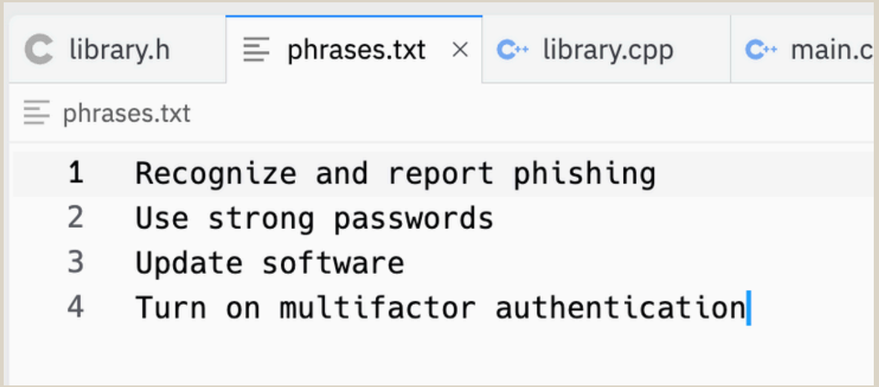
Encryption Orchestra

Algorithm Angels

Brielle McBarron, Ida Behrouz Vazeri, Jasmine van Zoest, June Phillips, Natalia Negrete, Sofia Tamayo

