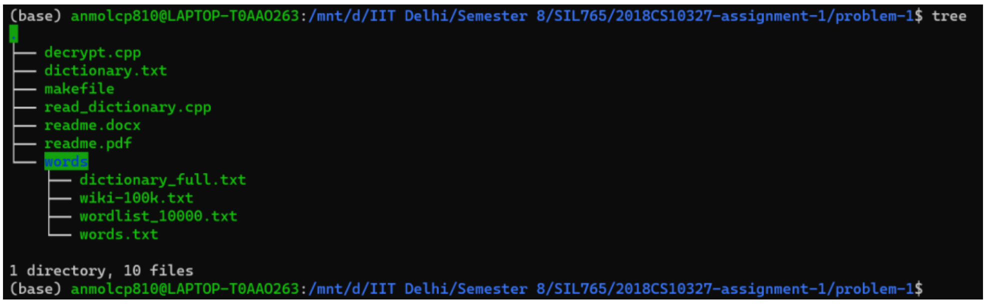
Readme

By: Anmol Agarwal (2018CS10327)

Directory Structure:



```
(base) anmolcp810@LAPTOP-T0AA0263:/mnt/d/IIT Delhi/Semester 8/SIL765/2018CS10327-assignment-1/problem-1$ tree
.
├── decrypt.cpp
├── dictionary.txt
├── makefile
├── read_dictionary.cpp
├── readme.docx
├── readme.pdf
└── words
    ├── dictionary_full.txt
    ├── wiki-100k.txt
    ├── wordlist_10000.txt
    └── words.txt

1 directory, 10 files
(base) anmolcp810@LAPTOP-T0AA0263:/mnt/d/IIT Delhi/Semester 8/SIL765/2018CS10327-assignment-1/problem-1$
```

How to Run: *(please run "make" commands inside **problem-1** folder.)*

decryptText: make decryptText file="name of file.txt"

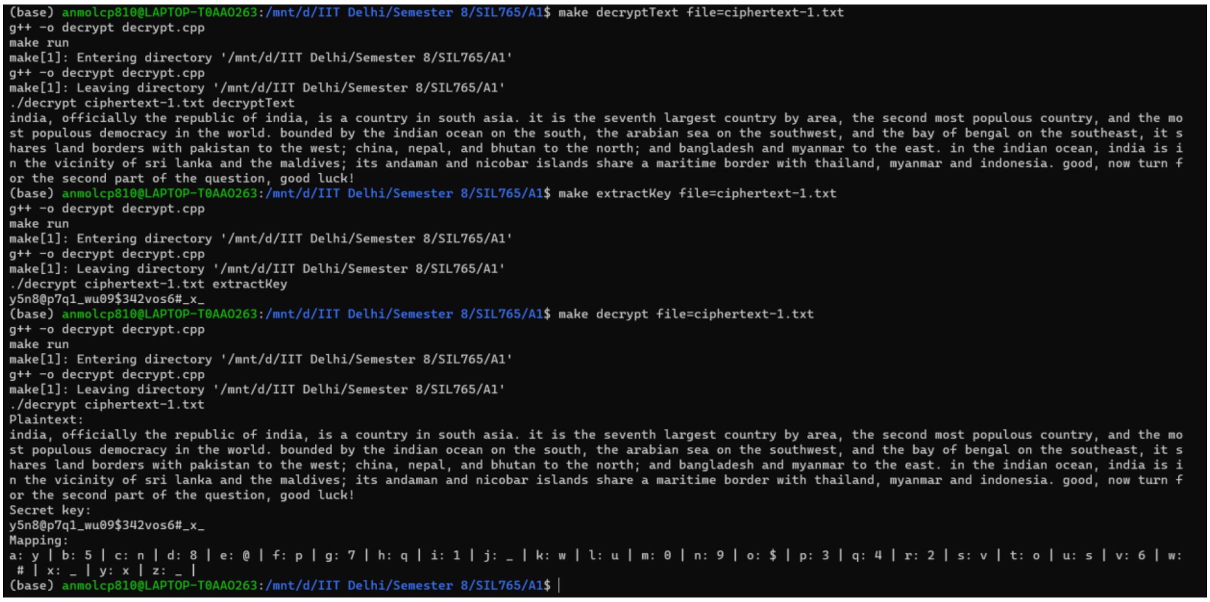
extractKey: make extractKey file="name of file.txt"

decrypt: make decrypt file="name of file.txt"

Details:

1. *decryptText:* gives plaintext.
2. *extractKey:* gives secret key.
3. *decrypt:* gives plaintext, secret key and mapping.

