Title: Rail Fence Cipher Encryption & Decryption.

Aims:

- Encrypting plaintexts by the technique of Rail Fence Cipher.
- Decrypting ciphertexts by the Rail Fence Cipher Method.

Tasks:

- Create a function of Rail Fence Cipher technique to Encrypt the given plaintexts.
- Create a function of Rail Fence Cipher technique to Decrypt the ciphertexts.

Activities:

1. Create a function of Rail Fence Cipher technique to Encrypt the given plaintexts

```
String getEncryptedData(String data) {
	char[] encrypted = new char[data.length()];
	int n = 0;


	for(int k = 0 ; k < numRails; k ++) {
		int index = k;
		boolean down = true;
		while(index < data.length() ) {
			//System.out.println(k + " " + index+ " "+ n );
			encrypted[n++] = data.charAt(index);

			if(k == 0 || k == numRails - 1) {
				index = index + 2 * (numRails - 1);
			}
			else if(down) {
				index = index +  2 * (numRails - k - 1);
				down = !down;
			}
			else {
				index = index + 2 * k;
				down = !down;
			}
		}
	}
	return new String(encrypted);
	}
```

2. Create a function of Rail Fence Cipher technique to Decrypt the ciphertexts

```
String getDecryptedData(String data) {
	char[] decrypted = new char[data.length()];
	int n = 0;
	for(int k = 0 ; k < numRails; k ++) {
		int index = k;
		boolean down = true;
		while(index < data.length() ) {
			//System.out.println(k + " " + index+ " "+ n );
```

```
                        decrypted[index] = data.charAt(n++);

                        if(k == 0 || k == numRails - 1) {
                              index = index + 2 * (numRails - 1);
                        }
                        else if(down) {
                              index = index +  2 * (numRails - k - 1);
                              down = !down;
                        }
                        else {
                              index = index + 2 * k;
                              down = !down;
                        }
                  }
            }
            return new String(decrypted);
            }
```

Program for Rail Fence Cipher Encryption and Decryption:

```
public class RailFenceCipher {
      int numRails;

      public RailFenceCipher(int numRails) {
            this.numRails = numRails;
      }

      String getDecryptedData(String data) {
            char[] decrypted = new char[data.length()];
            int n = 0;
            for(int k = 0 ; k < numRails; k ++) {
                  int index = k;
                  boolean down = true;
                  while(index < data.length() ) {
                        //System.out.println(k + " " + index+ " "+ n );
                        decrypted[index] = data.charAt(n++);

                        if(k == 0 || k == numRails - 1) {
                              index = index + 2 * (numRails - 1);
                        }
                        else if(down) {
                              index = index +  2 * (numRails - k - 1);
                              down = !down;
                        }
                        else {
                              index = index + 2 * k;
                              down = !down;
                        }
                  }
            }
            return new String(decrypted);
      }
```

```
	String getEncryptedData(String data) {
		char[] encrypted = new char[data.length()];
		int n = 0;


		for(int k = 0 ; k < numRails; k ++) {
			int index = k;
			boolean down = true;
			while(index < data.length() ) {
				//System.out.println(k + " " + index+ " "+ n );
				encrypted[n++] = data.charAt(index);

				if(k == 0 || k == numRails - 1) {
					index = index + 2 * (numRails - 1);
				}
				else if(down) {
					index = index +  2 * (numRails - k - 1);
					down = !down;
				}
				else {
					index = index + 2 * k;
					down = !down;
				}
			}
		}
		return new String(encrypted);
	}

public static void main(String[] args) {
		String data = "PINEAPPLE";

				RailFenceCipher railFenceCipher = new RailFenceCipher(2);

				String encrypted =railFenceCipher.getEncryptedData(data);
				System.out.println(encrypted);

				String decrypted =
railFenceCipher.getDecryptedData(encrypted);
				System.out.println(decrypted);
			}
}
```