Messages to Encrypt

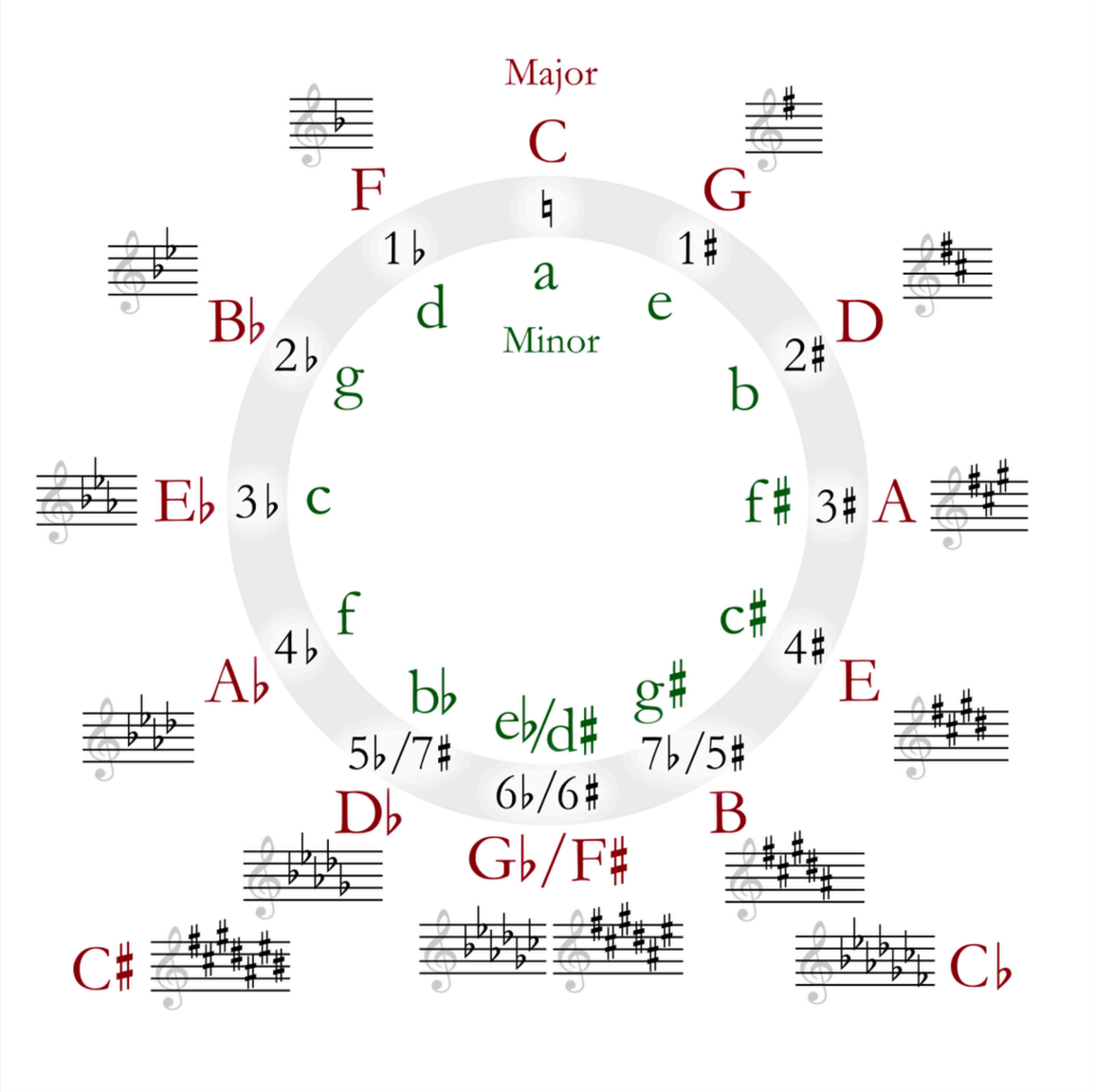


1. regonize and report phishing
2. use strong passwords
3. update software
4. turn on multifactor authentication

Encryption Rules

1. Accidentals (flats or sharps) will depend on the first letter of the phrase
    a. if the first letter of the first word in the phrase starts with the letters: A, B, C, D, E, F, G, H, I, J, K, L, M --> it will use flats
    b. if the first letter of the first word in the phrase starts with the letters: N, O, P, Q, R, S, T, U, V, W, X, Y, Z --> it will use sharps
2. The number of words in each phrase will determine the number of accidentals in the key signature
    a. if a phrase has 2 total words, it will have 2 accidentals
3. The alphabet will be repeated with each note representing a musical note A-G
    a. varying based on its placement in the alphabet
4. Each space will represent a rest

Key Signature Circle



Code

```
using namespace std;

// collect one finds put the scentence into an array.
it finds the number
// of words in the array, and the first letter of the

int num_rows;
void collect_one(ifstream &fin, Data &D) {
  string sentence,key;
  getline(fin, sentence, '.');
  D.length = sentence.length();
  for (int i = 0; i < sentence.length(); i++) {
    fin >> D.sentence[i];
  }
  for (int i = 0; i < sentence.length(); i++) {
    if (D.sentence[i] == " ") {
      D.number_of_words++;
    }
  } // end of for
  D.firstletter = D.sentence[0];
```

```
      D.number_of_words++;
    }
  } // end of for
  D.firstletter = D.sentence[0];
   key=findkey(D);
  numberofaccidentals(D,key,D.number_of_words);
  fullkeysignature(D, D.major, D.accidentals);



  convert_partA(D);
  convert_partB(D);
} // end of collect_one

void collect_all(ifstream &fin, Data D[]) {
  num_rows = 0;
  while (num_rows < MAX) {
    collect_one(fin, D[num_rows]);
    num_rows++;
  } // while not at the end of file, end of while
} // end of collect all

string findkey(Data &D) {
  string firstletter = D.firstletter;
  bool flats = false;
  bool sharps = false;
  string answer;

  if (firstletter == "A" || firstletter == "B" ||
firstletter == "C" ||
      firstletter == "D" || firstletter == "E" ||
firstletter == "F" ||
      firstletter == "G" || firstletter == "H" ||
firstletter == "I" ||
      firstletter == "J" || firstletter == "K" ||
firstletter == "L" ||
      firstletter == "M") {
    flats = true;
    sharps = false;
  } else {
    flats = false;
    sharps = true;
  }


  if( flats ==true){
    answer = "flats";
  }else{
     answer = "sharps";
  }
  // this finds whether or not the key signature is flats
or sharps
  // once we find this, we will return it and use this to
find the overall key
  // signture
  return answer;
}
```