Screenshot Demo:



```
(base) anmolcp810@LAPTOP-T0AA0263:/mnt/d/IIT Delhi/Semester 8/SIL765/A1$ make decryptText file=ciphertext-1.txt
g++ -o decrypt decrypt.cpp
make run
make[1]: Entering directory '/mnt/d/IIT Delhi/Semester 8/SIL765/A1'
g++ -o decrypt decrypt.cpp
make[1]: Leaving directory '/mnt/d/IIT Delhi/Semester 8/SIL765/A1'
./decrypt ciphertext-1.txt decryptText
india, officially the republic of india, is a country in south asia. it is the seventh largest country by area, the second most populous country, and the mo
st populous democracy in the world. bounded by the indian ocean on the south, the arabian sea on the southwest, and the bay of bengal on the southeast, it s
hares land borders with pakistan to the west; china, nepal, and bhutan to the north; and bangladesh and myanmar to the east. in the indian ocean, india is i
n the vicinity of sri lanka and the maldives; its andaman and nicobar islands share a maritime border with thailand, myanmar and indonesia. good, now turn f
or the second part of the question, good luck!
(base) anmolcp810@LAPTOP-T0AA0263:/mnt/d/IIT Delhi/Semester 8/SIL765/A1$ make extractKey file=ciphertext-1.txt
g++ -o decrypt decrypt.cpp
make run
make[1]: Entering directory '/mnt/d/IIT Delhi/Semester 8/SIL765/A1'
g++ -o decrypt decrypt.cpp
make[1]: Leaving directory '/mnt/d/IIT Delhi/Semester 8/SIL765/A1'
./decrypt ciphertext-1.txt extractKey
y5n8@p7q1_wu09$342vos6#_x_
(base) anmolcp810@LAPTOP-T0AA0263:/mnt/d/IIT Delhi/Semester 8/SIL765/A1$ make decrypt file=ciphertext-1.txt
g++ -o decrypt decrypt.cpp
make run
make[1]: Entering directory '/mnt/d/IIT Delhi/Semester 8/SIL765/A1'
g++ -o decrypt decrypt.cpp
make[1]: Leaving directory '/mnt/d/IIT Delhi/Semester 8/SIL765/A1'
./decrypt ciphertext-1.txt
Plaintext:
india, officially the republic of india, is a country in south asia. it is the seventh largest country by area, the second most populous country, and the mo
st populous democracy in the world. bounded by the indian ocean on the south, the arabian sea on the southwest, and the bay of bengal on the southeast, it s
hares land borders with pakistan to the west; china, nepal, and bhutan to the north; and bangladesh and myanmar to the east. in the indian ocean, india is i
n the vicinity of sri lanka and the maldives; its andaman and nicobar islands share a maritime border with thailand, myanmar and indonesia. good, now turn f
or the second part of the question, good luck!
Secret key:
y5n8@p7q1_wu09$342vos6#_x_
Mapping:
a: y | b: 5 | c: n | d: 8 | e: @ | f: p | g: 7 | h: q | i: 1 | j: _ | k: w | l: u | m: 0 | n: 9 | o: $ | p: 3 | q: 4 | r: 2 | s: v | t: o | u: s | v: 6 | w:
 # | x: _ | y: x | z: _ |
(base) anmolcp810@LAPTOP-T0AA0263:/mnt/d/IIT Delhi/Semester 8/SIL765/A1$ |
```

Approach used for Cracking Cipher:

Thought Process

1. Firstly, I tried using frequency analysis, but noticed that most frequent ciphers are not necessarily most frequent characters in English language. So mere frequency analysis might not work.
2. Then I tried looking for single letter words like ‘a’ and ‘i’. The higher frequency letter was substituted as ‘a’ and then another as ‘i' (if present).
3. Then among three letter words, ‘the’ is most common followed by ‘and’. So, I substituted two most frequent words by “the” and “and” respectively.
4. Finally, I looked for other words like ‘r’ through ‘there, here, their’. For ‘o’, and ‘n’, “on” and “no” are the only two letter words which are palindromic, so I checked for that.
5. Then I checked for “for” (through presence of ‘r’ or “for” is the most frequent three letter word appearing before “the”), “of”, “if”, “is”, “ing”.

