

# GUI Overview

The screenshot displays the Playfair Cipher application window. The title bar reads "Playfair Cipher" and includes standard window controls. The menu bar contains "Customize" and "Help". The main interface is divided into two modes: "File Mode" (selected) and "Quick Mode".

**File Mode Section:**

- Input File:** A text field with a "Browse" button next to it.
- Output File Name:** A text field.

**Quick Mode Section:**

- Enter Keyword:** A text field with a note "Must be between 5 - 255" and a "show password" checkbox.
- Input Message:** A text area with a "Clear" button and a placeholder text "Insert a short message to encrypt or decrypt".
- Output Message:** A text area with a placeholder text "Output message will be displayed here."

**Action Buttons:**

- Encode:** A large button at the bottom left.
- Decode:** A large button at the bottom right.

# Two encoding modes: File/Quick

The screenshot shows the Playfair Cipher application window. The title bar reads "Playfair Cipher" with standard window controls. Below the title bar are "Customize" and "Help" menu items. The main interface is divided into two modes: "File Mode" (selected) and "Quick Mode".

**File Mode:**

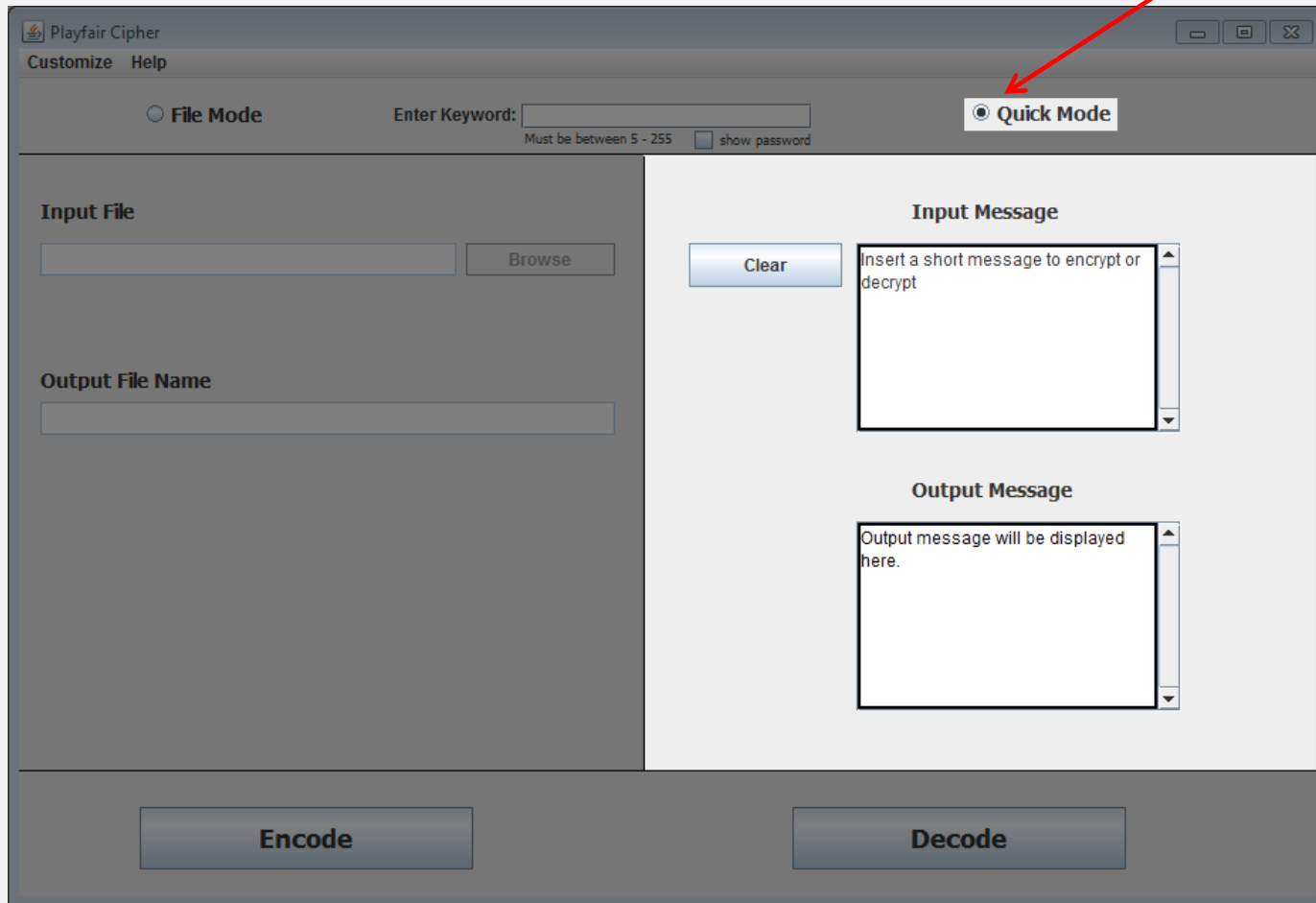
- Input File:** A text field with a "Browse" button next to it.
- Output File Name:** A text field.

