CMPT365 Programming Assignment 2

Junchen Li

301385486

2021/4/7

Environment:

For the second programming assignment, I chose the java as my coding language. Running in the IntelliJ IDEA CE. For the first question I use the libraries as follow:

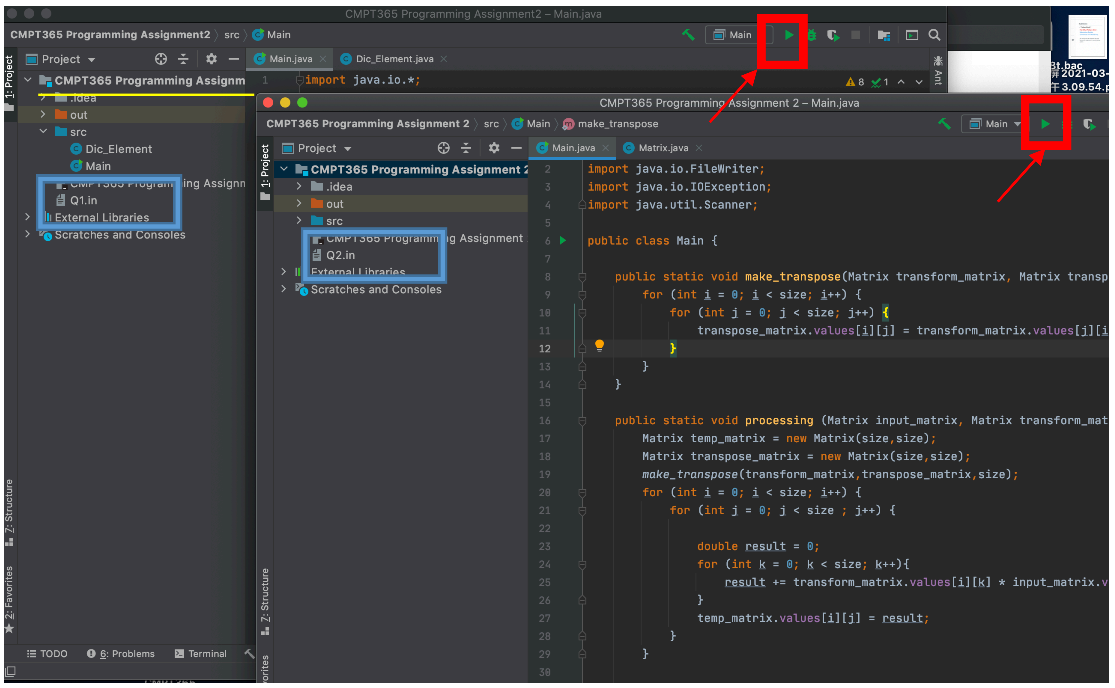
```
import java.io.*; This is for reading the file and writing into the file
import java.util.ArrayList;  Using the array list structure for storing data
import java.util.Scanner;  Scanner is used for reading in the file
```

For the question 2, the libraries as follow:

```
import java.io.FileReader; For reading the file
import java.io.FileWriter; For writing into the file
import java.io.IOException; Throw an exception if not found file
import java.util.Scanner;  Scanner is used for reading in the file
```

Actually the first three lines in the second part are belongs to the java.io.*. I expand them in order to showing which exactly function I used.

