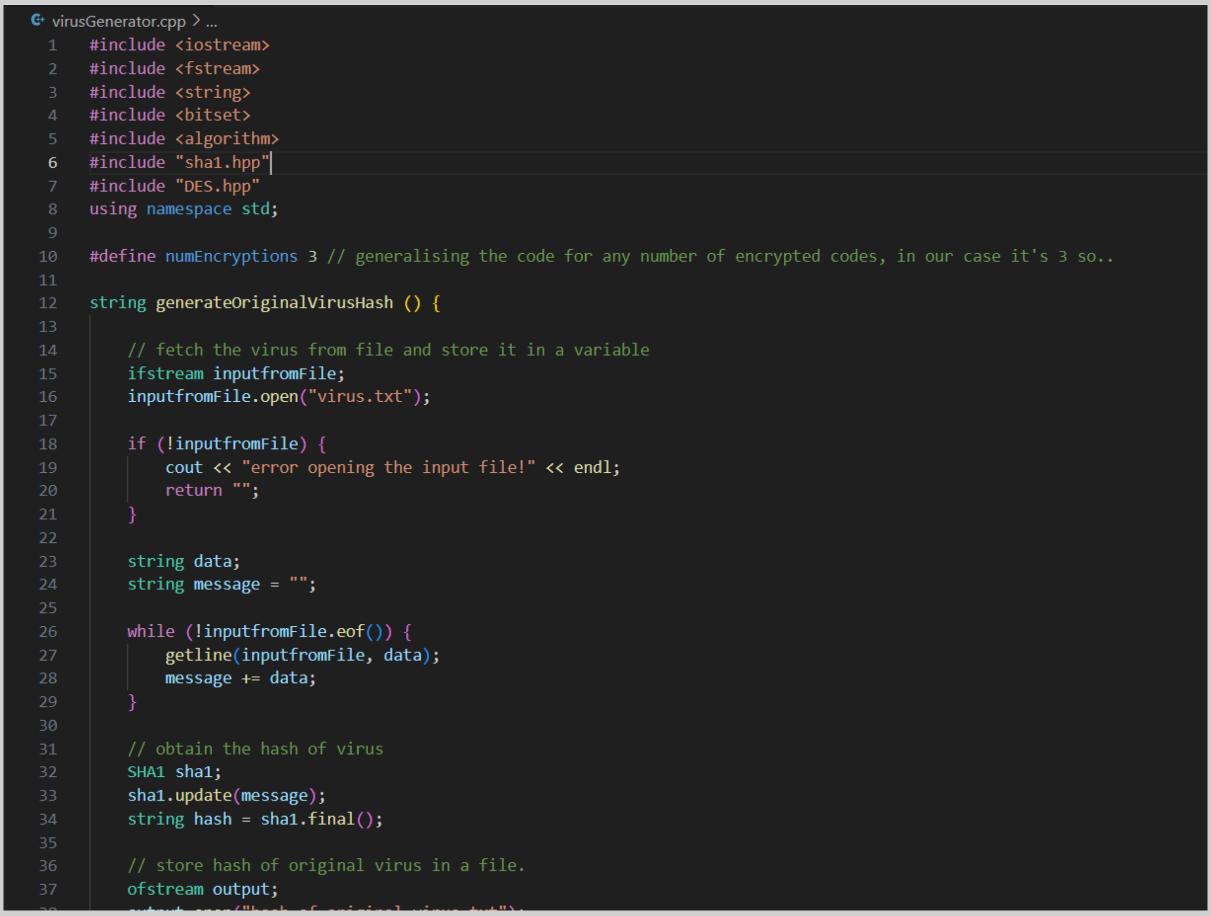
Muhammad Hammad (23K-2005)