**Quick Mode:**

- Enter Keyword:** A text field with a note "Must be between 5 - 255" and a "show password" checkbox.
- Input Message:** A text area with a "Clear" button and a placeholder "Insert a short message to encrypt or decrypt".
- Output Message:** A text area with a placeholder "Output message will be displayed here."

At the bottom of the window are two large buttons: "Encode" and "Decode".

# Quick mode:



The screenshot shows the Playfair Cipher application window. The title bar reads "Playfair Cipher". Below the title bar are two menu items: "Customize" and "Help". The main interface has a dark grey background. At the top, there are two radio buttons: "File Mode" (unselected) and "Quick Mode" (selected). A red arrow points to the "Quick Mode" radio button. To the right of the radio buttons is a text field labeled "Enter Keyword:" with a placeholder text "Must be between 5 - 255" and a checkbox labeled "show password". Below the radio buttons, the interface is split into two main sections. The left section is titled "Input File" and contains a text field and a "Browse" button. Below this is a section titled "Output File Name" with a text field. The right section is titled "Input Message" and contains a "Clear" button and a text area with the placeholder text "Insert a short message to encrypt or decrypt". Below the "Input Message" section is a section titled "Output Message" with a text area containing the placeholder text "Output message will be displayed here.". At the bottom of the window, there are two large buttons: "Encode" and "Decode".