AI {✓} C++ Ln 36, Col 20 ● Spaces: 2 Histor

Run

README.txt | library.h | phrases.txt | library.cpp | m

library.h

```
21
22
23  void collect_one(ifstream &fin, Data &D);
24  void collect_all(ifstream &fin, Data D[]);
25  char findmajor(ifstream &fin);
26  string findkey(Data &D);
27  int numberofaccidentals(Data &D, string keysignature, int
    numberofwordsinphrase);
28  string fullkeysignature(Data &D, string keysignature, int
    accidentals);
29  void convert_partA(Data &D);
30  void convert_partB(Data &D);
31  void display(Data P[], int p);
32
33  #endif
```

library.h

```
1   #ifndef LIBRARY_H
2   #define LIBRARY_H
3   #include <iomanip>
4   #include <iostream>
5   #include <fstream>
6
7   using namespace std;
8
9
10  const int MAX= 900;
11  struct Data{
12  string sentence[MAX];
13  string firstletter;
14  string major;
15  int number_of_words=0;
16  int length;
17  string length_of_note[MAX];
18  string note[MAX];
19  int accidentals;
20  };
21
22
23  void collect_one(ifstream &fin, Data &D);
24  void collect_all(ifstream &fin, Data D[]);
25  char findmajor(ifstream &fin);
26  string findkey(Data &D);
```

AI {✓} C Ln 1, Col 1 Spaces: 2

```
155
156 void convert_partA(Data &D) {
157   for (int i = 0; i < D.length; i++) {
158     if (((D.sentence[i] >= "a") && (D.sentence[i] <= "g"))
    ||
159         ((D.sentence[i] >= "A") && (D.sentence[i] <= "G")))
    {
160       D.length_of_note[i] = "quater note";
161     } else if (((D.sentence[i] >= "h") && (D.sentence[i] <=
    "n")) ||
162                ((D.sentence[i] >= "H") && (D.sentence[i] <=
    "N"))) {
163       D.length_of_note[i] = "eighth note";
164     } else if (((D.sentence[i] >= "o") && (D.sentence[i] <=
    "u")) ||
165                ((D.sentence[i] >= "O") && (D.sentence[i] <=
    "U"))) {
166       D.length_of_note[i] = "sixteenth note";
167     } else if (((D.sentence[i] == "v") || (D.sentence[i] ==
    "z")) ||
168                ((D.sentence[i] == "V") || (D.sentence[i] ==
    "Z"))) {
169       D.length_of_note[i] = "thirty second note";
170     } else { // if the letter is a space, it is a rest
171       D.length_of_note[i] = "rest";
172     }
173   }
174 }
175
176 void convert_partB(Data &D) {
177   for (int i = 0; i < D.length; i++) {
```

```
227                (D.major == "B major") || (D.major == "F#
    major") ||
228                (D.major == "C# major")) {
229       D.note[i] = "D#";
230     } else {
231       D.note[i] = "D";
232     }
233   } else if ((D.sentence[i] == "e") || (D.sentence[i] ==
    "E") ||
234              (D.sentence[i] == "l") || (D.sentence[i] ==
    "L") ||
235              (D.sentence[i] == "s") || (D.sentence[i] ==
    "S")) {
236     if ((D.major == "Eb major") || (D.major == "Ab
    major") ||
237         (D.major == "Db major") || (D.major == "Gb
    major") ||
238         (D.major == "Cb major")) {
239       D.note[i] = "Eb";
240     } else if ((D.major == "E major") || (D.major == "B
    major") ||
241                (D.major == "F# major") || (D.major == "C#
    major")) {
242       D.note[i] = "E#";
243     } else {
244       D.note[i] = "E";
245     }
246   } else if ((D.sentence[i] == "f") || (D.sentence[i] ==
    "F") ||
247              (D.sentence[i] == "m") || (D.sentence[i] ==
```

update software

↳ 4 = sharps

↳ 2 words in the phrase: 2 accidentals

2 sharps = D major

a b c d e f g h i j k l m n o p q r s t u v w x y z

a b c d e f g a b c d e f g a b c d e f g a b c d e

quarter note | eighth note | sixteenth note | thirty-second note

use strong passwords

g c e c f d a g g b a c c b a d d e

1/16 1/8 1/4 1/2 1/8 1/8 1/8 1/8 1/4 1/2 8th 1/2 1/4 1/8 1/16 1/8 1/8 1/8 1/2 1/8

update software

a b d a f e e a f f b a d c

1/16 1/8 1/4 1/2 1/8 1/2 1/8 1/4 1/8 1/16 1/2 1/8 1/2

turn on multifactor authentication

f a d g a g f a e f b f a c c g d a g e a e g f b e a f b a g

1/8 1/16 1/8 1/4 1/8 1/4 1/4 1/16 1/16 1/8 1/8 1/4 1/2 1/2 1/2 1/4 1/8 1/8 1/2 1/8 1/4 1/4 1/2 1/4 1/8 1/16 1/16 1/2 1/8 1/4 1/4 1/8 1/4