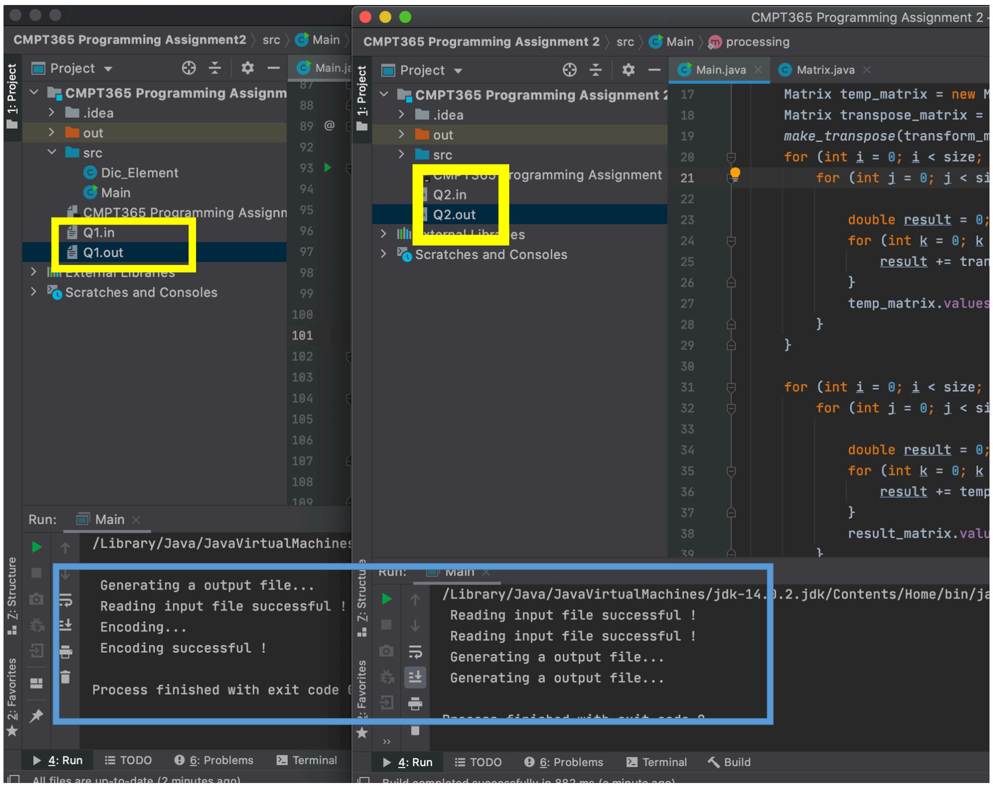
Operate:


CMPT365 Programming Assignment2 – Main.java
CMPT365 Programming Assignment2 〉 src 〉 Main
Project
Main.java
Dic_Element.java
CMPT365 Programming Assignm
.idea
out
src
Dic_Element
Main
CMPT365 Programming Assignm
Q1.in
External Libraries
Scratches and Consoles
CMPT365 Programming Assignment 2 – Main.java
CMPT365 Programming Assignment 2 〉 src 〉 Main 〉 make_transpose
Main.java
Matrix.java
CMPT365 Programming Assignment 2
Q2.in
TODO
6: Problems
Terminal

```
import java.io.*;
```

```
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class Main {

    public static void make_transpose(Matrix transform_matrix, Matrix transp
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                transpose_matrix.values[i][j] = transform_matrix.values[j][i
            }
        }
    }

    public static void processing (Matrix input_matrix, Matrix transform_mat
        Matrix temp_matrix = new Matrix(size,size);
        Matrix transpose_matrix = new Matrix(size,size);
        make_transpose(transform_matrix,transpose_matrix,size);
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size ; j++) {

                double result = 0;
                for (int k = 0; k < size; k++){
                    result += transform_matrix.values[i][k] * input_matrix.v
                }
                temp_matrix.values[i][j] = result;
            }
        }

```

Two blue square show the order of question. Q1.in means the first question and Q2.in means second question in the above screenshot. The only thing is we need to drag and drop the input file in the directory. The current project position directory. If we want to read and write in a specify location. Just adding the absolute path in the code. If we want to running the code, just click on the green arrow in the red box. It will automatically generate two output files for us, and you can see some action statements in the output.


CMPT365 Programming Assignment2 〉 src 〉 Main
Project
CMPT365 Programming Assignm
.idea
out
src
Dic_Element
Main
CMPT365 Programming Assignn
Q1.in
Q1.out
External Libraries
Scratches and Consoles
Run: Main
4: Run
TODO
6: Problems
Terminal
CMPT365 Programming Assignment 2
CMPT365 Programming Assignment 2 〉 src 〉 Main 〉 processing
Main.java
Matrix.java
Q2.in
Q2.out
Build

```
/Library/Java/JavaVirtualMachines

 Generating a output file...
 Reading input file successful !
 Encoding...
 Encoding successful !

Process finished with exit code
```