Playfair Cipher

Customize Help

☐ File Mode ☒ Quick Mode

Enter Keyword:  Must be between 5 - 255 ☐ show password

**Input File**

Browse

**Output File Name**

**Input Message**

Clear

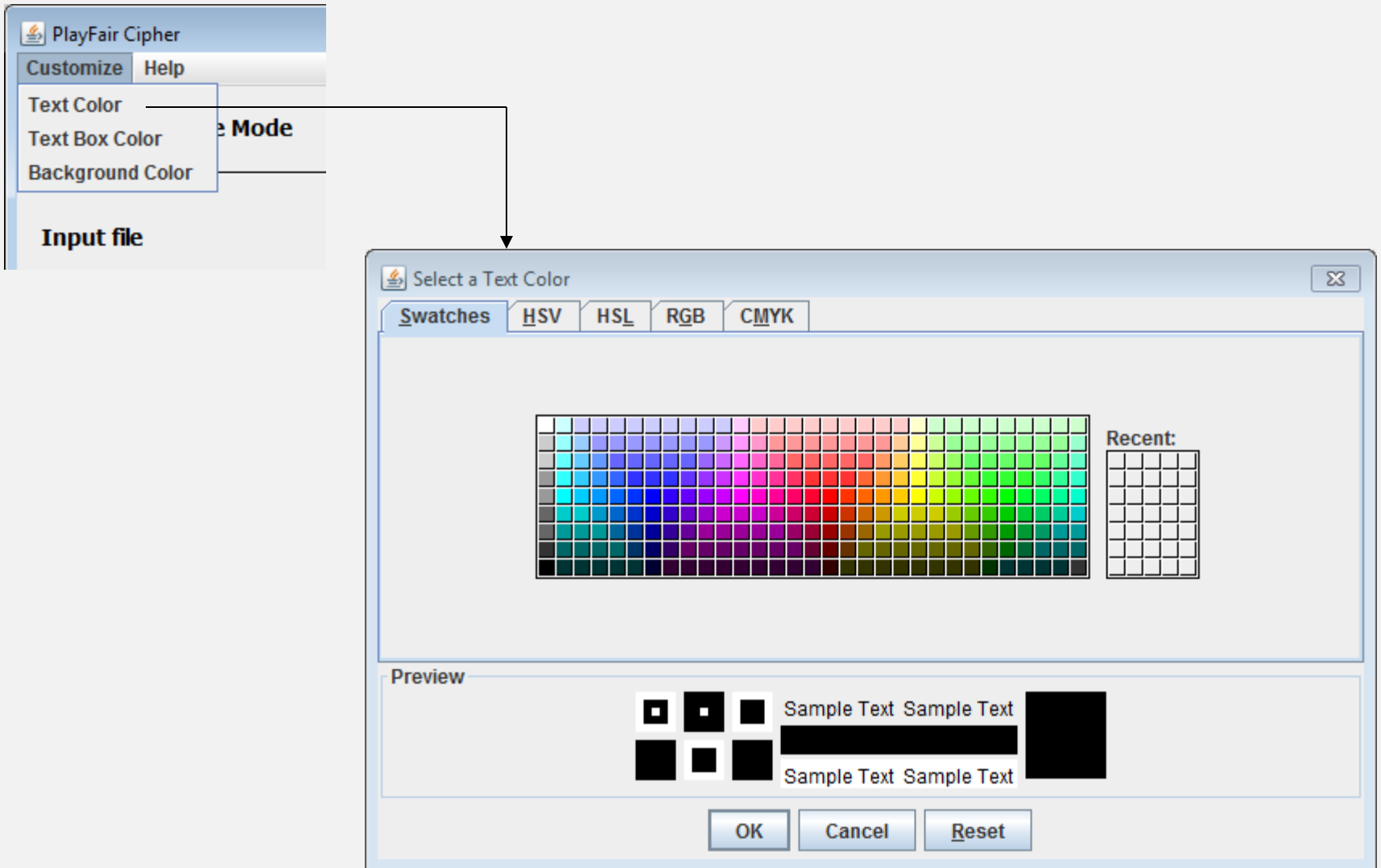
Insert a short message to encrypt or decrypt

**Output Message**

Output message will be displayed here.

Encode Decode

# GUI Customization



# GUI Customization

The image shows a software window titled "Playfair Cipher" with a menu bar containing "Customize" and "Help". The window is divided into two main sections: "File Mode" (selected) and "Quick Mode".

**File Mode:**

- Input File:** A text field with a "Browse" button next to it.
- Output File Name:** A text field below the input file field.

**Quick Mode:**

- Enter Keyword:** A text field with a "show password" checkbox and a note "Must be between 5 - 255".
- Input Message:** A text area with a "Clear" button and a placeholder text "Insert a short message to encrypt or decrypt".
- Output Message:** A text area with a placeholder text "Output message will be displayed here."

**Buttons:**

- Encode:** A button at the bottom left.
- Decode:** A button at the bottom right.

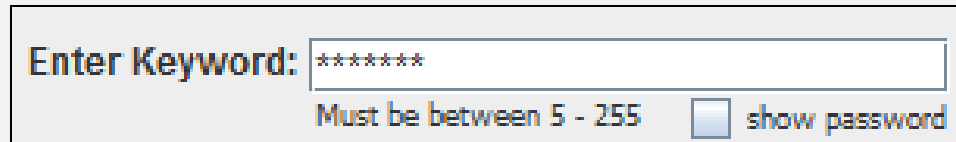
# Preprocessing

## Quick Mode

---

For the encoding of brief segments  
of text within the GUI.

keyword ← txtKeyword.getText();



Enter Keyword:

