The following materials have been collected from the numerous sources such as Stanford CS106 and Harvard CS50 including my own and my students over the years of teaching and experiences of programming. Please help me to keep this tutorial up-to-date by reporting any issues or questions. Please send any comments or criticisms to idebtor@gmail.com. Your assistances and comments will be appreciated.

Midterm 1 – Dial

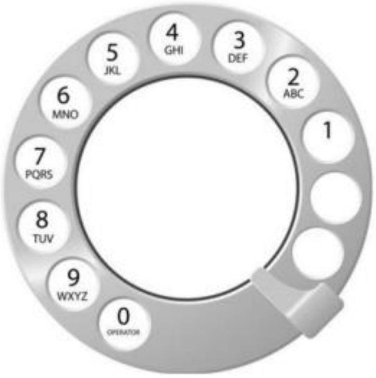
Table of Contents

Files provided

dial.pdf	this file
dial800.cpp	a skeleton code
dial800x.exe	a sample solution
dialmap.cpp	a skeleton code
dialmapx.exe	a sample solution

Problem 1 – Dial800

My grandma still uses an ancient pulse dial telephone with a rotary dial as shown in the following picture:

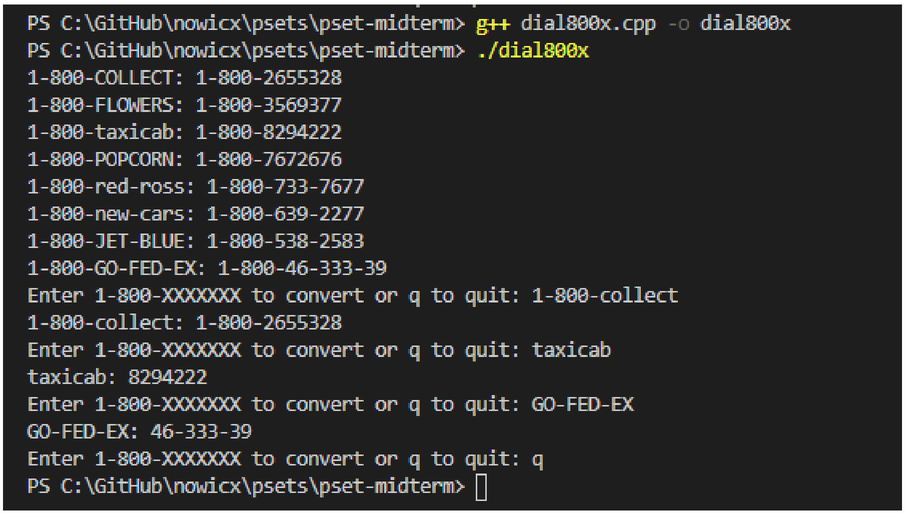


She still remembers many 1-800 toll-free numbers such as 1-800-COLLECT, 1-800-FLOWERS, and 1-800-GO-FED-EX instead of 1-800-2655328, 1-800-3569377, and 1-800-46-333-39. Unfortunately, some phones do not have numbers with alphabets. My job is to convert 1-800-NUMBERS into numeric ones.

Write a function called, `dialnum()`, that takes my grandma's number and returns a numeric number in place. For example, it takes 1-800-COLLECT and returns 1-800-2655328.

- Use a skeleton file, dial800.cpp provided and complete the function, `dialnum()`.
- Ignore all the digits and non-numeric symbols as they are.
- Convert alphabets from a to z (or A to Z) into digit in ASCII.
- Notice that some of 1-800-NUMBERS are shown for your testing.

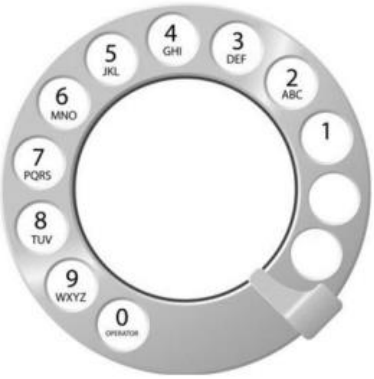
Sample Run:



```
PS C:\GitHub\nowicx\psets\pset-midterm> g++ dial800x.cpp -o dial800x
PS C:\GitHub\nowicx\psets\pset-midterm> ./dial800x
1-800-COLLECT: 1-800-2655328
1-800-FLOWERS: 1-800-3569377
1-800-taxicab: 1-800-8294222
1-800-POPCORN: 1-800-7672676
1-800-red-ross: 1-800-733-7677
1-800-new-cars: 1-800-639-2277
1-800-JET-BLUE: 1-800-538-2583
1-800-GO-FED-EX: 1-800-46-333-39
Enter 1-800-XXXXXXX to convert or q to quit: 1-800-collect
1-800-collect: 1-800-2655328
Enter 1-800-XXXXXXX to convert or q to quit: taxicab
taxicab: 8294222
Enter 1-800-XXXXXXX to convert or q to quit: GO-FED-EX
GO-FED-EX: 46-333-39
Enter 1-800-XXXXXXX to convert or q to quit: q
PS C:\GitHub\nowicx\psets\pset-midterm> 
```

Problem 2 – DialMap

My grandma still uses an ancient pulse dial telephone with a rotary dial as shown in the following picture:



For each digit that we want to dial, we need to turn the rotary dial clockwise until the chosen digit reaches the finger stop (metal fin). Then we let go of the dial and wait for it to return to its original position before we can dial another digit.

