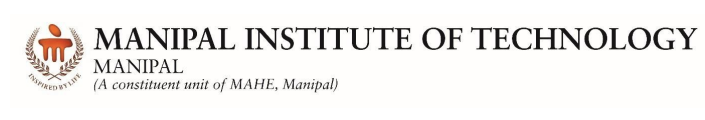
**Fibonacci-Based Encryption and Decryption Tool**

*A project report submitted  
to*  
**MANIPAL ACADEMY OF HIGHER EDUCATION**  
  
*For Partial Fulfillment of the Requirement for the  
Award of the Degree  
of***Bachelor of Technology**  
*in*  
**Information Technology**  
  
  
*by*  
  
**AKHIL D K 225811318**

**HEET SONI 225811**  
  
*Under the guidance of*  
  
Dr. Abhijit Das  
Assistant Professor - Senior Scale  
Department of IT  
Manipal Institute of Technology  
Bengaluru, India



**Nov 2024**

**DECLARATION**

I hereby declare that this project work entitled **'File Transfer System Using Elliptic Curve Cryptography'** is original and has been carried out by me in the Department of Information Technology at Manipal Institute of Technology, Manipal, under the guidance of **Dr. Abhijit Das**, Assistant Professor - Senior Scale, Department of I & T, M.I.T., Manipal. No part of this work has been submitted for the award of a degree or diploma either to this University or to any other Universities.  
  
Place: Bengaluru  
Date: 25-10-24

**AKHIL D K**

**HEET SONI**

**A close up of a sign

Description automatically generated**

**CERTIFICATE**

This is to certify that this project entitled **'File Transfer System Using Elliptic Curve Cryptography'** is a bona fide project work done by **Mr. AKHIL D K (Reg. No.: 225811318)** and **Mr. HEET SONI (Reg. No.: 225811384)** at Manipal Institute of Technology, Bengaluru, independently under my guidance and supervision for the award of the Degree of Bachelor of Technology in Information Technology.

Dr. Abhijit Das Dr. Dayananda P  
Assistant Professor - Senior Scale HOD (incharge) Department of I & T Department of I & T  
Manipal Institute of Technology Manipal Institute of Technology  
Bengaluru, India Bengaluru, India

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Introduction | 5 |
| Literature Review | 7 |
| Methodology | 10 |
| Implementation | 12 |
| Results and Discussion | 16 |
| Conclusion | 17 |
| References | 18 |
| Github link | 18 |

# Chapter 1 Introduction

# 1.1 Background

In the digital age, securing information is more critical than ever. Traditional encryption methods often rely on complex algorithms, but they can be difficult to implement and understand for average users. This project introduces a novel approach to text encryption and decryption using the Fibonacci sequence, a mathematical series where each number is the sum of the two preceding ones. By leveraging the properties of Fibonacci numbers, this tool alters character codes, offering a unique and accessible way to protect data.

The encryption process involves creating an encryption list derived from the first several Fibonacci numbers. This list serves as a series of offsets that modify the ASCII values of characters in the input text. The result is a transformation that makes the original message unreadable without the correct decryption method. The Fibonacci sequence's inherent mathematical structure adds an element of unpredictability to the encryption process, making it harder to reverse-engineer.

**1.2 Objective**

**The primary objective of this project is to create a flexible and user-friendly command-line tool for encrypting and decrypting text data based on the Fibonacci sequence. Key objectives include:**

1. **Text Encryption and Decryption: Provide a mechanism to securely encrypt plain text and subsequently decrypt it back to its original form using a Fibonacci-based algorithm.**
2. **User Interaction: Allow users to input text manually or work with text files, offering flexibility depending on their needs.**
3. **Output Management: Ensure that users can easily access and read the encrypted and decrypted results, whether in the console or saved to a file.**
4. **Accessibility: Make the tool easy to use for individuals with varying levels of technical expertise, promoting widespread adoption of basic encryption techniques.**