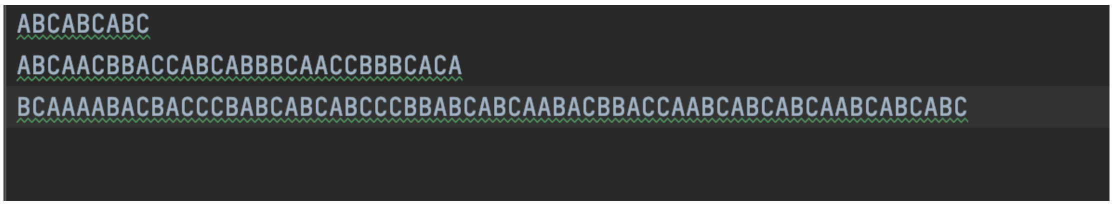
```
        Matrix temp_matrix = new M
        Matrix transpose_matrix =
        make_transpose(transform_m
        for (int i = 0; i < size;
            for (int j = 0; j < si

                double result = 0;
                for (int k = 0; k
                    result += tran
                }
                temp_matrix.values
            }
        }

        for (int i = 0; i < size;
            for (int j = 0; j < si

                double result = 0;
                for (int k = 0; k
                    result += temp
                }
                result_matrix.valu
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.2.jdk/Contents/Home/bin/ja
 Reading input file successful !
 Reading input file successful !
 Generating a output file...
 Generating a output file...
```

Resulting:

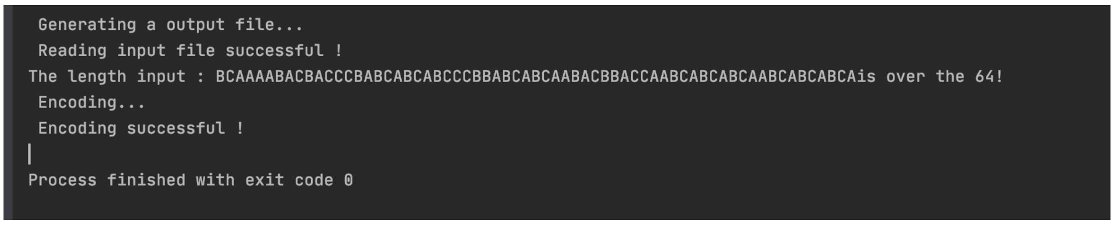
Q1:

In the first question there are two file, first file is for main function and another one is a class called Dic_element. I generate three different sequence which their length is in the different range in the Qi.in file.

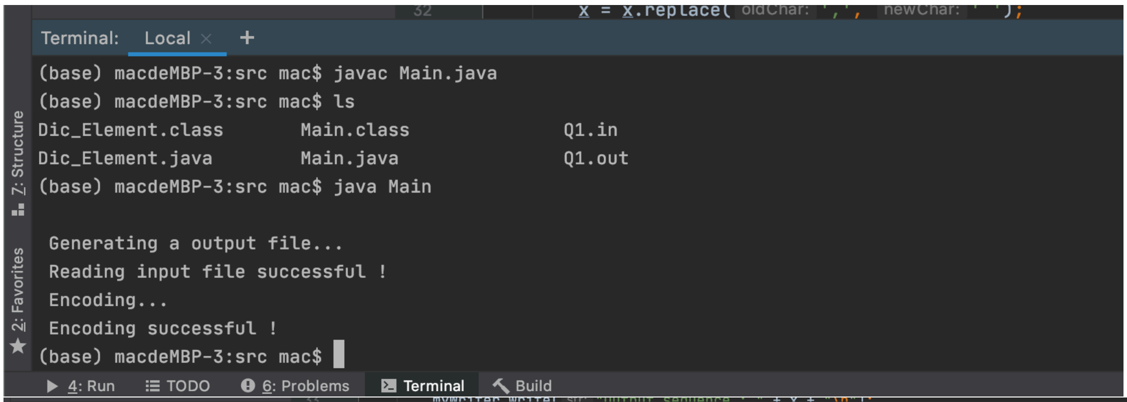


```
ABCABCABC
ABCAACBBACCABCABBBCAACCBBBCACA
BCAAAABACBACCCBABCABCABCCCBBABCABCAABACBBACCAABCABCABCAABCABCABC
```