Must be between 5 - 255 ☐ show password

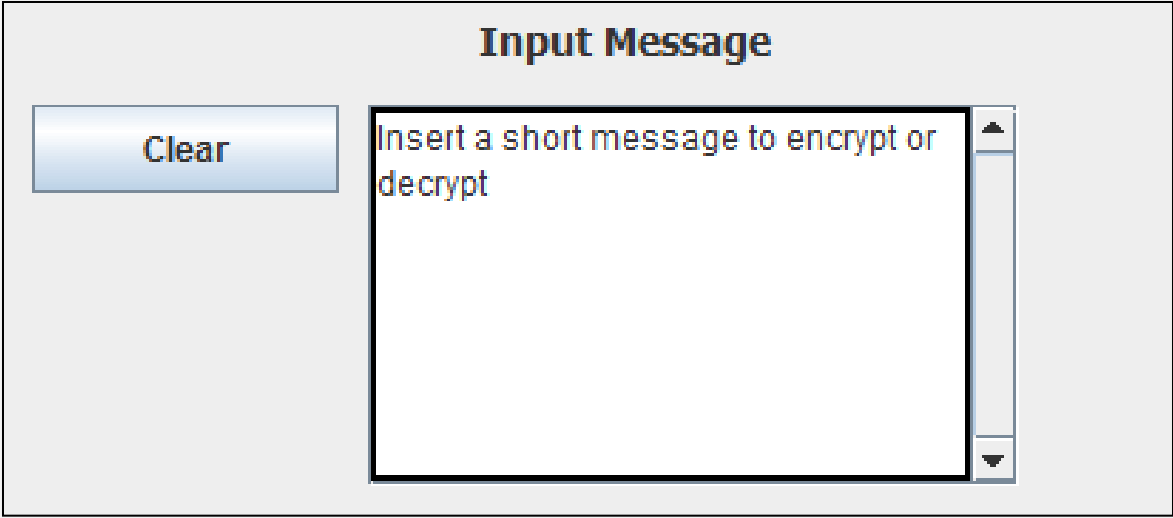
Validate:

$5 \leq \text{Length} < 255$  characters

# Get message

---

Message ← txtAreaInput.getText();



The image shows a Java Swing window titled "Input Message". Inside the window, there is a text area on the right with the placeholder text "Insert a short message to encrypt or decrypt". To the left of the text area is a button labeled "Clear". The text area has a vertical scrollbar on its right side.



# GUI Encoding Interface

---

## Quick Mode

`message.quickmessage(Message);`



`messageAsIntArrayList`

"Hello"