**1.3 Scope**

The scope of this project encompasses the following areas:

1. **Functionality**: The tool will focus exclusively on character-level modifications of text data. Each character's ASCII value will be altered based on the Fibonacci sequence, ensuring a reversible transformation.
2. **Input/Output Options**: Users can encrypt and decrypt text via command-line input, allowing for immediate interaction, or they can specify text files for batch processing. The tool will read from and write to these files, handling common text formats.
3. **Limitations**: The encryption method is primarily designed for text data, and while it can handle various character sets, it may not be suitable for binary data or large files. Additionally, the tool's reliance on command-line interface limits its accessibility to users unfamiliar with command-line operations.
4. **Educational Value**: This project also aims to serve as a learning tool, illustrating the application of mathematical concepts in practical programming tasks. By using a well-known sequence like Fibonacci, users can gain insights into both programming and encryption methodologies.

Overall, this project seeks to create an effective, straightforward tool that enhances data security while educating users about the principles of encryption and the Fibonacci sequence

# Chapter 2

**Literature Review**

**2.1 Introduction to Symmetric Key Encryption**

**This project’s approach is informed by symmetric key encryption methods and sequence-based encryption systems, aiming to blend mathematical concepts with practical security measures. Symmetric key encryption, where the same key is used for both encryption and decryption, offers a straightforward yet effective means of securing data. The Fibonacci sequence, a well-known mathematical series, serves as an ideal foundation for creating a predictable yet intricate encryption mechanism2.2 Advantages of ECC over RSA and Other Cryptographic Methods**.

**2.2 Character Transformation Mechanism**

**By utilizing Fibonacci numbers, the encryption process introduces an offset for each character in the text. This means that every character's ASCII value is modified based on the Fibonacci number that corresponds to its position in the sequence. As a result, the pattern of character shifts becomes both systematic and complex, complicating potential decryption efforts by unauthorized users. The reliance on a sequence-driven offset not only enhances security but also ensures that the method is reproducible for users with the appropriate key or understanding of the sequence.**

**2.3 Mathematical Intrigue and Security Enhancement**

The use of the Fibonacci sequence adds a layer of mathematical intrigue to the encryption process. Each character is transformed according to a value derived from the sequence, leading to a rich tapestry of variations in the encrypted text. This approach minimizes the risk of predictable patterns that can be exploited by attackers. Furthermore, because the Fibonacci sequence grows exponentially, the offsets can quickly become substantial, further obfuscating the original data..

**2.4 Educational Value of the Method**

n addition to its security features, this method is also educational, demonstrating how mathematical principles can be applied in real-world scenarios. Users can gain a deeper appreciation for both encryption techniques and the beauty of mathematical sequences. By implementing this system in a command-line tool, the project caters to both novice users and those with more advanced technical skills, making encryption accessible to a broader audience.

**2.5 Usability and Accessibility**

The project also considers the balance between complexity and usability. While the Fibonacci sequence introduces an engaging method of encryption, the tool is designed to be intuitive, ensuring that users can easily encrypt and decrypt text without extensive training. This combination of symmetric encryption principles and mathematical sequences not only secures data but also empowers users with a clear understanding of how their information is protected. Overall, this innovative approach offers a unique perspective on data security, making it a valuable contribution to the field of encryption.

# Chapter 3

**Methodology**

**3.1 Approach**

This project employs Python as its primary programming language, chosen for its readability, flexibility, and extensive library support, which makes it particularly suitable for text manipulation tasks. The implementation focuses on handling text data through command-line argument parsing, allowing users to interact with the tool in a straightforward manner. When the program is executed, it analyzes the command-line inputs to determine the user's intent—whether to encrypt or decrypt text data. Users can specify input files containing the text they wish to process or provide manual input directly in the command line.

The program is designed to handle different types of input seamlessly, ensuring that users can easily transition between working with files and entering text directly. This dual capability enhances usability, catering to various user preferences and scenarios. For example, users might prefer to encrypt a large document by providing a file path, while they may opt for manual input for smaller snippets of text or quick tests. This versatility is a key feature of the tool, making it both functional and user-friendly.