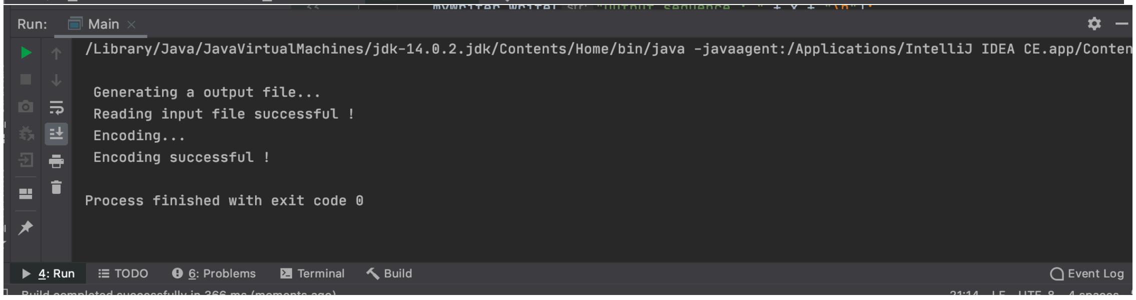
If the length of sequence is over the 64, then it will show the warning message and not count that sequence into the output file.



```
 Generating a output file...
 Reading input file successful !
The length input : BCAAAABACBACCCBABCABCABCCCBBABCABCAABACBBACCAABCABCABCAABCABCABCAis over the 64!
 Encoding...
 Encoding successful !

Process finished with exit code 0
```



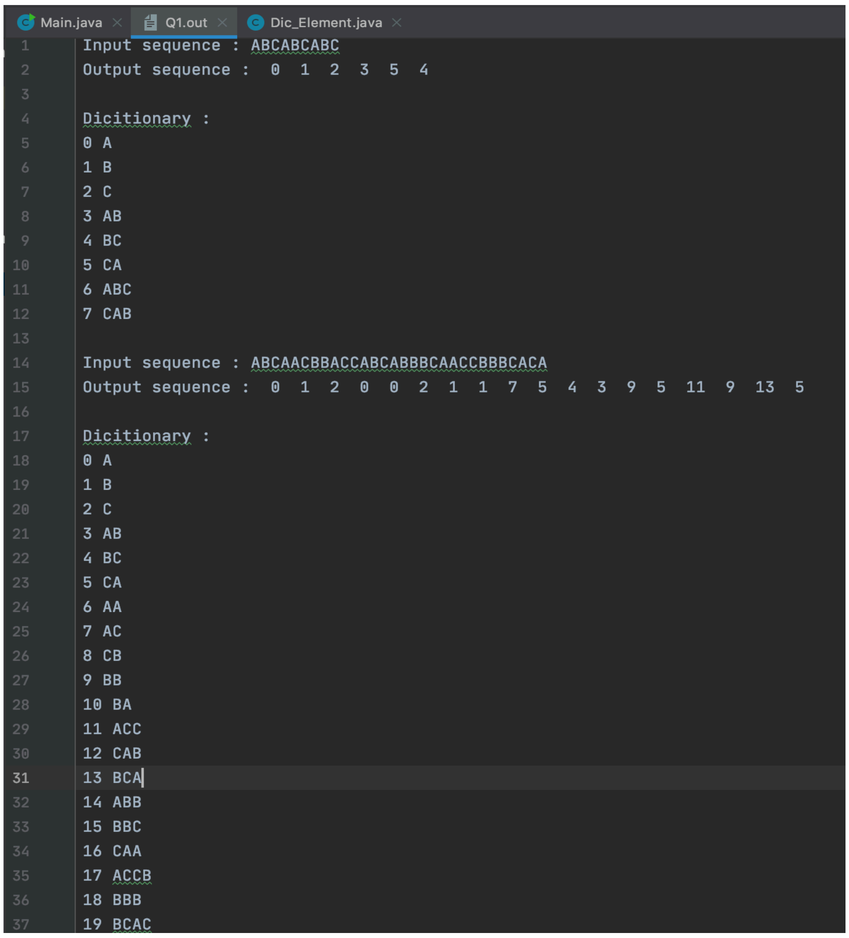
```
(base) macdeMBP-3:src mac$ javac Main.java
(base) macdeMBP-3:src mac$ ls
Dic_Element.class       Main.class              Q1.in
Dic_Element.java        Main.java               Q1.out
(base) macdeMBP-3:src mac$ java Main

 Generating a output file...
 Reading input file successful !
 Encoding...
 Encoding successful !
(base) macdeMBP-3:src mac$
```



```
/Library/Java/JavaVirtualMachines/jdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Conten

 Generating a output file...
 Reading input file successful !
 Encoding...
 Encoding successful !

Process finished with exit code 0
```

These two screenshots show the result of compiling and running execution in the terminal and running in the IDE.

