

## 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 import random
76 def simpleColumnar(inputText: str, column: int):
77     row: int = (len(inputText) / column).__ceil__()
78
79     if inputText.islower():
80         while(len(inputText) < row*column):
81             inputText += random.choice("").join(ch for ch in 'abcdefghijklmnopqrstuvwxyz' if
ch not in [*inputText])) * (row*column - len(inputText))
82             matrix: list[str] = [
83                 inputText[i : i + column] for i in range(0, len(inputText), column)
84             ]
85
86             encrypt: str = ""
87             for text in matrix:
88                 print(*text, sep=" ")
89
90             for i in range(column):
91                 for text in matrix:
92                     encrypt += text[i].upper()
93
94             print('-'*40)
95             print("Encrypted:", encrypt)
96     else:
97         while(len(inputText) < row*column):
98             inputText += random.choice("").join(ch for ch in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' if
ch not in [*inputText])) * (row*column - len(inputText))
99             matrix: list[list[chr]] = [ list(" ") * column for r in range(row)]
100
101             i: int = 0
102             for c in range(column):
103                 for r in range(row):
104                     matrix[r][c] = inputText[i]
105                     i+=1
106
107             for text in matrix:
108                 print(*text, sep=" ")
109
110             decrypt: str = ""

```

```

111         for text in matrix:
112             decrypt += ''.join(text).lower()
113
114         print('-'*40)
115         print("Decrypted:", decrypt)
116
117
118 def main() → None:
119     while True:
120         print(f'{'start':-^40}')
121         inputText: str = input("Enter your String :").replace(" ", "")
122         column: int = int(input("Enter Column: "))
123         print("-" * 40)
124         simpleColumnar(inputText=inputText, column=column)
125         print(f'{'end':-^40}')
126
127 if __name__ == "__main__":
128     main()
129 #####
130         #Practical 2C
131 #####
132 def main() → None:
133     print(f'{'start':-^40}')
134     enkey: list[int] = map(int, input("Enter Your key: ").strip().split(" "))
135
136
137     enkey: list[list[int]] = list(map(list, enumerate(enkey, start=1)))
138     dekey: list[list[int]] = sorted(enkey, key=lambda x: x[1])
139
140     enkey: list[int] = [x[1] for x in enkey]
141     dekey: list[int] = [x[0] for x in dekey]
142
143     print('-' * 40)
144     print(f'Encryption Key: ',*enkey)
145
146     print(f'Decryption Key: ',*dekey)
147     print(f'{'end':-^40}')
148
149 if __name__ == '__main__':
150     main()
151 #####
152         #Practical 2D
153 #####
154 import random
155 def hybridTransposition(inputText: str, enkey: list[list[int]]) → None:
156     enkey: list[list[int]] = list(map(list, enumerate(enkey, start=1)))
157     dekey: list[list[int]] = sorted(enkey, key=lambda x: x[1])
158
159     col: int = len(enkey)
160     row: int = (len(inputText) / col).__ceil__()
161
162     enkey: list[int] = [x[1] for x in enkey]
163     dekey: list[int] = [x[0] for x in dekey]
164
165     result = ""
166
167     if inputText.islower():

```

```

168         while(len(inputText) < row*col):
169             inputText += random.choice("".join(ch for ch in 'abcdefghijklmnopqrstuvwxyz' if
ch not in [*inputText])) * (row*col - len(inputText))
170
171         print(f'Encrypted Key: ', enkey)
172         print('-'*40)
173
174         matrix: list[list[chr]] = [
175             inputText[x : x + col] for x in range(0, len(inputText), col)
176         ]
177         for text in matrix:
178             print(*text, sep=" ")
179         print('-'*40)
180
181         for e in enkey:
182             for l in matrix:
183                 result += l[e - 1].upper()
184
185         print(f'Encrypted: ', result)
186     else:
187         while(len(inputText) < row*col):
188             inputText += random.choice("".join(ch for ch in 'ABCDEFGHIJKLMNOPQRSTUVWXYZ' if
ch not in [*inputText])) * (row*col - len(inputText))
189         print(f'Decrypted Key: ', dekey)
190         print('-'*40)
191
192         matrix: list[list[chr]] = [ list(" ") * col for r in range(row)]
193
194         i: int = 0
195         for c in range(col):
196             for r in range(row):
197                 matrix[r][c] = inputText[i]
198                 i+=1
199
200         for text in matrix:
201             print(*text, sep=" ")
202         print('-'*40)
203
204         for l in matrix:
205             for d in dekey:
206                 result += l[d - 1].lower()
207
208         print(f'Decrypted: ', result)
209
210
211 def main() → None:
212     print(f'{'start':-^40}')
213     inputText: str = input("Enter your String :").strip().replace(" ", "")
214     enkey: list[int] = map(int, input("Enter Your key: ").strip().split(" "))
215     print('-'*40)
216     hybridTransposition(inputText, enkey)
217     print(f'{'end':-^40}')
218
219
220 if __name__ == "__main__":
221     main()

```