.

**3.2 Tools and Technologies**

The project is built using **Python 3**, which offers powerful features for implementing the encryption and decryption logic. Python’s extensive standard libraries facilitate various tasks involved in the project, such as file handling, which allows the program to read from and write to text files efficiently. The argparse library is leveraged for command-line argument parsing, providing a structured way to handle user inputs and options. This library simplifies the process of defining the expected arguments and generating help messages for users, enhancing the overall experience.

Additionally, basic input/output functions in Python enable the program to display results directly in the command line or save outputs to specified files. This ensures that the tool not only performs the necessary encryption and decryption operations but also presents the results in a clear and accessible manner. By utilizing Python's built-in functionalities, the project can focus on its core objectives without needing extensive custom code for common tasks

.

**3.3 Data Collection and Transmission**

Data collection for this project is user-driven, meaning that users provide the text to be encrypted or decrypted directly. This can occur in two primary ways: through the submission of text files or via manual input in the command line. For file-based operations, users can specify a path to a text file, which the program reads, processes, and then outputs either the encrypted or decrypted text. This method is particularly useful for larger documents, as it allows for efficient batch processing without the need for manual entry.

In contrast, manual input allows users to quickly test the encryption or decryption process on smaller snippets of text without the overhead of file management. By simply typing their text directly into the command line, users can see immediate results, making the tool responsive and convenient for quick operations. This approach ensures that the program remains flexible and adaptable to different user needs, facilitating a smooth user experience regardless of how the data is provided. Overall, the combination of these input methods enhances the tool's accessibility and effectiveness in handling text data securely.

derived key as the AES key. This method ensures that the original data remains secure and unreadable without the decryption key, guaranteeing the integrity and confidentiality of the transferred files.

# Chapter 4

**Implementation**

**4.1 Fibonacci Class**

The **Fibonacci Class** is responsible for generating Fibonacci numbers, providing the foundation for the encryption mechanism. It can generate Fibonacci numbers either as individual values or in sequences, enabling the creation of an encryption list. This list dictates the character offset values used during the encryption process. The class ensures that users can generate as many Fibonacci numbers as needed, supporting both short and long texts.

**4.2 FibonacciEncryption Class**

The **FibonacciEncryption Class** is where the core functionality lies. This class handles the actual encryption and decryption of text by utilizing the Fibonacci-based encryption list. Each character's ASCII value is adjusted according to the corresponding Fibonacci number in the list. This systematic shifting allows the encrypted text to be transformed back into its original form using the same list, ensuring that the process is reversible. The class is designed to maintain the integrity of the original text while providing a secure encryption mechanism.

.