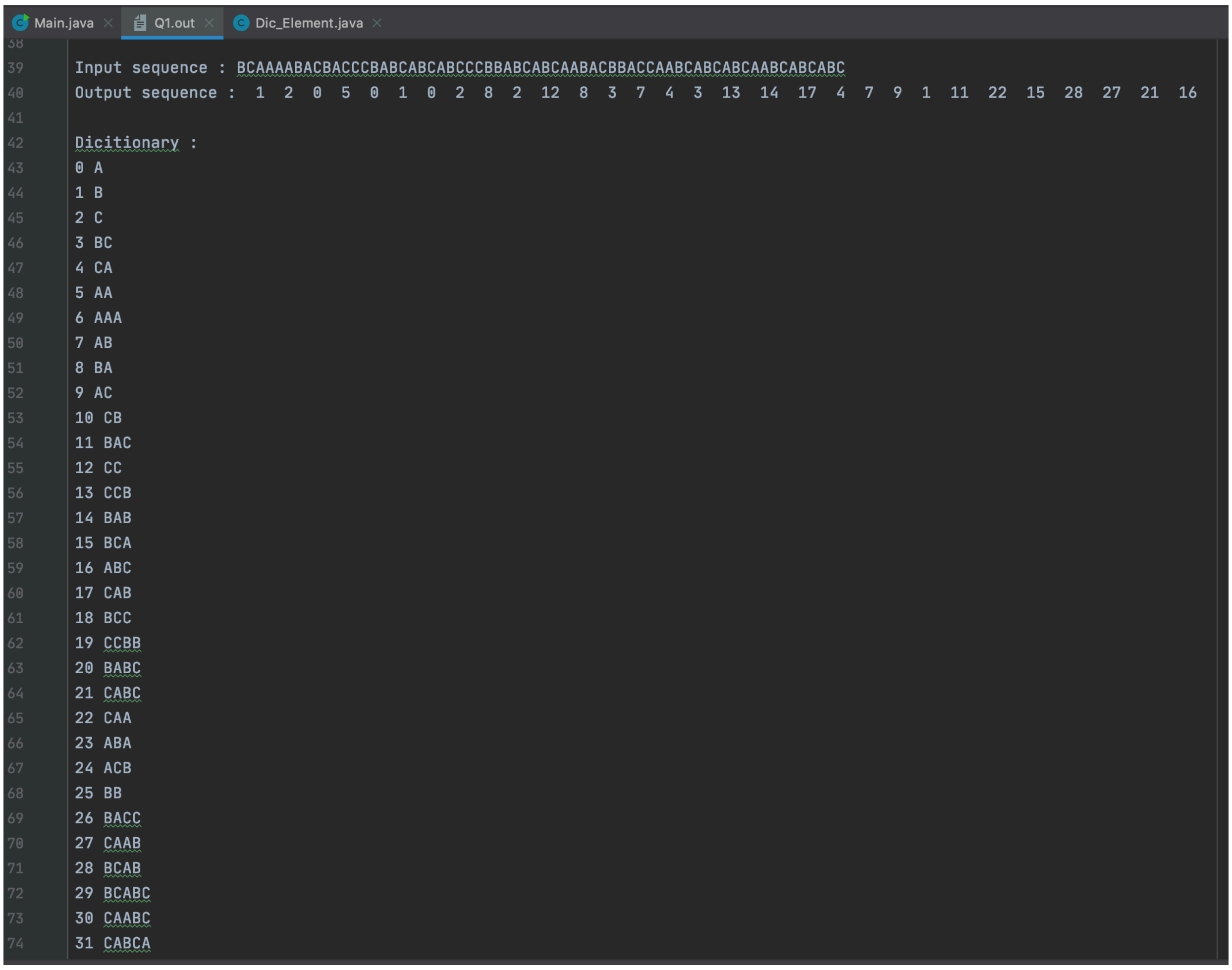
The resulting for the question one as follow:



```
Input sequence : ABCABCABC
Output sequence :  0  1  2  3  5  4

Dicitionary :
0 A
1 B
2 C
3 AB
4 BC
5 CA
6 ABC
7 CAB

Input sequence : ABCAACBBACCABCABBBCAACCBBBCACA
Output sequence :  0  1  2  0  0  2  1  1  7  5  4  3  9  5  11  9  13  5

Dicitionary :
0 A
1 B
2 C
3 AB
4 BC
5 CA
6 AA
7 AC
8 CB
9 BB
10 BA
11 ACC
12 CAB
13 BCA
14 ABB
15 BBC
16 CAA
17 ACCB
18 BBB
19 BCAC
```