# Cyber Security

## Assignment 03

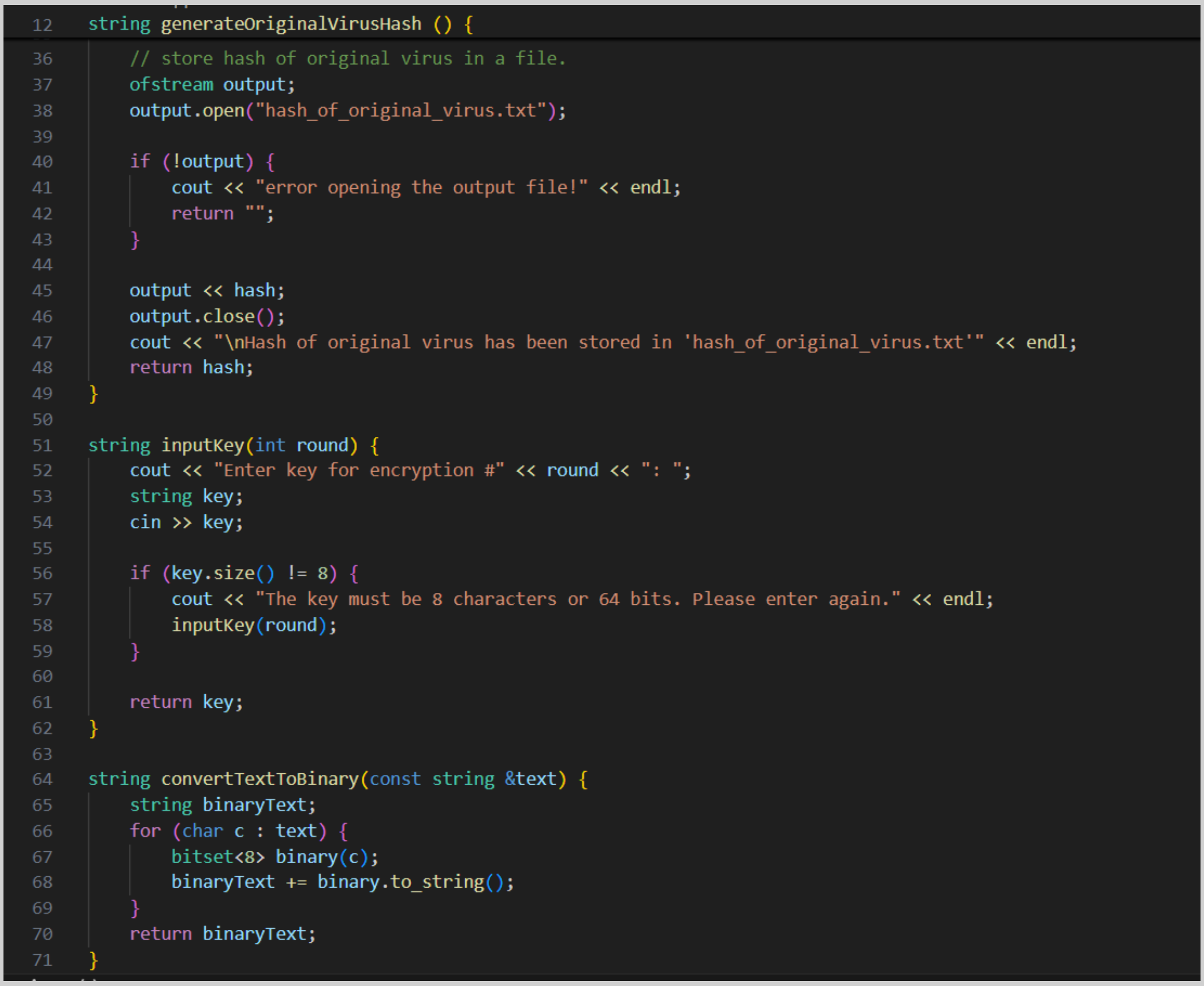
## Code:



virusGenerator.cpp > ...