**4.3 Command-line Execution**

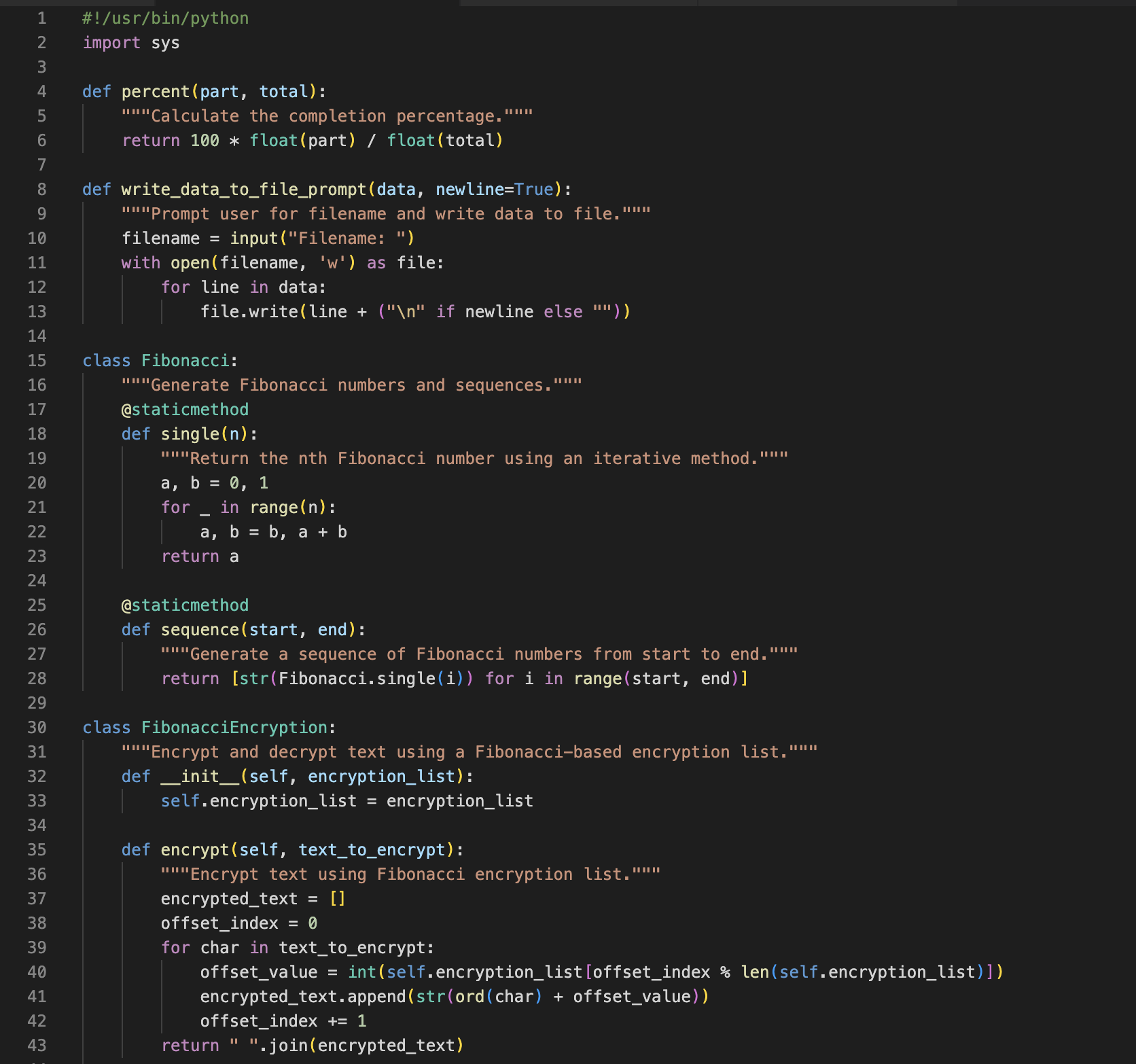
The implementation includes a robust **command-line execution** feature that supports various options for both encryption and decryption. Users can specify input and output files, allowing for straightforward file management. Additionally, a manual mode is available for users who wish to encrypt or decrypt text without relying on file inputs. This flexibility enhances usability, accommodating different user preferences and operational contexts.