H	e	l	l	o
72	101	108	108	111

# GUI Encoding Interface

---

## Quick Mode

```
ciphergrid.encode(keyword, messageAsIntArr);
```



ENCODING PROCESS



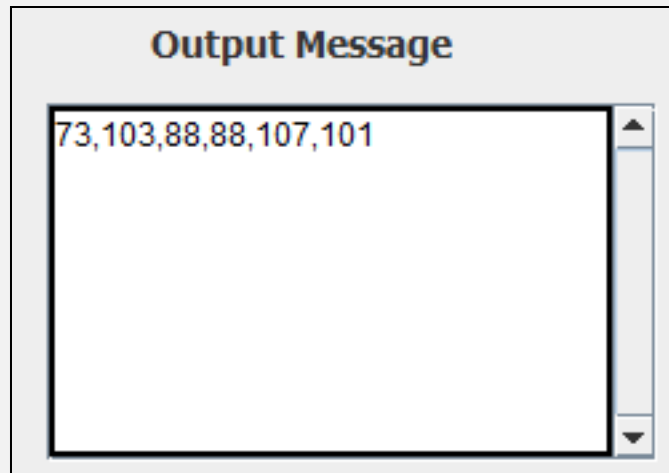
```
encodedMessageAsIntArrayList
```

# Process encoded message for display

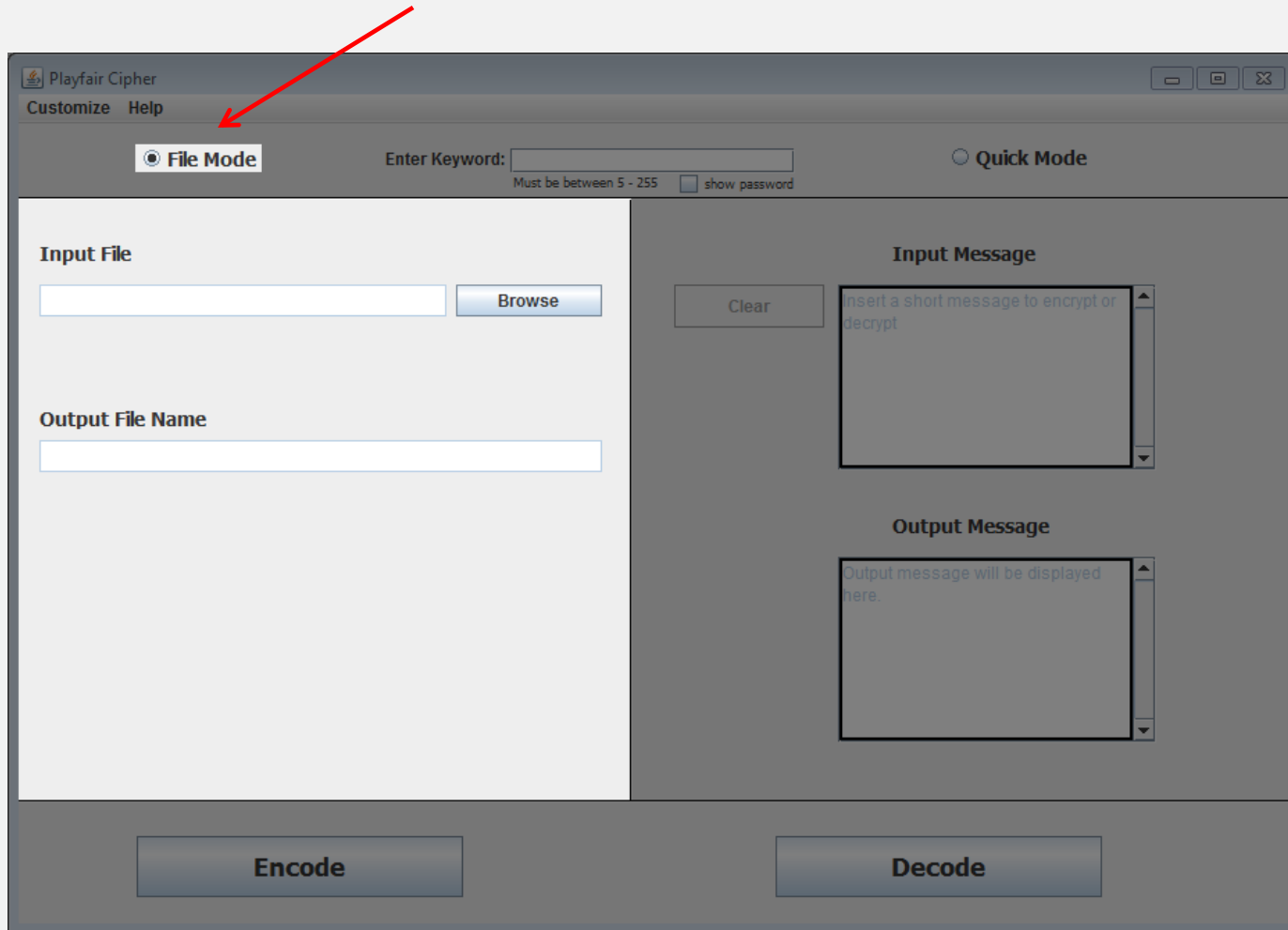
---

Format string  $\longrightarrow$  encodedMessageAsString

Display: `txtAreaOutput.setText(encodedMessageAsString);`



# File mode:



# Preprocessing

## File Mode

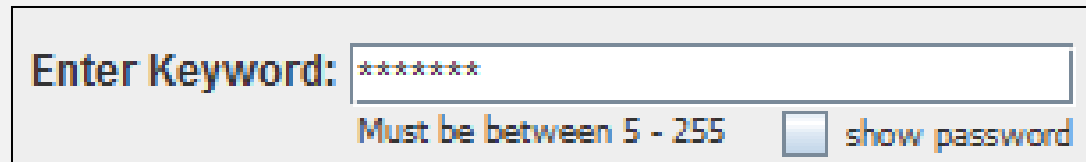
---

For the encoding of existing external text files to be saved in a new encoded file.

# File Mode

---

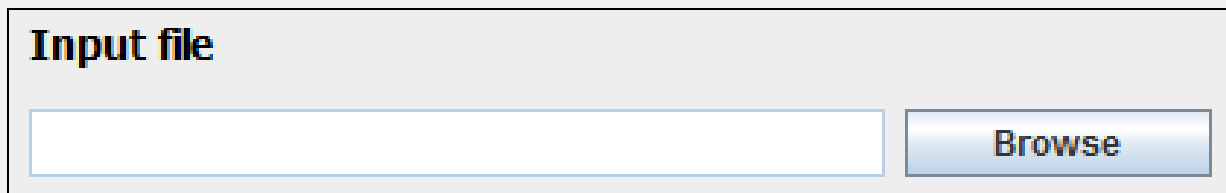
keyword ← JTextField.getText();