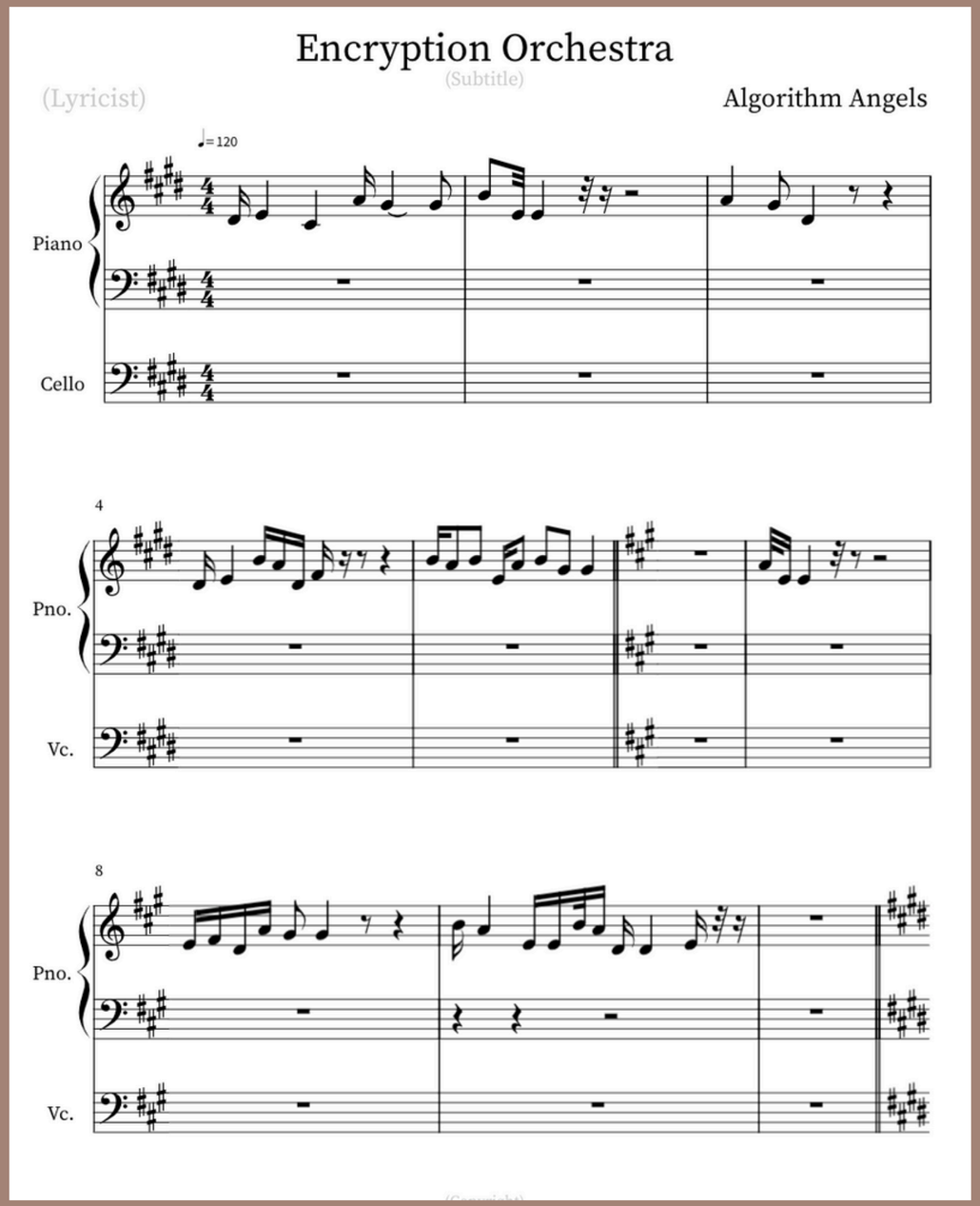
create music.

