Semester Project Title: CryptX – File Encryption & Compression Tool

Student Details:

	Student Name	Student Reg #	Student Degree
Student-1	Sheikh Nabeel	2024588	BSSE
Student-2	Hamza Farooq	2024390	BSSE
Student-3	Muhammad Daniyal	2024618	BSSE

1. Main Features

1. Huffman-Based File Compression
2. Lossless File Decompression
3. XOR File Encryption
4. XOR File Decryption
5. Text Encryption in HEX Format
6. Text Decryption from HEX Format
7. View Compressed File Content
8. Operation History Using Stack
9. Clear Operation History
10. Menu-Driven Console Interface

2. Types of Users & Requirements

1. User

1.1 Will be able to compress text files using Huffman coding.
1.2. Will be able to decompress previously compressed files.
1.3. Will be able to encrypt files using XOR encryption with a user-defined key.
1.4. Will be able to decrypt encrypted files using the correct key.
1.5. Will be able to encrypt plain text and receive output in hexadecimal format.
1.6. Will be able to decrypt hexadecimal encrypted text back to original form.
1.7. Will be able to view the content of compressed files without saving them.
1.8. Will be able to view the history of performed operations.
1.9. Will be able to clear the operation history when required.

2. System

2.1. Will efficiently perform file compression and decompression without data loss.
2.2. Will ensure correct encryption and decryption using XOR logic.
2.3. Will maintain operation history using a stack data structure.
2.4. Will validate user input such as file paths, encryption keys, and HEX text.
2.5. Will provide a menu-driven interface for smooth user interaction.

3. Requirements Breakdown

1. File Compression (Huffman Coding):

1.1. The system shall allow the user to select a text file for compression.
1.2. The system shall generate Huffman codes and compress the file.
1.3. The system shall save the compressed output in a binary file.
1.4. The system shall record the operation in the history stack.

2. File Decompression:

2.1. The system shall read the compressed file and reconstruct the Huffman tree.
2.2. The system shall decode the file and restore original content.
2.3. The system shall save or display the decompressed data.

3. File Encryption

3.1. The system shall encrypt a selected file using XOR encryption.
3.2. The system shall require a user-defined key for encryption.
3.3. The system shall save the encrypted file with a new extension.

4. File Decryption

4.1. The system shall decrypt the encrypted file using the correct key.
4.2. The system shall restore the original file content.

5. Text Encryption

5.1. The system shall encrypt user-input text using XOR encryption.
5.2. The system shall display encrypted output in hexadecimal format.

6. Text Decryption

6.1. The system shall decrypt hexadecimal encrypted text using the key.
6.2. The system shall display the original text.

7. Operation History

7.1. The system shall store all operations using a stack.
7.2. The system shall allow users to view and clear the history.

8. User Interface

8.1. The system shall provide a menu-driven console interface.
8.2. The system shall handle invalid user inputs properly.

4. Features to Codding Matrix

Sr #	Feature Name	DSA Concept Used	Operation Performed	Complexity Analysis (Approximate)	No. of Variables & Objects Created	Functions Created	Line of Code Written
1	File Compression	Trees, Priority Queue, Map	Huffman Tree + Encoding	O(n log n)	6	4	80
2	File Decompression	Trees, Priority Queue, Map	Huffman Tree + Decoding	O(n log n)	5	3	70
3	File Encryption (XOR)	Arrays, Loops	XOR Encryption	O(n)	4	1	45
4	File Decryption (XOR)	Arrays, Loops	XOR Decryption	O(n)	4	1	50
5	Text Encryption (HEX)	Arrays, Loops	XOR + HEX Conversion	O(n)	3	1	35
6	Text Decryption (HEX)	Arrays, Loops	XOR + HEX Parsing	O(n)	3	1	40
7	Operation History Stack	Linked List, Stack	Push, Pop, Display	O(n)	3	3	70
8	Menu Interface	Loops, Switch-Case	Navigation + Function Calls	O(1) per operation	2	1	30

5. Project Screenshots

Figure 5.1: CryptX Main Menu Screen

Figure 5.2: File Compression Using Huffman Coding

Figure 5.3: File decompression Using Huffman Coding

Figure 5.4: Display Compressed Content

```
Choice: 4
Enter input file path: image.jpg
Enter key: image123
Added to history: Encrypt File
File encrypted successfully!
Encrypted file: image.jpg.enc
```

image	11/19/2025 9:57 PM	JPG File	1,008 KB
image.jpg	12/19/2025 4:31 PM	ENC File	1,008 KB

Figure 5.5: File Encryption Using XOR Technique

```
Choice: 5
Enter encrypted file path: image.jpg.enc
Enter key: image123
Added to history: Decrypt File
File decrypted successfully!
Decrypted file: image_decrypted.jpg
```

image	11/19/2025 9:57 PM	JPG File	1,008 KB
image.jpg	12/19/2025 4:31 PM	ENC File	1,008 KB
image_decrypted	12/19/2025 4:33 PM	JPG File	1,008 KB

Figure 5.6: File Decryption Using XOR Technique

```
Choice: 6
Enter text to encrypt: The Sword of Islam: The Life and Legacy of Salahuddin Ayyubi
Enter key: salahuddin_ayyubi
Added to history: Encrypt Text
Encrypted Text (HEX): 270909413b020b160d4e30075930060e081e5b4c350010442800083a41181711422516060d0211550b02493d3e0d181100060d1a0f4c20110c1106
00
Choice: 7
Enter HEX encrypted text: 270909413b020b160d4e30075930060e081e5b4c350010442800083a41181711422516060d0211550b02493d3e0d181100060d1a0f4c20110c
110600
Enter key: salahuddin_ayyubi
Added to history: Decrypt Text
Decrypted Text: The Sword of Islam: The Life and Legacy of Salahuddin Ayyubi
```

Figure 5.7: Text Encryption & Decryption Using XOR Technique

Figure 5.8: Operation History Stack Display

6. **GitHub Repository Link:**

   https://github.com/Nabeel-8090/CryptX-DSA-Project

THE END