This screenshot contains the result of first two sequence and the dictionary for each one.

This is the last example for the question one. (64 letters)

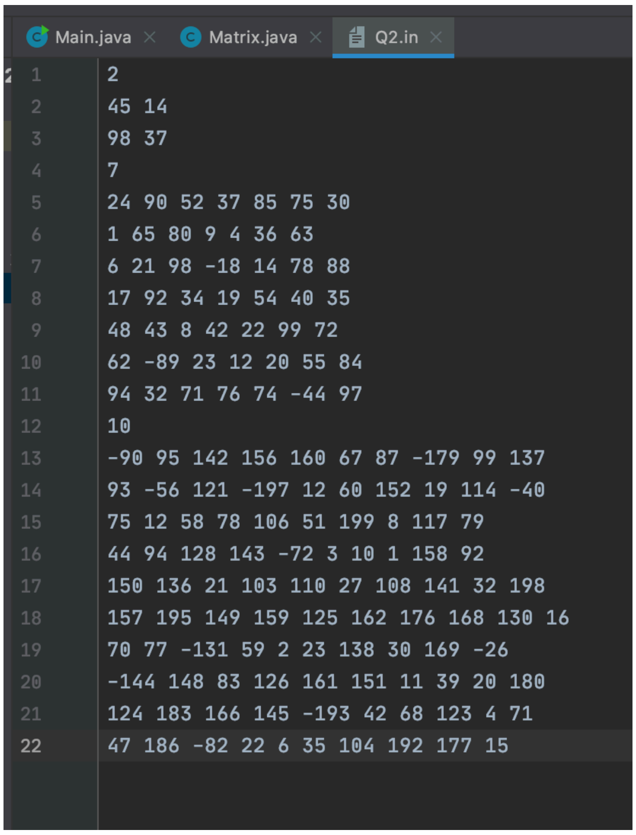


```
Input sequence : BCAAAABACBACCCBABCABCABCCCBBABCABCAABACBBACCAABCABCABCAABCABCABC
Output sequence :  1  2  0  5  0  1  0  2  8  2  12  8  3  7  4  3  13  14  17  4  7  9  1  11  22  15  28  27  21  16

Dicitionary :
0 A
1 B
2 C
3 BC
4 CA
5 AA
6 AAA
7 AB
8 BA
9 AC
10 CB
11 BAC
12 CC
13 CCB
14 BAB
15 BCA
16 ABC
17 CAB
18 BCC
19 CCBB
20 BABC
21 CABC
22 CAA
23 ABA
24 ACB
25 BB
26 BACC
27 CAAB
28 BCAB
29 BCABC
30 CAABC
31 CABCA
```