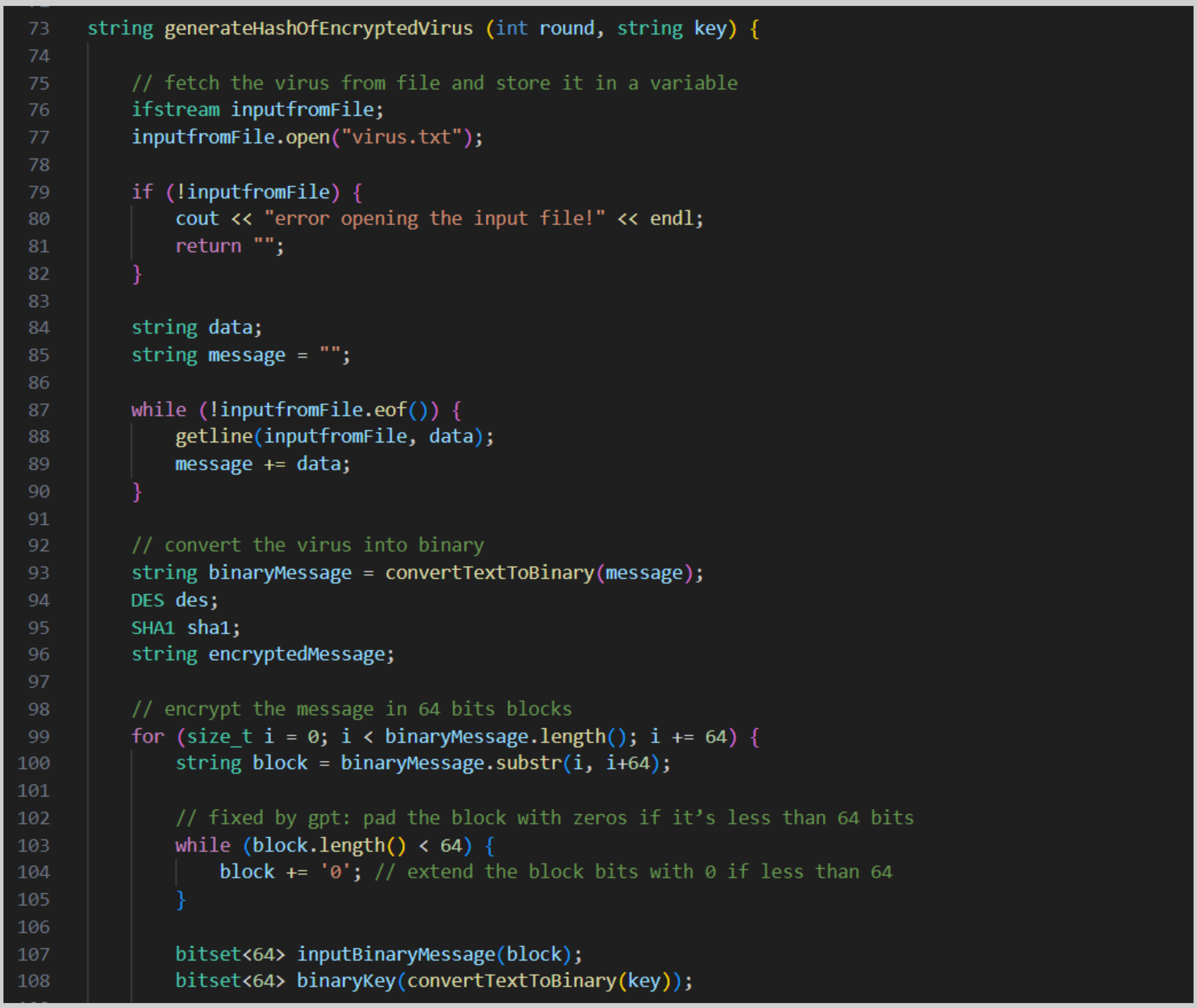
```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <bitset>
#include <algorithm>
#include "sha1.hpp"
#include "DES.hpp"
using namespace std;

#define numEncryptions 3 // generalising the code for any number of encrypted codes, in our case it's 3 so..

string generateOriginalVirusHash () {

    // fetch the virus from file and store it in a variable
    ifstream inputfromFile;
    inputfromFile.open("virus.txt");

    if (!inputfromFile) {
        cout << "error opening the input file!" << endl;
        return "";
    }

    string data;
    string message = "";

    while (!inputfromFile.eof()) {
        getline(inputfromFile, data);
        message += data;
    }

    // obtain the hash of virus
    SHA1 sha1;
    sha1.update(message);
    string hash = sha1.final();

    // store hash of original virus in a file.
    ofstream output;
    output.open("hash of original virus.txt");
```

Muhammad Hammad (23K-2005)