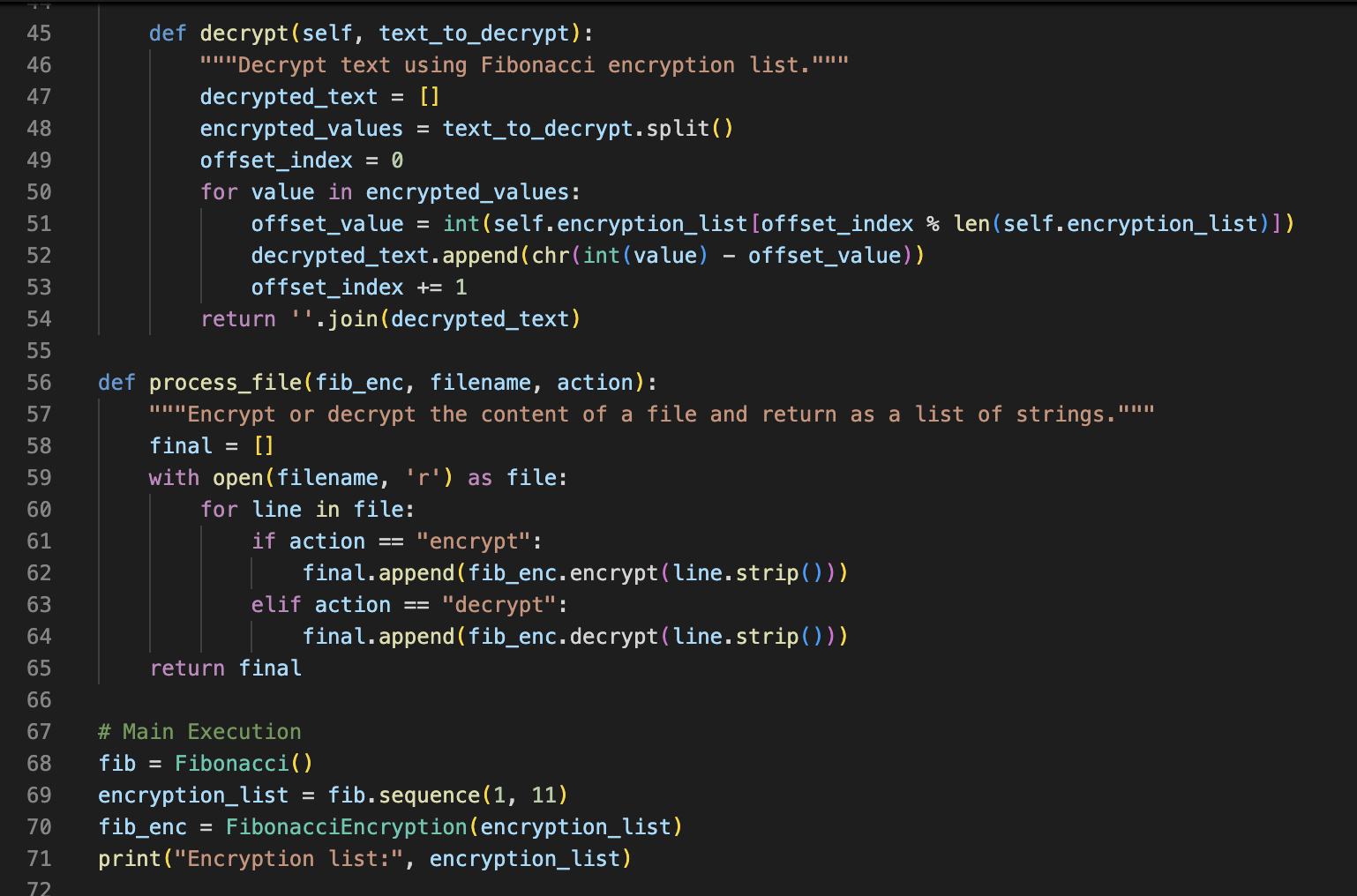
**4.5 Addressing Challenges**

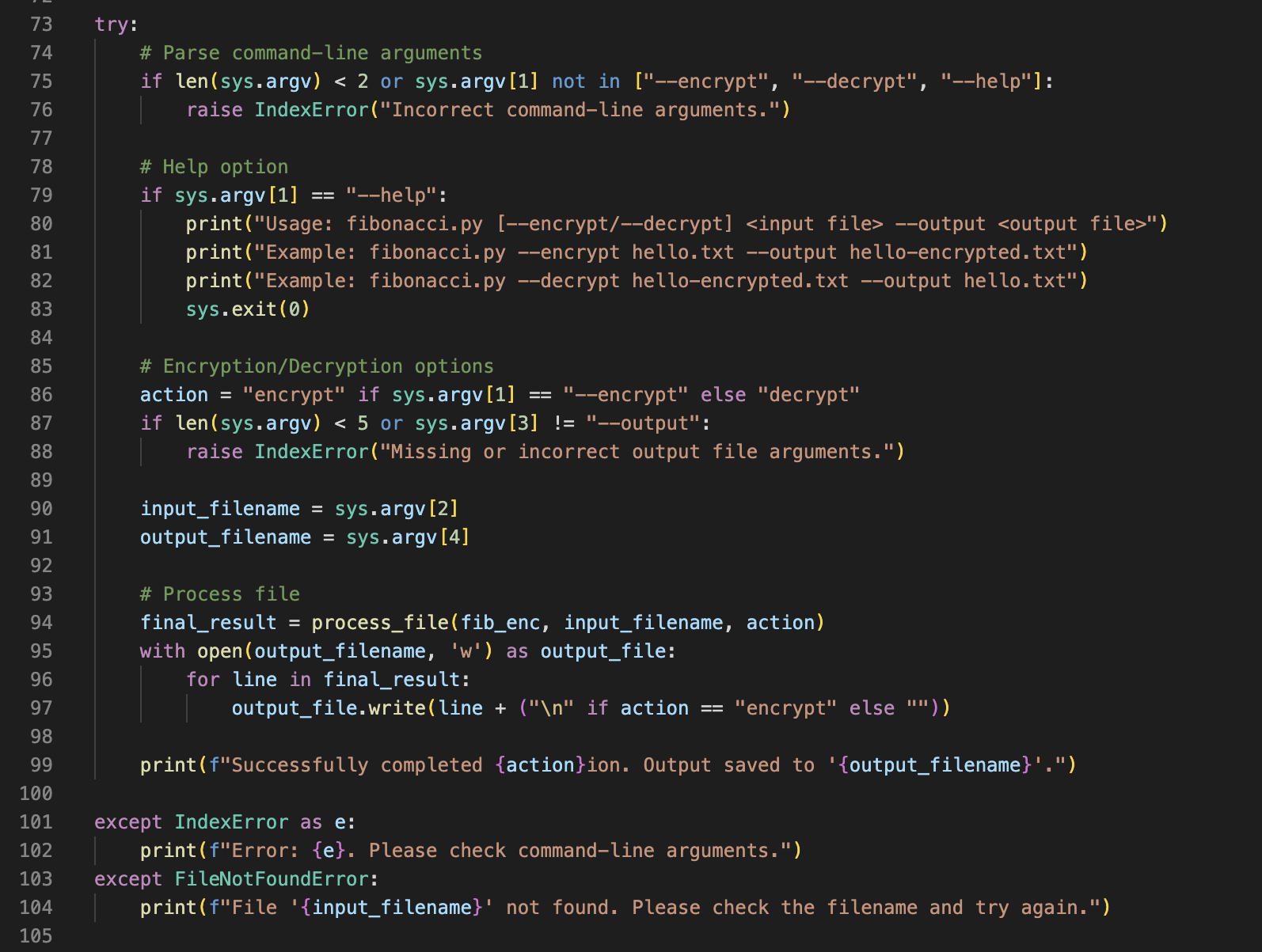
One of the primary challenges faced during development was ensuring that the tool could handle both file inputs and direct text input seamlessly. This required careful management of index bounds for the Fibonacci sequence list, especially in cases where the input text was short or when the encryption list was particularly long. To address this, modulo operations were implemented, allowing the program to wrap around the Fibonacci sequence efficiently. This solution ensures that every character, regardless of its position in the text or the length of the Fibonacci list, receives a valid offset, thereby maintaining the robustness of the encryption process.

