



# VIT<sup>®</sup>

## Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

**B.Tech. Winter Semester 2024-25**  
**School Of Computer Science and Engineering**  
**(SCOPE)**

# Digital Assignment - I

## Cryptography and Network Security Lab

Apurva Mishra: 22BCE2791

Date: 9 Feb, 2025

### Contents

<b>1</b>	<b>Ceaser Cipher</b>	<b>2</b>
1.1	Code	2
1.2	Output	3
<b>2</b>	<b>Playfair Cipher</b>	<b>3</b>
2.1	Code	3
2.2	Output	7
<b>3</b>	<b>Rail Fence Cipher</b>	<b>7</b>
3.1	Code	7
3.2	Output	10
<b>4</b>	<b>Vigenere Cipher</b>	<b>10</b>
4.1	Code	10
4.2	Output	12

# 1 Ceaser Cipher

## 1.1 Code

### Code 0: main.c

```
1  #include <ctype.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  #define MAX_LEN 1000
6
7  void encrypt(char *text, int key) {
8      for (int i = 0; text[i] != '\0'; i++) {
9          if (isalpha(text[i])) {
10             char base = isupper(text[i]) ? 'A' : 'a';
11             text[i] = (text[i] - base + key) % 26 + base;
12         }
13     }
14 }
15
16 void decrypt(char *text, int key) {
17     for (int i = 0; text[i] != '\0'; i++) {
18         if (isalpha(text[i])) {
19             char base = isupper(text[i]) ? 'A' : 'a';
20             text[i] = (text[i] - base - key + 26) % 26 + base;
21         }
22     }
23 }
24
25 int main() {
26     char text[MAX_LEN];
27     int key;
28     int op;
29
30     printf("Enter text: ");
31     fgets(text, MAX_LEN, stdin);
32     text[strcspn(text, "\n")] = '\0';
33
34     printf("Enter key value: ");
35     scanf("%d", &key);
36
37     printf("Choose operation: 1 for Encryption, 2 for Decryption: ");
38     scanf("%d", &op);
39
40     switch (op) {
41     case 1:
42         encrypt(text, key);
43         printf("Encrypted text: %s\n", text);
44         break;
45     case 2:
46         decrypt(text, key);
47         printf("Decrypted text: %s\n", text);
```

```

48     break;
49     default:
50         printf("Invalid choice!\n");
51         break;
52     }
53
54     return 0;
55 }

```

## 1.2 Output

```

da/ass1/q1 via C v16.0.0-clang
> just run
zig cc main.c -o main
./main
Enter text: there
Enter key value: 3
Choose operation: 1 for Encryption, 2 for Decryption: 1
Encrypted text: wkhuh

da/ass1/q1 via C v16.0.0-clang took 10s
> just run
zig cc main.c -o main
./main
Enter text: wkhuh
Enter key value: 3
Choose operation: 1 for Encryption, 2 for Decryption: 2
Decrypted text: there

da/ass1/q1 via C v16.0.0-clang took 7s
> 

```

## 2 Playfair Cipher

### 2.1 Code

Code 0: main.l

```

1  #include <ctype.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  #define SIZE 5
6  #define MAX_TEXT 100
7

```

```

8 char keySquare[SIZE][SIZE];
9
10 void generateKeySquare(const char *key) {
11     int map[26] = {0};
12     int x = 0, y = 0;
13     char processedKey[26] = "";
14     int index = 0;
15
16     for (int i = 0; key[i] != '\0'; i++) {
17         char ch = toupper(key[i]);
18         if (ch == 'J')
19             ch = 'I';
20         if (!map[ch - 'A'] && isalpha(ch)) {
21             map[ch - 'A'] = 1;
22             processedKey[index++] = ch;
23         }
24     }
25
26     for (char ch = 'A'; ch <= 'Z'; ch++) {
27         if (ch == 'J')
28             continue;
29         if (!map[ch - 'A']) {
30             processedKey[index++] = ch;
31         }
32     }
33
34     index = 0;
35     for (int i = 0; i < SIZE; i++) {
36         for (int j = 0; j < SIZE; j++) {
37             keySquare[i][j] = processedKey[index++];
38         }
39     }
40 }
41
42 void findPosition(char ch, int *row, int *col) {
43     if (ch == 'J')
44         ch = 'I';
45     for (int i = 0; i < SIZE; i++) {
46         for (int j = 0; j < SIZE; j++) {
47             if (keySquare[i][j] == ch) {
48                 *row = i;
49                 *col = j;
50                 return;
51             }
52         }
53     }
54 }
55
56 void prepareText(char *text) {
57     int len = strlen(text);
58     for (int i = 0; i < len; i++) {
59         text[i] = toupper(text[i]);
60         if (text[i] == 'J')

