

Project 6: Subwords

CS 2370

Background

The purpose of this assignment is to use the *containers* and *algorithms* in the C++ standard library to make short work of data processing.

You may have played word games where you are given a word and are asked to find all subwords of the given word by mixing up subsets of the letters of the original word. This is how apps such as Text Twist and Word Scapes work.

Requirements

You will use a word passed in to the program, or choose a word at random from a collection of 81,484 words found in the file *words.txt*, provided for you on Canvas. You will then find all possible *subwords* using the characters of the chosen string of length 3 through the length of the word. Here is a sample run:

```
$ words
The word is sodiums:
```

```
dim
dis
dos
duo
ids
ism
mid
mis
mod
mud
mus
oms
sis
sod
sos
sou
sum
dms
diss
duos
isms
miss
mods
moss
muds
muss
sods
sous
suds
sumo
sums
misdo
odium
sumos
```

```
odiums
sodium
sodiums
```

Note that the words must appear grouped by length, smallest to largest, and the words in each length-group are sorted alphabetically. This can be done with a single call to sort using a suitable comparator function. Every time you run the program you should get a different word. For simplicity, the original word is part of the list. There could be multiple words of the maximum length:

The word is rehearings:

```
age
air
ani
are
ash
ear
ens
...
shearing
sheering
garnishee
rehearing
rehearings
rehearsing
```

Implementation Notes

Read all words into a vector. Then choose a random integer between 0 and NWORDS, where NWORDS = 81483. You can do this with the following code:

```
default_random_engine dre(time(nullptr));           // Seed the engine
uniform_int_distribution<int> di(0,NWORDS);          // Use uniform distribution
int n = di(dre);                                     // Get a random int
```

You need to include `<random>` and `<ctime>` for this code to work. You now check to see if the word in position `n` in your vector is of length 3 or greater. As soon as you get such a word, the fun begins!

SPECIFICATION:

The program may take one command-line argument, the word to use. If the word is at least 3 characters long, use it. If there is no command-line argument, or it is less than 3 characters long, select a random word as described above.

- How will you check words in the dictionary to see if they are arrangements of the letters of the original word?
- How will you sort your vector by groups?
- With white space and comments, my program is less than 50 lines.

Have fun!