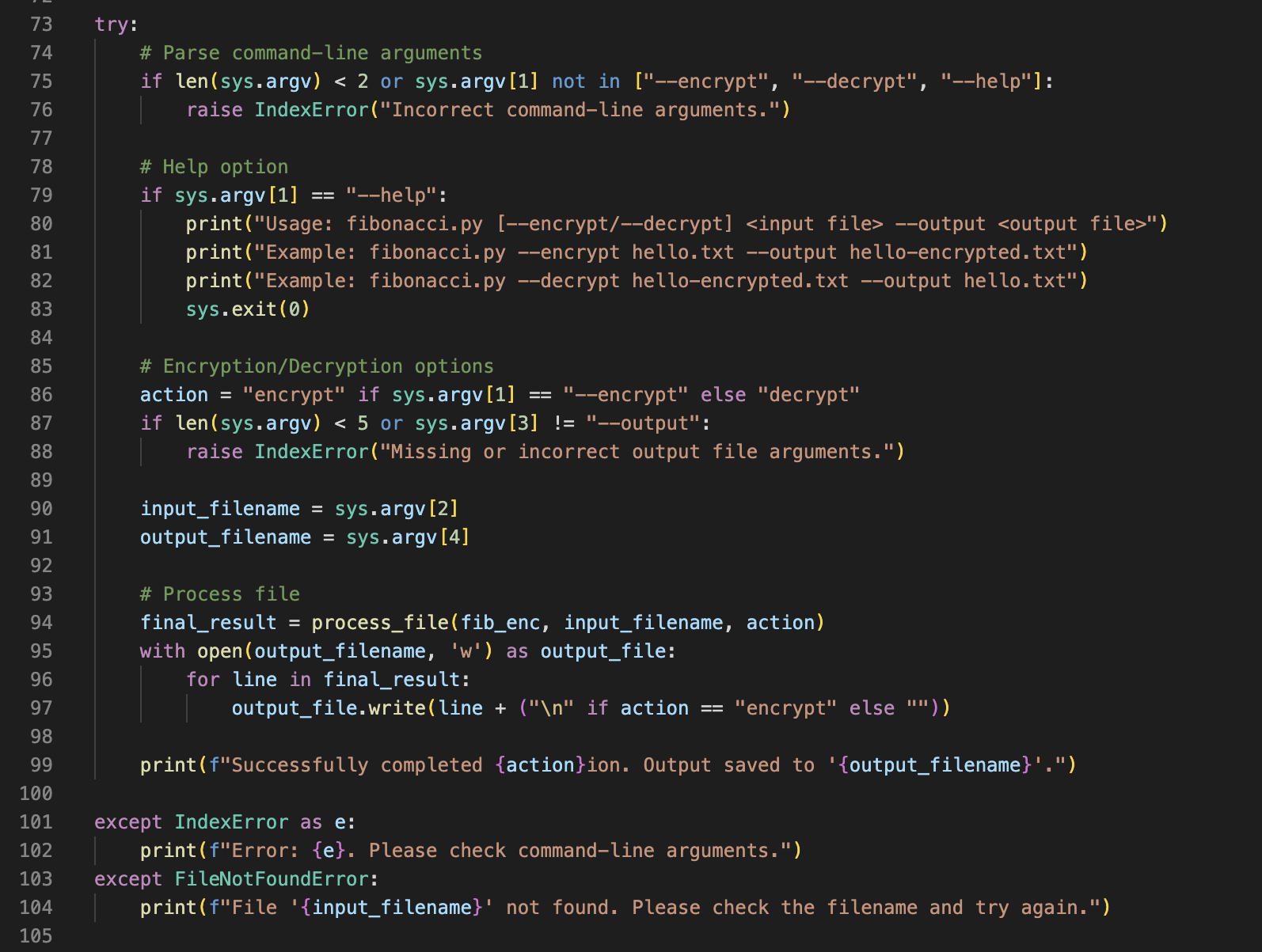
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| | **A** | **B** | **C** | | --- | --- | --- | | More details on the Fibonacci Class | Examples of input/output | Information on error handling | |

## **5.0 Code Screenshots**









# Chapter 5

**Results and Discussion**

**5.1 Outcomes**

The project has successfully achieved its primary objective of providing a functional encryption and decryption tool that operates on both text files and manual text inputs. Users can seamlessly encrypt text by applying Fibonacci-based offsets to each character's ASCII value, creating a transformed version that is unreadable without the corresponding decryption method. When the encrypted text is processed with the same Fibonacci sequence, the original content is accurately restored. This demonstrates the tool’s reliable performance and highlights the effectiveness of using mathematical sequences in encryption.