```

```

61     text[i] = 'I';
62 }
63
64 char newText[MAX_TEXT];
65 int newIndex = 0;
66
67 for (int i = 0; i < len; i++) {
68     if (!isalpha(text[i]))
69         continue;
70     newText[newIndex++] = text[i];
71     if (i + 1 < len && text[i] == text[i + 1]) {
72         newText[newIndex++] = 'X';
73     }
74 }
75
76 if (newIndex % 2 != 0) {
77     newText[newIndex++] = 'X';
78 }
79 newText[newIndex] = '\0';
80 strcpy(text, newText);
81 }
82
83 void playfairCipher(char *text, int encrypt) {
84     for (int i = 0; i < strlen(text); i += 2) {
85         int r1, c1, r2, c2;
86         findPosition(text[i], &r1, &c1);
87         findPosition(text[i + 1], &r2, &c2);
88
89         if (r1 == r2) {
90             text[i] = keySquare[r1][(c1 + encrypt + SIZE) % SIZE];
91             text[i + 1] = keySquare[r2][(c2 + encrypt + SIZE) % SIZE];
92         } else if (c1 == c2) {
93             text[i] = keySquare[(r1 + encrypt + SIZE) % SIZE][c1];
94             text[i + 1] = keySquare[(r2 + encrypt + SIZE) % SIZE][c2];
95         } else {
96             text[i] = keySquare[r1][c2];
97             text[i + 1] = keySquare[r2][c1];
98         }
99
100     text[i] = tolower(text[i]);
101     text[i + 1] = tolower(text[i + 1]);
102 }
103 }
104
105 int main() {
106     char key[MAX_TEXT], text[MAX_TEXT];
107     int op;
108
109     printf("Enter key: ");
110     fgets(key, MAX_TEXT, stdin);
111     key[strcspn(key, "\n")] = '\0';
112
113     generateKeySquare(key);

```

```

114
115     printf("Enter text: ");
116     fgets(text, MAX_TEXT, stdin);
117     text[strcspn(text, "\n")] = '\0';
118
119     prepareText(text);
120
121     printf("Choose operation (1 for Encryption, 2 for Decryption): ");
122     scanf("%d", &op);
123
124     switch (op) {
125     case 1:
126         playfairCipher(text, 1);
127         printf("Encrypted text: %s\n", text);
128         break;
129     case 2:
130         playfairCipher(text, -1);
131         printf("Decrypted text: %s\n", text);
132         break;
133     default:
134         printf("Invalid choice!\n");
135         break;
136     }
137
138     return 0;
139 }

```

## 2.2 Output

```
da/ass1/q2 via C v16.0.0-clang
> just run
zig cc main.c -o main
./main
Enter key: batter
Enter text: heater
Choose operation (1 for Encryption, 2 for Decryption): 1
Encrypted text: grterb

da/ass1/q2 via C v16.0.0-clang took 10s
> just run
zig cc main.c -o main
./main
Enter key: batter
Enter text: grterb
Choose operation (1 for Encryption, 2 for Decryption): 2
Decrypted text: heater

da/ass1/q2 via C v16.0.0-clang took 7s
> |
```

## 3 Rail Fence Cipher

### 3.1 Code

Code 0: main.l

```
1  #include <ctype.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  #define MAX_LEN 1000
6
7  void encrypt(char *text, int key) {
8      int len = strlen(text);
9      char rail[key][len];
10
11     for (int i = 0; i < key; i++) {
12         for (int j = 0; j < len; j++) {
13             rail[i][j] = '\n';
14         }
15     }
16
17     int row = 0, dir_down = 0;
```

```

18     for (int i = 0; i < len; i++) {
19         if (row == 0 || row == key - 1) {
20             dir_down = !dir_down;
21         }
22         rail[row][i] = text[i];
23         row += (dir_down) ? 1 : -1;
24     }
25
26     printf("Encrypted Text: ");
27     for (int i = 0; i < key; i++) {
28         for (int j = 0; j < len; j++) {
29             if (rail[i][j] != '\n') {
30                 printf("%c", rail[i][j]);
31             }
32         }
33     }
34     printf("\n");
35 }
36
37 void decrypt(char *cipher, int key) {
38     int len = strlen(cipher);
39     char rail[key][len];
40
41     for (int i = 0; i < key; i++) {
42         for (int j = 0; j < len; j++) {
43             rail[i][j] = '\n';
44         }
45     }
46
47     int row = 0;
48     int dir_down = 0;
49     for (int i = 0; i < len; i++) {
50         if (row == 0 || row == key - 1) {
51             dir_down = !dir_down;
52         }
53         rail[row][i] = '*';
54         row += (dir_down) ? 1 : -1;
55     }
56
57     int index = 0;
58     for (int i = 0; i < key; i++) {
59         for (int j = 0; j < len; j++) {
60             if (rail[i][j] == '*' && index < len) {
61                 rail[i][j] = cipher[index++];
62             }
63         }
64     }
65
66     row = 0, dir_down = 0;
67     printf("Decrypted Text: ");
68     for (int i = 0; i < len; i++) {
69         if (row == 0 || row == key - 1) {
70             dir_down = !dir_down;