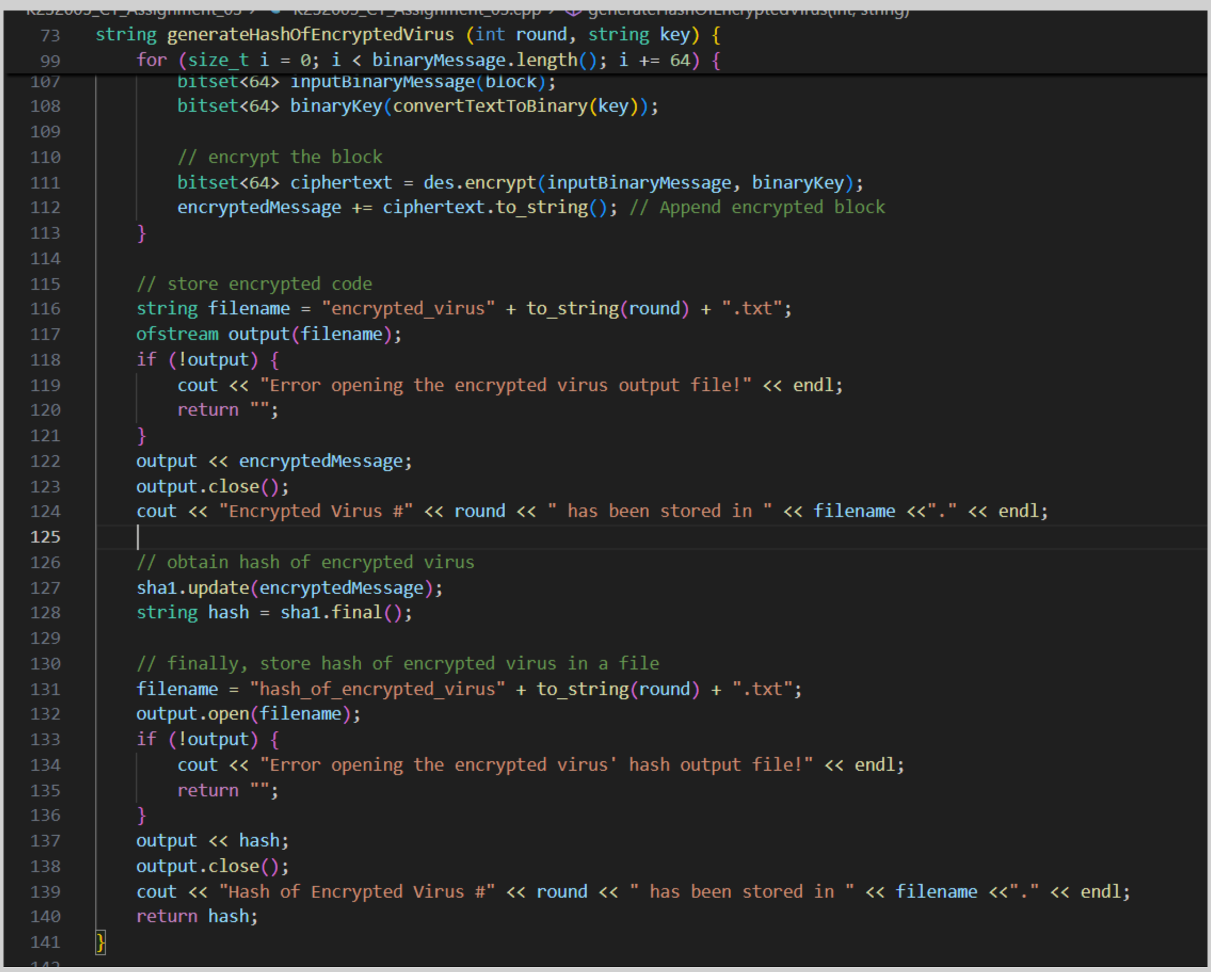
```cpp
string generateOriginalVirusHash () {
    // store hash of original virus in a file.
    ofstream output;
    output.open("hash_of_original_virus.txt");

    if (!output) {
        cout << "error opening the output file!" << endl;
        return "";
    }

    output << hash;
    output.close();
    cout << "\nHash of original virus has been stored in 'hash_of_original_virus.txt'" << endl;
    return hash;
}

string inputKey(int round) {
    cout << "Enter key for encryption #" << round << ": ";
    string key;
    cin >> key;

    if (key.size() != 8) {
        cout << "The key must be 8 characters or 64 bits. Please enter again." << endl;
        inputKey(round);
    }

    return key;
}

string convertTextToBinary(const string &text) {
    string binaryText;
    for (char c : text) {
        bitset<8> binary(c);
        binaryText += binary.to_string();
    }
    return binaryText;
}
```