Enter Keyword: \*\*\*\*\*

Must be between 5 - 255 ☐ show password

filePath ← JFileChooser



Input file

# GUI Encoding Interface

---

## File Mode

```
message.readMessageFile(filePath);
```



```
messageAsIntArrayList
```

---



File.txt

1: Alpha

2: Beta

3: Gamma

65, 108, 112, 104, 97

66, 101, 116, 97

71, 97, 109, 109, 97

# GUI Encoding Interface

---

## File Mode

```
ciphergrid.encode(keywordAsString, messageAsInts);
```



ENCODING PROCESS



```
encodedMessageAsIntArrayList
```



# GUI Encoding Interface

---

## File Mode

```
ciyWriter.writeEncodedMsgToFile(encodedMessage, filePath);
```

65, 108, 112, 104, 97

66, 101, 116, 97

71, 97, 109, 109, 97



Encoded  
File.txt

# Encoding Process

Under the hood

# Create a table

---

## Google Guava libraries



```
import com.google.common.collect.HashBasedTable;
```

```
private Table<Integer,Integer,Integer> table = HashBasedTable.create(1, 1);
```



[illegible]

[illegible]

# Fill table with ASCII integers 0-255 (skip any in keyword)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	107	101	121	119	111	114	100	0	1	2	3	4	5	6	7	8
1	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
3	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
4	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
5	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
6	89	90	91	92	93	94	95	96	97	98	99	102	103	104	105	106
7	108	109	110	112	113	115	116	117	118	120	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
10	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
11	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
12	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
13	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
14	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
15	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

# Encoding Process

---

Get a pair of letters from the message

(message is padded with special character to make an even number of pairs)

**F**irst letter

**S**econd letter



# Get First letter in the pair of letters

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	107	101	121	119	111	114	100	0	1	2	3	4	5	6	7	8
1	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
3	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
4	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
5	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
6	89	90	91	92	93	94	95	96	97	98	99	102	103	104	105	106
7	108	109	110	112	113	115	116	117	118	120	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
10	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
11	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
12	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
13	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
14	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
15	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

# Get **S**econd letter in the pair of letters

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	107	101	121	119	111	114	100	0	1	2	3	4	5	6	7	8
1	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
3	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
4	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
5	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
6	89	90	91	92	93	94	95	96	97	98	99	102	103	104	105	106
7	108	109	110	112	113	115	116	117	118	120	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
10	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
11	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
12	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
13	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
14	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
15	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

(not last column)

[illegible]

[illegible]

# Encoding Notation

---

$F_r$  = First letters row number.

$F_c$  = First letters column number.

$S_r$  = Second letters row number.

$S_c$  = Second letters column number.

	0	1	2	3	4	...	12	13	14	15
0										
1			F → F	S → S						
2										
3										
4										
...										
12										
13										
14										
15										

Encode

Pseudo-code

$F_c ++;$

$S_c ++;$

# Case: same row (last column)

[illegible]

(last column)

[illegible]



	0	1	2	3	4	...	12	13	14	15
0										
1	S						F → F			S
2										
3										
4										
...										
12										
13										
14										
15										

Encode

Pseudo-code

$F_c ++;$   
 $S_c == 0;$



# Case: same column

(not last row)

[illegible]

	0	1	2	3	4	...	12	13	14	15
0										
1			F							
2			F							
3			S							
4			S							
...										
12										
13										
14										
15										

Encode

## Pseudo-code

$F_r ++;$

$S_r ++;$

# Case: same column

---

(last row)

	0	1	2	3	4	...	12	13	14	15
0			S							
1										
2										
3										
4										
...										
12										
13			F							
14			F							
15			S							

	0	1	2	3	4	...	12	13	14	15
0			S							
1										
2										
3										
4										
...										
12										
13			F							
14			F							
15			S							

Encode

Pseudo-code

$F_r ++;$

$S_r == 0;$

(not same row or column)

[illegible]

(not same row or column)

[illegible]



	0	1	2	3	4	...	12	13	14	15
0										
1			F	→	F					
2										
3			S	←	S					
4										
...										
12										
13										
14										
15										

Encode

Pseudo-code

$F_c == S_c;$

$S_c == F_c;$