```



```

71     }
72     printf("%c", rail[row][i]);
73     row += (dir_down) ? 1 : -1;
74 }
75 printf("\n");
76 }
77
78 int main() {
79     char text[MAX_LEN];
80     int key, choice;
81
82     printf("Enter text: ");
83     fgets(text, MAX_LEN, stdin);
84     text[strcspn(text, "\n")] = '\0';
85
86     printf("Enter key (number of rails): ");
87     scanf("%d", &key);
88
89     printf("Choose operation: 1 for Encryption, 2 for Decryption: ");
90     scanf("%d", &choice);
91     getchar();
92
93     if (choice == 1) {
94         encrypt(text, key);
95     } else if (choice == 2) {
96         decrypt(text, key);
97     } else {
98         printf("Invalid choice!\n");
99     }
100
101     return 0;
102 }

```

## 3.2 Output

```
da/ass1/q3 via C v16.0.0-clang
> just run
zig cc main.c -o main
./main
Enter text: scatter
Enter key (number of rails): 3
Choose operation: 1 for Encryption, 2 for Decryption: 1
Encrypted Text: stctear

da/ass1/q3 via C v16.0.0-clang took 8s
> just run
zig cc main.c -o main
./main
Enter text: stctear
Enter key (number of rails): 3
Choose operation: 1 for Encryption, 2 for Decryption: 2
Decrypted Text: scatter

da/ass1/q3 via C v16.0.0-clang took 16s
> |
```

## 4 Vigenere Cipher

### 4.1 Code

Code 0: main.l

```
1  #include <ctype.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  #define MAX_LEN 1000
6
7  void encrypt(char *text, char *key) {
8      int textLen = strlen(text);
9      int keyLen = strlen(key);
10     char encryptedText[MAX_LEN];
11
12     for (int i = 0, j = 0; i < textLen; i++) {
13         if (isalpha(text[i])) {
14             char base = isupper(text[i]) ? 'A' : 'a';
15             char keyBase = isupper(key[j % keyLen]) ? 'A' : 'a';
16             encryptedText[i] =
17                 (text[i] - base + (key[j % keyLen] - keyBase)) % 26 + base;
```

```

18     j++;
19 } else {
20     encryptedText[i] = text[i];
21 }
22 }
23 encryptedText[textLen] = '\0';
24 printf("Encrypted Text: %s\n", encryptedText);
25 }
26
27 void decrypt(char *text, char *key) {
28     int textLen = strlen(text);
29     int keyLen = strlen(key);
30     char decryptedText[MAX_LEN];
31
32     for (int i = 0, j = 0; i < textLen; i++) {
33         if (isalpha(text[i])) {
34             char base = isupper(text[i]) ? 'A' : 'a';
35             char keyBase = isupper(key[j % keyLen]) ? 'A' : 'a';
36             decryptedText[i] =
37                 (text[i] - base - (key[j % keyLen] - keyBase) + 26) % 26 + base;
38             j++;
39         } else {
40             decryptedText[i] = text[i];
41         }
42     }
43     decryptedText[textLen] = '\0';
44     printf("Decrypted Text: %s\n", decryptedText);
45 }
46
47 int main() {
48     char text[MAX_LEN], key[MAX_LEN];
49     int choice;
50
51     printf("Enter text: ");
52     fgets(text, MAX_LEN, stdin);
53     text[strcspn(text, "\n")] = '\0';
54
55     printf("Enter key: ");
56     fgets(key, MAX_LEN, stdin);
57     key[strcspn(key, "\n")] = '\0';
58
59     printf("Choose operation: 1 for Encryption, 2 for Decryption: ");
60     scanf("%d", &choice);
61     getchar(); // Consume newline
62
63     if (choice == 1) {
64         encrypt(text, key);
65     } else if (choice == 2) {
66         decrypt(text, key);
67     } else {
68         printf("Invalid choice!\n");
69     }
70 }

```

```
71     return 0;
72 }
```

## 4.2 Output

```
da/ass1/q4 via C v16.0.0-clang took 6s
> just run
zig cc main.c -o main
./main
Enter text: there
Enter key: air
Choose operation: 1 for Encryption, 2 for Decryption: 1
Encrypted Text: tpvrm

da/ass1/q4 via C v16.0.0-clang took 8s
> just run
zig cc main.c -o main
./main
Enter text: tpvrm
Enter key: air
Choose operation: 1 for Encryption, 2 for Decryption: 2
Decrypted Text: there

da/ass1/q4 via C v16.0.0-clang took 7s
> |
```