bigram "af" does not appear in the text and its frequency is 0.

In the case where two unigrams or two bigrams have the same frequency, the one which appears earlier in alphabetical order should be reported.

Your program should read in a string which contains a sequence of English words in lowercase, and an integer, which is either 1 or 2. If the integer is 1, it should find and display the most frequent unigram with the frequency. If the integer is 2, it should find and display the most frequent bigram with the frequency.

You may assume that the input is valid and the maximum length of the string is 100 characters.

Write on the skeleton file **ngram.c** given to you. You must include the following two functions in your program:

- **`int mostFrequentUnigram(char text[], char result[])`**
  which takes in the string in the char array **`text`**, and returns the frequency of the most frequent unigram as an **`int`**. It should also return the unigram through the char array **`result`**.
- **`int mostFrequentBigram(char text[], char result[])`**
  which does the same as the previous function except that it is for bigram.

You may define additional functions as needed. Check sample runs for input and output format.

Sample Runs

Four sample runs are shown below with user input highlighted in **bold**.

```
Enter text: a friend in need is a friend indeed
Enter option: 1
Most frequent unigram: e
Frequency: 6
```

```
Enter text: a friend in need is a friend indeed
Enter option: 2
Most frequent bigram: nd
Frequency: 3
```

```
Enter text: mississippi is missing
Enter option: 1
Most frequent unigram: i
Frequency: 7
```

```
Enter text: mississippi is missing
Enter option: 2
Most frequent bigram: is
Frequency: 4
```