Dictionary

1. Doing all this analysis above, we can find 8-10 letters, but to find other letters we would need dictionary.
2. I tried various dictionaries, and finally picked two dictionaries and got combined dictionary:
    a. Gutenberg Project (The Project Gutenberg eBook of Webster's Unabridged Dictionary, by Various)
    b. /usr/share/dict/words : available in ubuntu machine.
3. Using dictionary, I was generally able to find all the words.

Final Solution

1. I have made two solvers; one uses only dictionary and another both frequency analysis and then dictionary.
2. First solver may not necessarily find words to substitute, so we may still need frequency analysis to get started.

More Details on how Dictionary Works:

1. I created a trie consisting of all words in dictionary.

2. I then ran a loop for each partially/not decrypted word in ciphertext, tried to match correct word for it in dictionary.
3. If matched words are only 1, then we use only that word to substitute.
4. If matched words are 0 or greater than 1, then we ignore the words for now.
5. Program will keep going through ciphertext again and again, until no word can be substituted, or all words are substituted.
6. **Match:** In match, I not only checked whether plaintext letters match at each position, *but there is no many-to-one mapping between ciphers and plain letters*. This helped in removing many words in search space.

Drawbacks of Final Solution:

1. Correctness of plaintext depends heavily on dictionary. The moment we substitute a single letter wrong, then whole algorithm is going to break.
2. Building very large dictionary is also not helpful because matched words could never be one in that case.

Output Format:

When printing secret key, I have replaced letters with 0 frequencies with _ symbol.

Result

ciphertext-1.txt

Plaintext:

india, officially the republic of india, is a country in south asia. it is the seventh largest country by area, the second most populous country, and the most populous democracy in the world. bounded by the indian ocean on the south, the arabian sea on the southwest, and the bay of bengal on the southeast, it shares land borders with pakistan to the west; china, nepal, and bhutan to the north; and bangladesh and myanmar to the east. in the indian ocean, india is in the vicinity of sri lanka and the maldives; its andaman and nicobar islands share a maritime border with thailand, myanmar and indonesia. good, now turn for the second part of the question, good luck!

Secret key:

y5n8@p7q1_wu09$342vos6#_x_

Mapping

a: y	b: 5	c: n	d: 8	e: @	f: p	g: 7	h: q	i: 1
j: NA	k: w	l: u	m: 0	n: 9	o: $	p: 3	q: 4	r: 2
s: v	t: o	u: s	v: 6	w: #	x: NA	y: x	z: NA	

ciphertext-2.txt

Plaintext:

defeated and leaving his dinner untouched, he went to bed. that night he did not sleep well, having feverish dreams, having no rest. he was unsure whether he was asleep or dreaming. conscious, unconscious, all was a blur. he remembered crying, wishing, hoping, begging, even laughing. he floated through the universe, seeing stars, planets, seeing earth, all but himself. when he looked down, trying to see his body, there was nothing. it was just that he was there, but he could not feel anything for just his presence.

Secret key:

8ot64spnrxzqwy$1_3vu205_#_

Mapping

a: 8	b: o	c: t	d: 6	e: 4	f: s	g: p	h: n	i: r
j: x	k: z	l: q	m: w	n: y	o: $	p: 1	q: NA	r: 3
s: v	t: u	u: 2	v: 0	w: 5	x: NA	y: #	z: NA	

example-ciphertext.txt

Plaintext:

charizard was designed by atsuko nishida for the first generation of pocket monsters games red and green, which were localized outside japan as pokemon red and blue. charizard was designed before charmander, the latter being actually based on the former. originally called lizardon in japanese, nintendo decided to give the various pokemon species clever and descriptive names related to their appearance or features when translating the game for western audiences as a means to make the characters more relatable to american children. as a result, they were renamed charizard, a portmanteau of the words charcoal or char and lizard.

Secret key:

y8s@z051$n3ruv#9_q4w7p6_ot

Mapping

a: y	b: 8	c: s	d: @	e: z	f: 0	g: 5	h: 1	i: $
j: n	k: 3	l: r	m: u	n: v	o: #	p: 9	q: NA	r: q
s: 4	t: w	u: 7	v: p	w: 6	x: NA	y: o	z: t	

Additional Files

1. dictionary.txt
2. read_dictionary.cpp : used to read dictionary files (dictionary_full.txt, words.txt) and create dictionary.txt.
3. words: contains various dictionaries that I tried.