**Muhammad Hammad (23K-2005)**

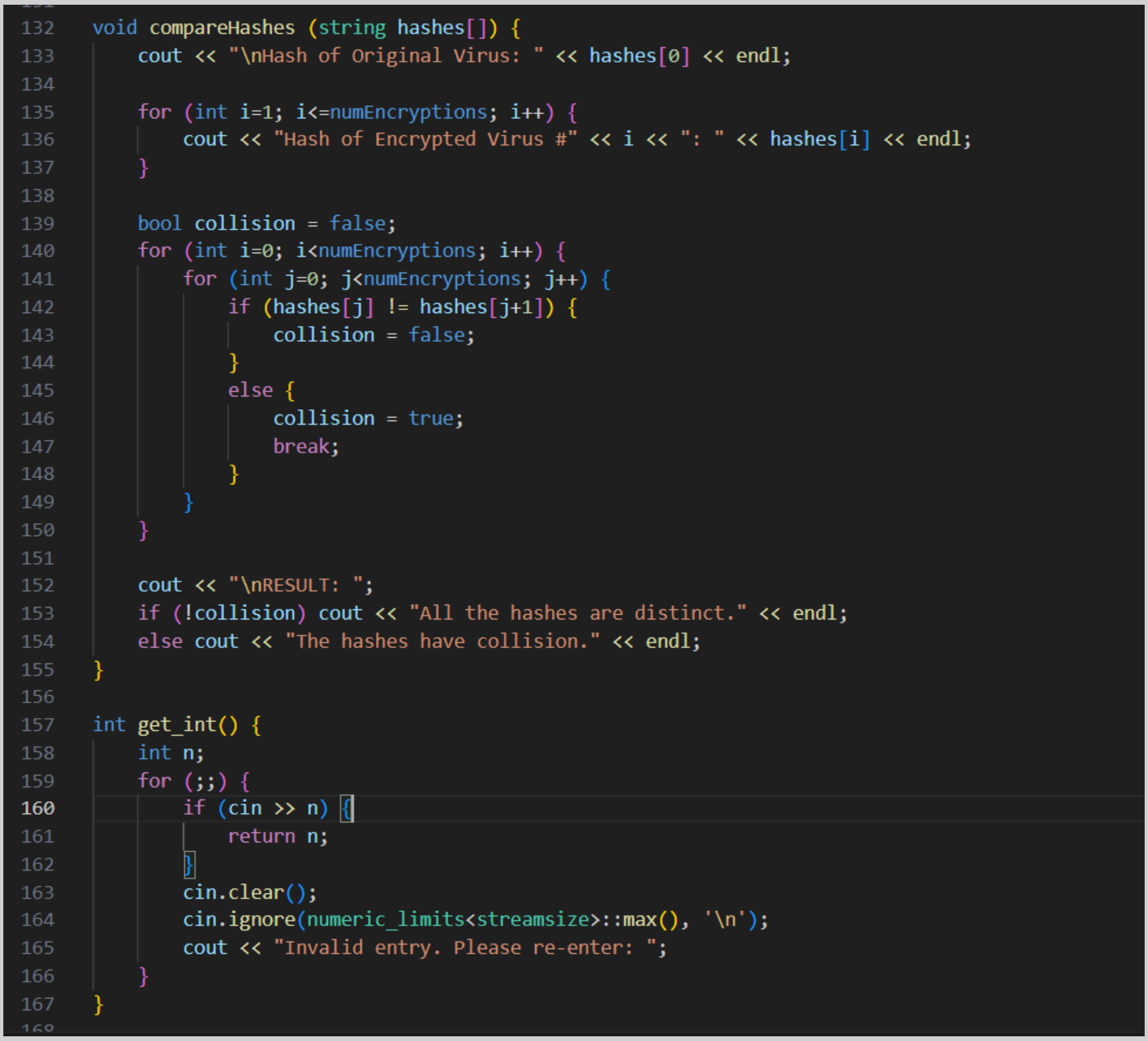


```cpp
string generateHashOfEncryptedVirus (int round, string key) {

    // fetch the virus from file and store it in a variable
    ifstream inputfromFile;
    inputfromFile.open("virus.txt");

    if (!inputfromFile) {
        cout << "error opening the input file!" << endl;
        return "";
    }

    string data;
    string message = "";

    while (!inputfromFile.eof()) {
        getline(inputfromFile, data);
        message += data;
    }

    // convert the virus into binary
    string binaryMessage = convertTextToBinary(message);
    DES des;
    SHA1 sha1;
    string encryptedMessage;

    // encrypt the message in 64 bits blocks
    for (size_t i = 0; i < binaryMessage.length(); i += 64) {
        string block = binaryMessage.substr(i, i+64);

        // fixed by gpt: pad the block with zeros if it’s less than 64 bits
        while (block.length() < 64) {
            block += '0'; // extend the block bits with 0 if less than 64
        }

        bitset<64> inputBinaryMessage(block);
        bitset<64> binaryKey(convertTextToBinary(key));
```

