

## Practical 2\Practical.py

```
1 #####
2                                     #Practical 2A
3 #####
4 def encrypt(inputText: str, key: int) → str:
5     matrix: list[list[chr]] = [list(" ") * len(inputText) for _ in range(key)]
6     i: int = 0
7     flag: bool = True
8     for j, ch in enumerate(inputText):
9         matrix[i][j] = ch
10        if flag:
11            i += 1
12        else:
13            i -= 1
14        if i == key - 1 or i == 0:
15            flag = not (flag)
16
17    for row in matrix:
18        print(*row)
19
20    result: str = ""
21    for row in matrix:
22        result += "".join(row).replace(" ", "").upper()
23
24    return result
25
26
27 def decrypt(inputText: str, key: int) → str:
28     matrix: list[list[chr]] = [list(" ") * len(inputText) for _ in range(key)]
29     result: list[str] = [" " for _ in range(len(inputText))]
30
31     ch: int = 0
32     for row in range(key):
33         i: int = 0
34         flag: bool = True
35         for x in range(len(inputText)):
36             if i == row:
37                 matrix[row][x] = inputText[ch]
38                 result[x] = inputText[ch].lower()
39                 ch += 1
40             if flag:
41                 i += 1
42             else:
43                 i -= 1
44             if i == key - 1 or i == 0:
45                 flag = not (flag)
46
47    for row in matrix:
48        print(*row)
49
50    return "".join(result).replace(" ", "")
51
52
```

```

53 def main() → None:
54     print(f'{'start':-^40}')
```

```

55     inputText: str = input("Enter your String :").strip().replace(" ", "")
56     key: int = int(input("Enter Key: "))
57
58     print("-" * 40)
59     if inputText.islower():
60         res = encrypt(inputText, key)
61         print("-" * 40)
62         print("Encrypt :", res)
63     else:
64         res = decrypt(inputText, key)
65         print("-" * 40)
66         print("Decrypt :", res)
67     print(f'{'end':-^40}')
```

```

68
69
70 if __name__ == "__main__":
71     main()
72 #####
73     #Practical 2B
74 #####
75 def simpleColumnar(inputText: str, column: int):
76     row: int = (len(inputText) / column).__ceil__()
77     inputText += "*" * (row * column - len(inputText))
78
79     if inputText.islower():
80         matrix: list[str] = [
81             inputText[i : i + column] for i in range(0, len(inputText), column)
82         ]
83
84         encrypt: str = ""
85         for text in matrix:
86             print(*text, sep=" ")
87
88         for i in range(column):
89             for text in matrix:
90                 encrypt += text[i].upper()
91
92         print('-'*40)
93         print("Encrypted:", encrypt)
94     else:
95         matrix: list[list[chr]] = [ list(" ") * column for r in range(row)]
96
97         i: int = 0
98         for c in range(column):
99             for r in range(row):
100                 matrix[r][c] = inputText[i]
101                 i+=1
102
103         for text in matrix:
104             print(*text, sep=" ")
105
106         decrypt: str = ""
```

```

107         for text in matrix:
108             decrypt += ''.join(text).lower()
109
110         print('-'*40)
111         print("Decrypted:", decrypt)
112
113
114 def main() → None:
115     while True:
116         print(f'{'start':-^40}')
117         inputText: str = input("Enter your String :").replace(" ", "")
118         column: int = int(input("Enter Column: "))
119         print("-" * 40)
120         simpleColumnar(inputText=inputText, column=column)
121         print(f'{'end':-^40}')
122
123 if __name__ == "__main__":
124     main()
125 #####
126         #Practical 2C
127 #####
128 def main() → None:
129     print(f'{'start':-^40}')
130     enkey: list[int] = map(int, input("Enter Your key: ").strip().split(" "))
131
132
133     enkey: list[list[int]] = list(map(list, enumerate(enkey, start=1)))
134     deque: list[list[int]] = sorted(enkey, key=lambda x: x[1])
135
136     enkey: list[int] = [x[1] for x in enkey]
137     deque: list[int] = [x[0] for x in deque]
138
139     print('-' * 40)
140     print(f'Encryption Key: ',*enkey)
141
142     print(f'Decryption Key: ',*deque)
143     print(f'{'end':-^40}')
144
145 if __name__ == '__main__':
146     main()
147 #####
148         #Practical 2D
149 #####
150 def hybridTransposition(inputText: str, enkey: list[list[int]]) → None:
151     enkey: list[list[int]] = list(map(list, enumerate(enkey, start=1)))
152     deque: list[list[int]] = sorted(enkey, key=lambda x: x[1])
153
154     col: int = len(enkey)
155     row: int = (len(inputText) / col).__ceil__()
156     inputText += "*" * ((row * col) - len(inputText))
157
158     enkey: list[int] = [x[1] for x in enkey]
159     deque: list[int] = [x[0] for x in deque]
160

```

```

161     result = ""
162
163     if inputText.islower():
164         print(f'Encrypted Key: ', enkey)
165         print('-'*40)
166
167         matrix: list[list[chr]] = [
168             inputText[x : x + col] for x in range(0, len(inputText), col)
169         ]
170         for text in matrix:
171             print(*text, sep=" ")
172         print('-'*40)
173
174         for e in enkey:
175             for l in matrix:
176                 result += l[e - 1].upper()
177
178         print(f'Encrypted: ', result)
179     else:
180         print(f'Decrypted Key: ', dekey)
181         print('-'*40)
182
183         matrix: list[list[chr]] = [ list(" ") * col for r in range(row)]
184
185         i: int = 0
186         for c in range(col):
187             for r in range(row):
188                 matrix[r][c] = inputText[i]
189                 i += 1
190
191         for text in matrix:
192             print(*text, sep=" ")
193         print('-'*40)
194
195         for l in matrix:
196             for d in dekey:
197                 result += l[d - 1].lower()
198
199         print(f'Decrypted: ', result)
200
201
202 def main() → None:
203     print(f'{'start':-^40}')
204     inputText: str = input("Enter your String :").strip().replace(" ", "")
205     enkey: list[int] = map(int, input("Enter Your key: ").strip().split(" "))
206     print('-'*40)
207     hybridTransposition(inputText, enkey)
208     print(f'{'end':-^40}')
209
210
211 if __name__ == "__main__":
212     main()

```