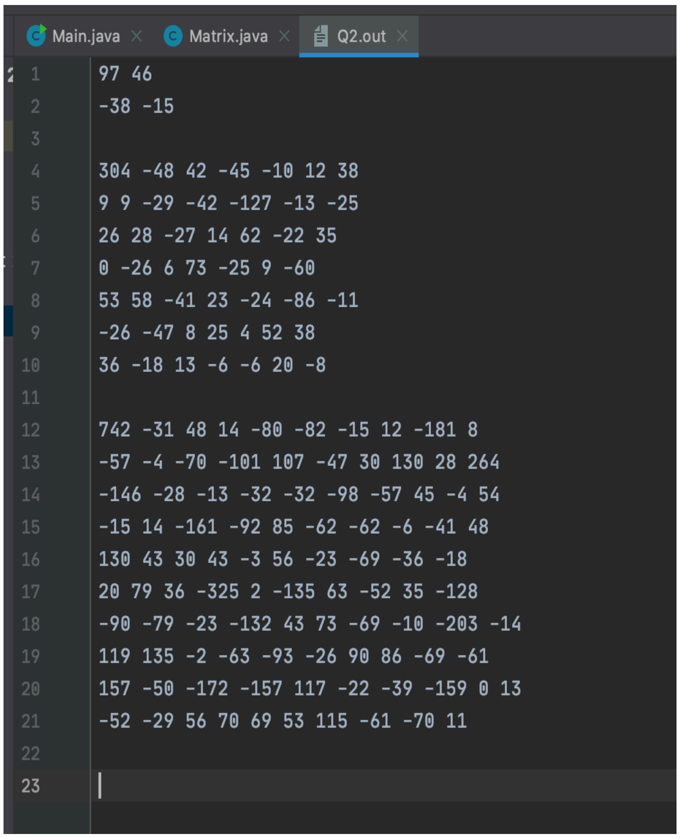
My code starts in the main function, opens the object file, reads each line and saves them in an array list which has string data type. Generates the initialized dictionary (containing the three known characters and the corresponding numbers). Adapt the pseudocode provided in class and generate the final answer. Since I chose to read all sequences at once, I used a nested loop to read each character. Before proceeding with each question, find the first character (S) and then read the next character (C) in sequence. If some combination of S plus C already exist in the dictionary, let S is equal to S plus C. If not in the dictionary, print the number corresponding to the S character. Add S+C to the dictionary and give it a corresponding number. And so on until you get to the end of the sequence. So now we have a compressed answer. Compression is done three times in the same way. Also I created a new class called DIC_ELEMENT to act as a dictionary.

Q2:

In the second question there are two files, one is for main function an another one is a class called Matrix. In the input file Q2.in, there are different three inputs with different size. Details are shown as follow:

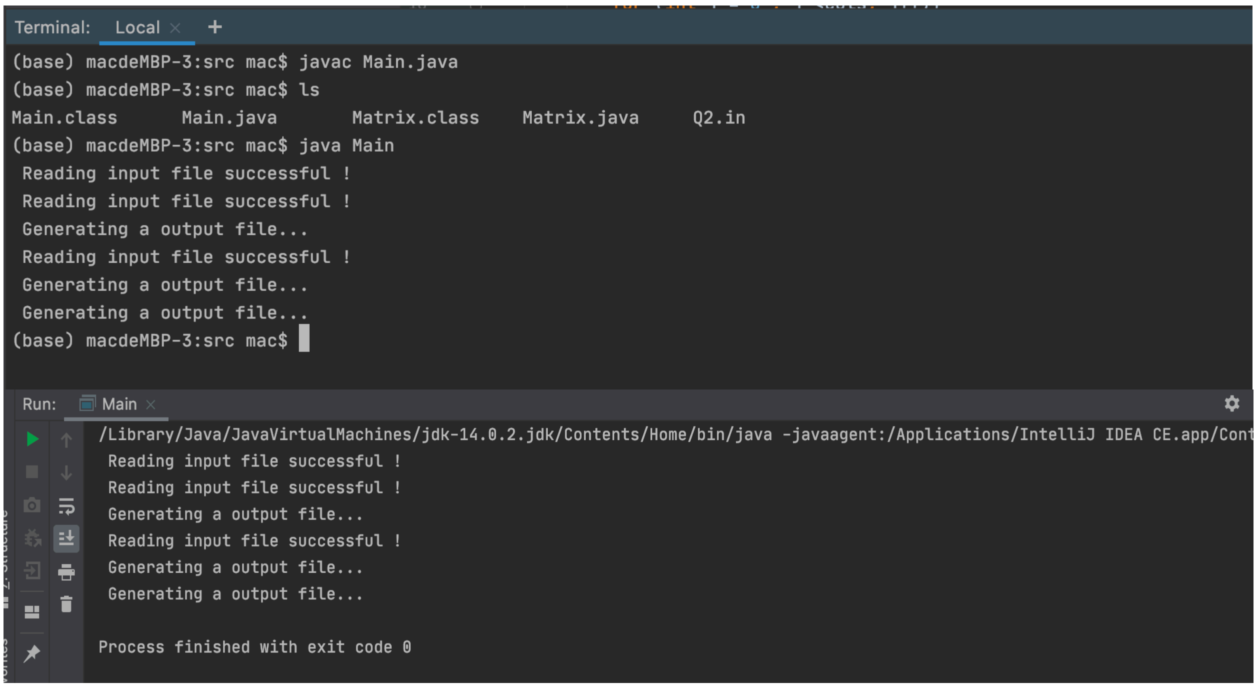


```
2
45 14
98 37
7
24 90 52 37 85 75 30
1 65 80 9 4 36 63
6 21 98 -18 14 78 88
17 92 34 19 54 40 35
48 43 8 42 22 99 72
62 -89 23 12 20 55 84
94 32 71 76 74 -44 97
10
-90 95 142 156 160 67 87 -179 99 137
93 -56 121 -197 12 60 152 19 114 -40
75 12 58 78 106 51 199 8 117 79
44 94 128 143 -72 3 10 1 158 92
150 136 21 103 110 27 108 141 32 198
157 195 149 159 125 162 176 168 130 16
70 77 -131 59 2 23 138 30 169 -26
-144 148 83 126 161 151 11 39 20 180
124 183 166 145 -193 42 68 123 4 71
47 186 -82 22 6 35 104 192 177 15
```



```
97 46
-38 -15

304 -48 42 -45 -10 12 38
9 9 -29 -42 -127 -13 -25
26 28 -27 14 62 -22 35
0 -26 6 73 -25 9 -60
53 58 -41 23 -24 -86 -11
-26 -47 8 25 4 52 38
36 -18 13 -6 -6 20 -8

742 -31 48 14 -80 -82 -15 12 -181 8
-57 -4 -70 -101 107 -47 30 130 28 264
-146 -28 -13 -32 -32 -98 -57 45 -4 54
-15 14 -161 -92 85 -62 -62 -6 -41 48
130 43 30 43 -3 56 -23 -69 -36 -18
20 79 36 -325 2 -135 63 -52 35 -128
-90 -79 -23 -132 43 73 -69 -10 -203 -14
119 135 -2 -63 -93 -26 90 86 -69 -61
157 -50 -172 -157 117 -22 -39 -159 0 13
-52 -29 56 70 69 53 115 -61 -70 11

