1. Atbash Cipher

Encryption

Map each letter to its Atbash equivalent (reverse)

Plaintext Letter	Position	25 - Position	Atbash Letter
J	9	16	Q
A	0	25	Z
I	8	17	R
S	18	7	H
H	7	18	S
K	10	15	P
H	7	18	S
A	0	25	Z
N	13	12	M

Encryption Result: `JAISHKHAN` → `QZRHSPSZM`

Decryption

Because Atbash is symmetrical, decrypting means applying the same process again:

`QZRHSPSZM` → `JAISHKHAN`

---

2. Caesar Cipher (Key = 3)

Encryption

Shift each letter **forward by 3** positions (A=0,...,Z=25).

Plaintext Letter	Position	Position + Key	Cipher Letter
J	9	(9 + 3) = 12	M
A	0	(0 + 3) = 3	D
I	8	(8 + 3) = 11	L
S	18	(18 + 3) = 21	V

Plaintext Letter	Position	Position + Key	Cipher Letter
H	7	(7 + 3) = 10	K
K	10	(10 + 3) = 13	N
H	7	(7 + 3) = 10	K
A	0	(0 + 3) = 3	D
N	13	(13 + 3) = 16	Q

Encryption Result: `JAISHKHAN` → `MDLVKNKDQ`

Decryption

Shift each letter **backward by 3** positions.

Cipher Letter	Position	Position - 3	Plaintext Letter
M	12	(12 - 3) = 9	J
D	3	(3 - 3) = 0	A
L	11	(11 - 3) = 8	I
V	21	(21 - 3) = 18	S
K	10	(10 - 3) = 7	H
N	13	(13 - 3) = 10	K
K	10	(10 - 3) = 7	H
D	3	(3 - 3) = 0	A
Q	16	(16 - 3) = 13	N

Decryption Result: `MDLVKNKDQ` → `JAISHKHAN`

---

3. Vigenère Cipher (Key = KEY)

Encryption

1. Plaintext: `JAISHKHAN`
2. Key: `KEY` → repeat to match the plaintext: `KEYKEYKEY`
3. Convert letters to positions (A=0, ..., Z=25)

Plaintext	Position (P)	Key	Position (K)	P+K mod 26	Cipher
J	9	K	10	(9 + 10) = 19	T

Plaintext	Position (P)	Key	Position (K)	P+K mod 26	Cipher
A	0	E	4	(0 + 4) = 4	E
I	8	Y	24	(8 + 24) = 32 → 6	G
S	18	K	10	28 → 2	C
H	7	E	4	(7 + 4) = 11	L
K	10	Y	24	34 → 8	I
H	7	K	10	(7 + 10) = 17	R
A	0	E	4	(0 + 4) 4	E
N	13	Y	24	37 → 11	L

Encryption Result: `JAISHKHAN` → `TEGCLIREL`

Decryption

Subtract Key's position from Cipher's position to get plaintext position. If you get a negative number then add 26.

Cipher (C)	Key (K)	C−K (+26)	Plain
T = 19	K = 10	(19 - 10) = 9	J
E = 4	E = 4	(4 - 4) = 0	A
G = 6	Y = 24	(6 - 24) = -18 + 26 = 8	I
C = 2	K = 10	(2 - 10) = -8 + 26 = 18	S
L = 11	E = 4	(11 - 4) = 7	H
I = 8	Y = 24	(8 - 24) = -16 + 26 = 10	K
R = 17	K = 10	(17 - 10) = 7	H
E = 4	E = 4	(4 - 4) = 0	A
L = 11	Y = 24	(11 - 24) = -13 + 26 = 13	N

Decryption Result: `TEGCLIREL` → `JAISHKHAN`

---

4. Columnar Cipher (Key = NULL)

Encryption

Plaintext: `JAISHKHAN`, Length: 9
Key: `"NULL"`, Key length: 4 → rows needed = round up$(9/4)$ = **3**

Fill with an extra character (like `X`) to make it a full grid.

```
Columns:   N U L L

Grid:
J   A   I   S
H   K   H   A
N   X   X   X   ← padding with 'X' and blanks if needed
```

Now read the columns based on the key letters alphabetical order.

Column	Column Letters
N (3)	J H N
U (4)	A K X
L (1)	I H X
L (2)	S A X

Ciphertext (column-wise in key order): `I H X S A X J H N A K X`

Decryption

Ciphertext: `IHXSAXJHNAKX`, Length: 9
Key: `"NULL"`, Key length: 4 → rows needed = round up$(9/4)$ = **3**

Fill columns in the key order with the ciphertext. Reconstructed grid:

N (3)	U (4)	L (1)	L (2)
J	A	I	S
H	K	H	A
N	X	X	X

Decrypted Text: (Read row by row)
`J A I S H K H A N X X X` → remove padding → **JAISHKHAN**

---

5. Rail Fence Cipher (Key = 3)

Encryption

Plaintext: `MUHAMMAD JAISH KHAN`

- Remove spaces → `MUHAMMADJAISHKHAN`
- Length: 18 characters

Write the message in a zigzag pattern across **3 rails** (because key = 3) like this:

```
Rail 1: M   M   J   H   N
Rail 2:  U A M D A S K A
Rail 3:   H   A   I   H
```

To do this manually:
Index by index → assign to rails in the pattern:
Rail 1 → Rail 2 → Rail 3 → Rail 2 → Rail 1 → Rail 2 → Rail 3 → ...

Index	Char	Rail
0	M	1
1	U	2
2	H	3
3	A	2
4	M	1
5	M	2
6	A	3
7	D	2
8	J	1
9	A	2
10	I	3
11	S	2
12	H	1
13	K	2
14	H	3
15	A	2
16	N	1

Encrypted Text: (Now read rails row by row):

- **Rail 1**: M M J H N → `MMJHN`
- **Rail 2**: U A M D A S K A N → `UAMDASKA`

- **Rail 3**: H A I H → `HAIH`

Final Ciphertext (Rail 1 + Rail 2 + Rail 3): `MMJHNUAMDASKANHAIH`

Decryption

Ciphertext: `MMJHNUAMDASKANHAIH`
Rails: `3`

First, we need to figure out **how characters were distributed** across the rails when encrypted. This is based on a repeating zigzag pattern.

The pattern for 3 rails repeats every 4 characters like so:

```
Rail 1: 0       4       8       12      16
Rail 2:   1   3   5   7   9   11  13  15  17
Rail 3:     2       6       10      14
```

So:

- Rail 1 gets positions: 0, 4, 8, 12, 16 (5 characters)
- Rail 2 gets: 1, 3, 5, 7, 9, 11, 13, 15, 17 (9 characters)
- Rail 3 gets: 2, 6, 10, 14 (4 characters)

Fill Rails With Ciphertext Letters from left to right:

- Rail 1 (5 letters): `MMJHN`
- Rail 2 (9 letters): `UAMDASKAN`
- Rail 3 (4 letters): `HAIH`

Rebuild the Zigzag Path to Read Decrypted Text. Follow the zigzag pattern and **pick letters rail-wise in that order**.

Zigzag order (same as above):
`1 → 2 → 3 → 2 → 1 → 2 → 3 → 2 → 1 → 2 → 3 → 2 → 1 → 2 → 3 → 2 → 1 → 2`

Go one by one:
Final Decrypted Text: `MUHAMMADJAISHKHAN`