1 Assignment 4 Introduction

This assignment is due *before* 11:59pm on the listed date. Which means, submissions made *on or after* 11:59pm will be counted as a late submissions. Late submissions *before* 11:59pm on the following day will receive a 25% point deduction as penalty. Submissions made *on or after* 11:59pm on the same day will *not* be graded. We strongly recommend completing the assignment before then.

Assignments due on Mondays generally involve materials covered in lectures from the previous week, whereas assignments due on Thursdays involve materials covered in lectures from the running week. So be sure to watch the lectures and go over the reading materials before attempting the assignments.

Good luck!

2 Word Scrambler (50 points)

Write a program that generates obfuscated texts by jumbling the middle letters of words. You will do this by reading from standard input a sequence whitespace separated lowercase-only words. Your program should output these words (8 words per line separated by spaces) scrambled. A scrambled word is a word that has its first and last letters unchanged, but has its middle letters reversed. In your solution, you must use a `void` function named `ScrambleWord`, that takes in a string, and scrambles it based on the problem specification.

Example Input:

```
typoglycemia a portmanteau of typo and hypoglycemia is a neologism for a purported
discovery about the cognitive processes involved in reading text the principle is
that readers can comprehend text despite spelling errors and misplaced letters in
the words it is an urban legend and internet meme that only appears to be correct
```

Expected Output:

```
timecylgopya a paetnamtrou of tpyo and himecylgopya is
a nsigoloem for a petroprud drevocsiy auobt the
cvitingoe pessecors ievlovnd in rnidaeg txet the plpicnire
is taht rredaes can cneherpmod txet dtipsee snillepg
erorrs and mecalpsid lrettes in the wdros it
is an uabrn lneged and ienretnt mmee taht
olny araepps to be ccerrot
```

Note: all words should be followed by a space, unless they are the 8th word on the line.
General Instructions:

- When writing your program, you may only use concepts you have learned in the course thus far.
- Your program should compile and run.
- The formatting must match exactly as shown in the example above, including whitespaces.

3 ASCII Neighbors (50 points)

Write a program that reads from standard input a sequence of whitespace-separated 'words'. The program should output each word on its own line, but only including the characters that are equal or adjacent in the ASCII table. For example, the character 'i' has an ASCII value of 105, and it should be included in the output only if it is next to an h (104), an i (105), or a j (106). In your solution, you must use a `void` function named `FilterNeighbors`, that takes in a string, and filters it based on the problem specification.

Example Input:

```
ahidkkrtsraaav
```

Expected Output:

```
hikktsraaa
```

The first 'a' isn't in the output because its only neighbor (h) is not adjacent to it in the ASCII table. The 'h' and the 'i' are adjacent in the table so they are outputted. The 'k's are equal to each other, so they are also outputted. And so on.

Example Input:

```
23321 gbsyfbjlbsadfoebw ./-$$!&%aAJ
{|}:<:>=k?
```

Expected Output:

```
23321

./$$&%
{|}>=
```

Remember that there are lots of characters in ASCII beyond just letters. The "gbsyfbjlbsadfoebw" word contains to letters that are adjacent in the word and in the table, so its line is empty.

- When writing your program, you may only use concepts you have learned in the course thus far.
- Your program should compile and run.
- The formatting must match exactly as shown in the example above, including whitespaces.