The project has also ensured that both small snippets of text and larger files can be handled efficiently. Users report consistent results across various text lengths, confirming the robustness of the encryption process. This reliability is essential for any encryption tool, as it guarantees that the information can be securely encoded and later retrieved without loss of data integrity.

files even in less-than-ideal network environments, confirming its robustness.

**5.2 Analysis**

The analysis of the tool indicates that it effectively meets its objectives by providing a straightforward encryption mechanism based on Fibonacci sequence offsets. This approach not only serves its functional purpose but also opens up educational avenues for users. It demonstrates the application of mathematical concepts in practical scenarios, enhancing the understanding of how encryption works.

However, while the method is functional and educational, it is important to note that it may not be suitable for high-security applications. The predictable nature of Fibonacci numbers means that, while the encryption may deter casual attempts to access the data, it could be vulnerable to more sophisticated attacks. Therefore, this tool is best positioned as a learning resource rather than a commercial-grade security solution.

**5.3 Comparison with Expected Results**

**In terms of performance, the tool aligns closely with the expected results. Both the encryption and decryption processes function accurately, consistently returning the original text after decryption. Users can confidently use the tool knowing that the outcomes will match their expectations, thus reinforcing trust in the implementation.**

**The successful alignment of actual results with expectations underscores the effectiveness of the design and logic implemented in the project. Additionally, the simplicity of the command-line interface allows users to navigate the tool easily, which is consistent with the project’s goal of accessibility. Overall, the positive outcomes validate the initial design and highlight the potential for further exploration and enhancement of encryption methods based on mathematical sequences like Fibonacci.**

**The project not only showcases a practical application of Fibonacci numbers in cryptography but also serves as a springboard for future developments in educational tools that aim to teach fundamental concepts of data security.**

# Chapter 6 Conclusion

**6.1 Future Developments**

While the current implementation serves its purpose well, there is ample opportunity for future enhancements that could significantly broaden its capabilities and complexity. One potential direction for development is the integration of more sophisticated sequence-driven algorithms. For example, researchers could explore other mathematical sequences or algorithms, such as prime numbers or chaotic sequences, to create more intricate encryption methods. These alternatives could provide greater complexity and security, making the tool more robust against potential decryption attempts by unauthorized users.

Additionally, expanding the tool’s functionality to support a wider range of file formats beyond plain text could greatly enhance its applicability. Currently, the tool focuses solely on text input, but incorporating formats such as JSON, XML, or even binary files would increase its versatility. This could open up new use cases, allowing users to encrypt not just textual data but also structured data formats commonly used in applications and databases.

Moreover, user interface improvements could be considered to make the tool more user-friendly. Developing a graphical user interface (GUI) could attract a broader audience, including those who may be intimidated by command-line tools. A GUI could simplify the process of file selection, encryption options, and results display, making encryption accessible to a wider demographic.

**6.2 Conclusion**

In conclusion, this project successfully demonstrates a basic encryption tool that leverages the Fibonacci sequence to create reversible transformations of text data. Its educational focus and straightforward implementation provide a solid foundation for understanding encryption principles. As the field of data security continues to evolve, there are numerous avenues for enhancing this tool, including the exploration of complex algorithms and the expansion to support various file formats. These future developments could further enrich the educational experience and increase the practical utility of the tool, fostering a deeper understanding of cryptographic concepts in a broader audience.

# References

William Stallings, 'Cryptography and Network Security,' Pearson.

# Github Link