In our modern, instant gratification world, the dial return often lasts much longer than our patience. More precisely, dialing the digit 1 takes a total of two seconds, while dialing any larger digit takes an additional second for each additional finger circle counting from 1 to the dialed digit (as shown in the picture).

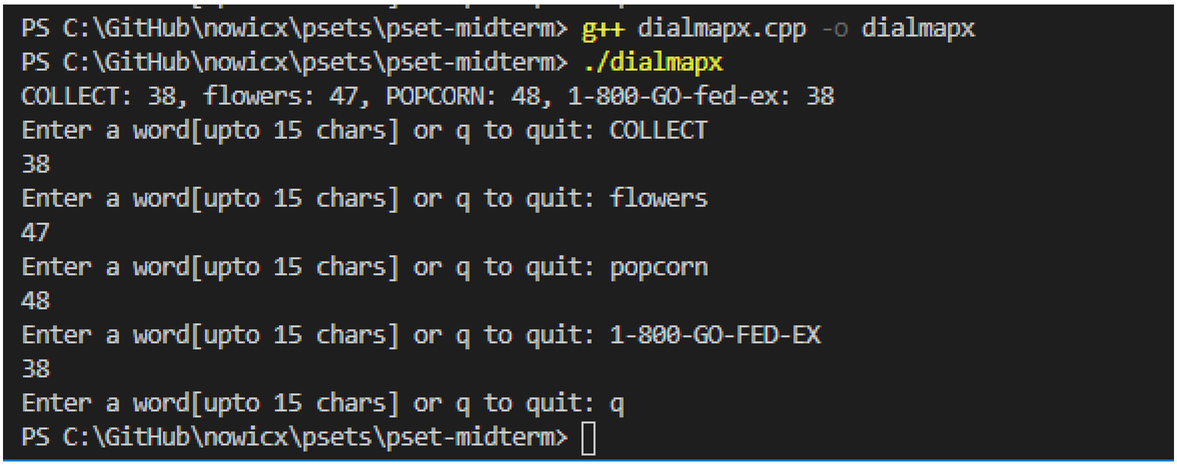
My grandma remembers phone numbers by memorizing a corresponding word which, when dialed, results in the correct number being dialed. When dialing a word, for each letter, we dial the digit which has that letter written next to it on the dial (for example, the digit 7 for the letter S).

For example, the word COLLECT corresponds to the number 2655328. Your task is determining, for a given word, the total time required to dial that word.

- Input: The input repeatedly contains a single word consisting of between 2 and 15 (inclusive) uppercase or lowercase English letters. Ignore other characters such as digits, dash and others if any.
- Output: The first and only line of output must contain the required **dialing time**.
- No skeleton file is provided.
- Use STL map container. Otherwise you may get only one half of the full credit for this problem.

NOTE: There are many different ways to approach this problem. Most preferred method to use this problem is STL map class since this problem is to review STL map function we learned before. If you use other methods such as a lot of case statements or arrays etc., you may still get one half of the full credit.

Sample Run:



```
PS C:\GitHub\nowicx\psets\pset-midterm> g++ dialmapx.cpp -o dialmapx
PS C:\GitHub\nowicx\psets\pset-midterm> ./dialmapx
COLLECT: 38, flowers: 47, POPCORN: 48, 1-800-GO-fed-ex: 38
Enter a word[upto 15 chars] or q to quit: COLLECT
38
Enter a word[upto 15 chars] or q to quit: flowers
47
Enter a word[upto 15 chars] or q to quit: popcorn
48
Enter a word[upto 15 chars] or q to quit: 1-800-GO-FED-EX
38
Enter a word[upto 15 chars] or q to quit: q
PS C:\GitHub\nowicx\psets\pset-midterm>
```

Submitting your solution

- **On my honour, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.**
  Signed: _____________ Section: _______ Student Number: _________________
- Make sure your code **compiles** and **runs** right before you submit it. Don't make "a tiny last-minute change" and assume your code still compiles. You will not receive sympathy for code that "almost" works.
- If you only manage to work out the Project problem partially before the deadline, you still need to turn it in. However, don't turn it in if it does not compile and run.
- Place your source files in the folder you and I are sharing.
- After submitting, if you realize one of your programs is flawed, you may fix it and submit again as long as it is **before the deadline**. You will have to resubmit any related files together, even if you only change one. You may submit as often as you like. **Only the last version** you submit before the deadline will be graded.

Files to submit and Grade

Submit the following files on Piazza folder.

- **dial800.cpp**
- **dialmap.cpp**

Due

- Due: 11:55 pm