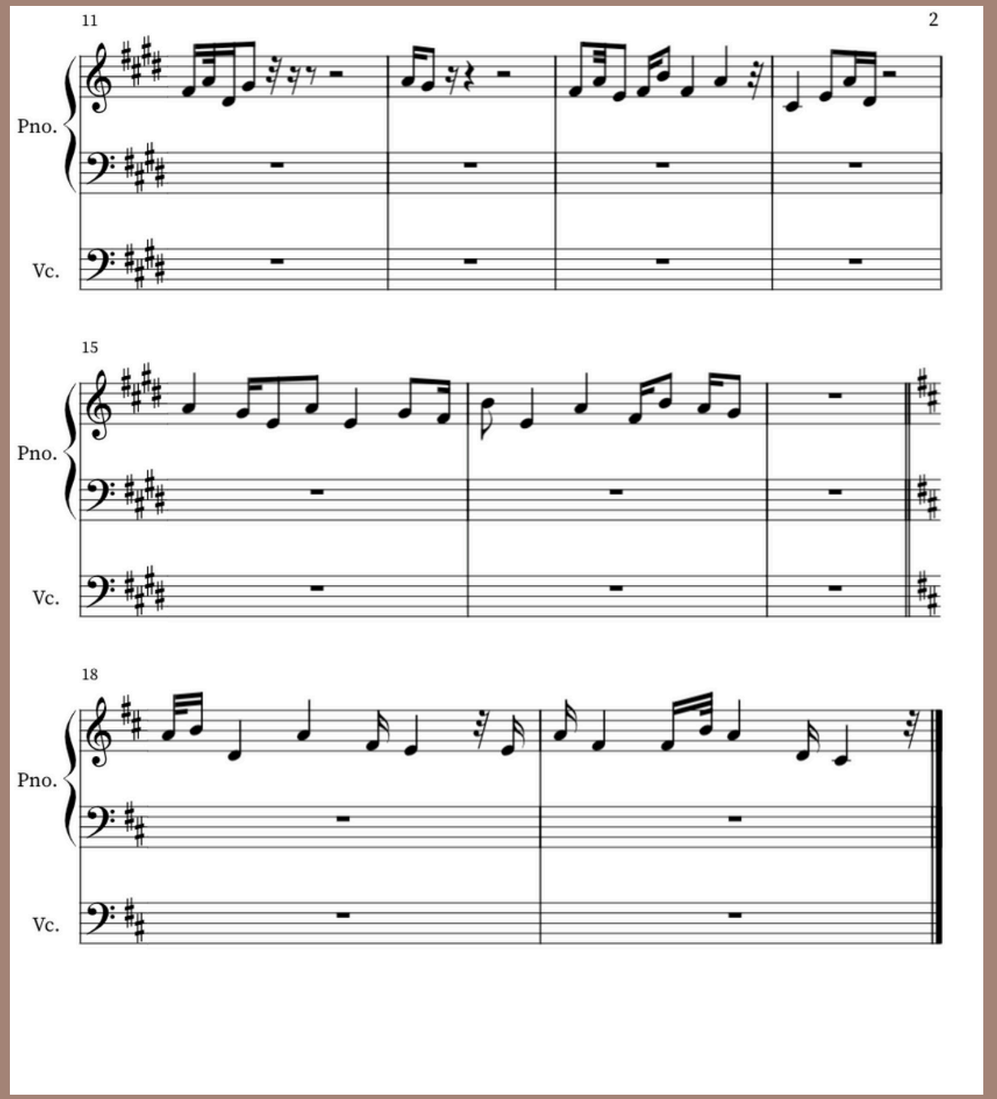
Encryption Orchestra

(Subtitle)

(Lyricist)

Algorithm Angels




11
2
Pno.
Vc.
15
Pno.
Vc.
18
Pno.
Vc.

Encryption Orchestra

Thank You For Listening