```

The left one is an input file and right one is output file. They have size two, seven and ten respectively.



```
(base) macdeMBP-3:src mac$ javac Main.java
(base) macdeMBP-3:src mac$ ls
Main.class      Main.java       Matrix.class    Matrix.java     Q2.in
(base) macdeMBP-3:src mac$ java Main
 Reading input file successful !
 Reading input file successful !
 Generating a output file...
 Reading input file successful !
 Generating a output file...
 Generating a output file...
(base) macdeMBP-3:src mac$
```

```
/Library/Java/JavaVirtualMachines/jdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Cont
 Reading input file successful !
 Reading input file successful !
 Generating a output file...
 Reading input file successful !
 Generating a output file...
 Generating a output file...

Process finished with exit code 0
```

For this question, the way of running is as same as the previous one.
These two screenshots show the result of compiling and running execution in the terminal and running in the IDE.

Because the size of the file is different (the first line is usually the size of the matrix, or if there is only one element in a row, that thing is the size of the matrix), the rest of the input matrix is the element. I use two Boolean values to determine whether I need to continue reading or do something later. Once again, when we read a row with only one number, we should do the Discrete Cosine Transform for the previous reading matrix. When we read an input matrix, we need to generate a transformation matrix based on its size. Usually the first row of the transformation matrix has only $\sqrt{\frac{1}{N}}$. N is standard for the size of matrix. Elements after the first row follow the formula $C_{i,j} = a * \cos(\frac{(2j+1)*i*\pi}{2N})$. Where i is the row and j is the column, and a = $\sqrt{\frac{2}{N}}$. When we have this matrix. According to formula Y = TXT^T, we also need to find the transpose matrix for the transformation matrix. By definition, rows and columns to be swapped out, we can get marked as T^T. Once again, according to the formula, we're going to multiply three matrices in the order of transformation matrix, input matrix, and transpose matrix together. Then we doing the Discrete Cosine Transform successfully.