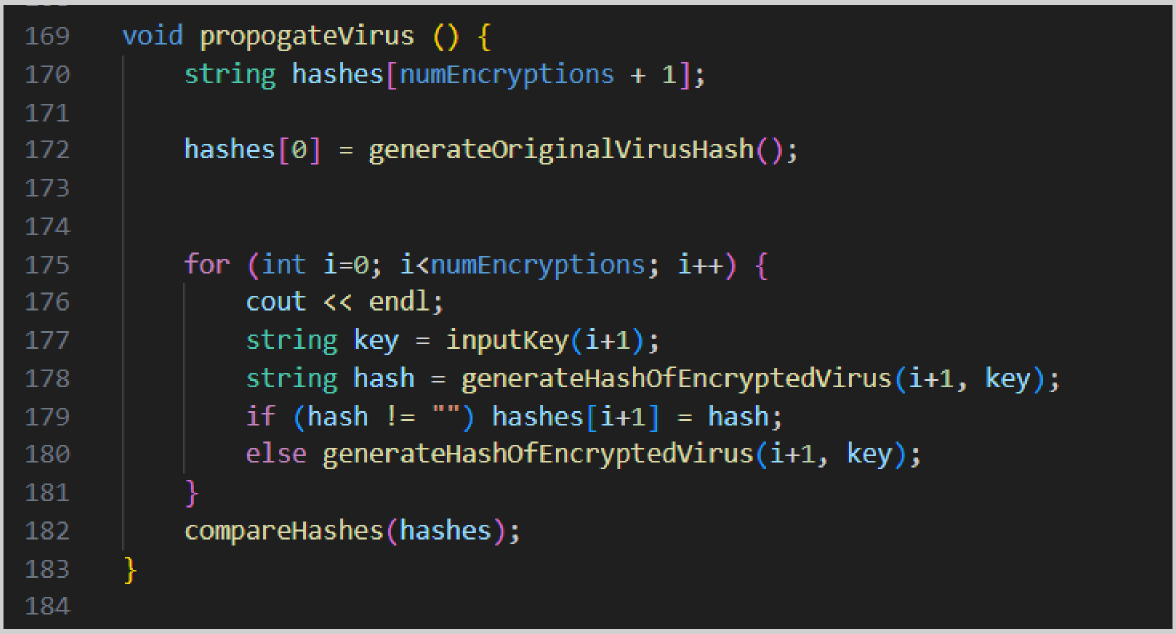
**Muhammad Hammad (23K-2005)**


```cpp
string generateHashOfEncryptedVirus (int round, string key) {
    for (size_t i = 0; i < binaryMessage.length(); i += 64) {
        bitset<64> inputBinaryMessage(block);
        bitset<64> binaryKey(convertTextToBinary(key));

        // encrypt the block
        bitset<64> ciphertext = des.encrypt(inputBinaryMessage, binaryKey);
        encryptedMessage += ciphertext.to_string(); // Append encrypted block
    }

    // store encrypted code
    string filename = "encrypted_virus" + to_string(round) + ".txt";
    ofstream output(filename);
    if (!output) {
        cout << "Error opening the encrypted virus output file!" << endl;
        return "";
    }
    output << encryptedMessage;
    output.close();
    cout << "Encrypted Virus #" << round << " has been stored in " << filename <<"." << endl;

    // obtain hash of encrypted virus
    sha1.update(encryptedMessage);
    string hash = sha1.final();

    // finally, store hash of encrypted virus in a file
    filename = "hash_of_encrypted_virus" + to_string(round) + ".txt";
    output.open(filename);
    if (!output) {
        cout << "Error opening the encrypted virus' hash output file!" << endl;
        return "";
    }
    output << hash;
    output.close();
    cout << "Hash of Encrypted Virus #" << round << " has been stored in " << filename <<"." << endl;
    return hash;
}
```

**Muhammad Hammad (23K-2005)**

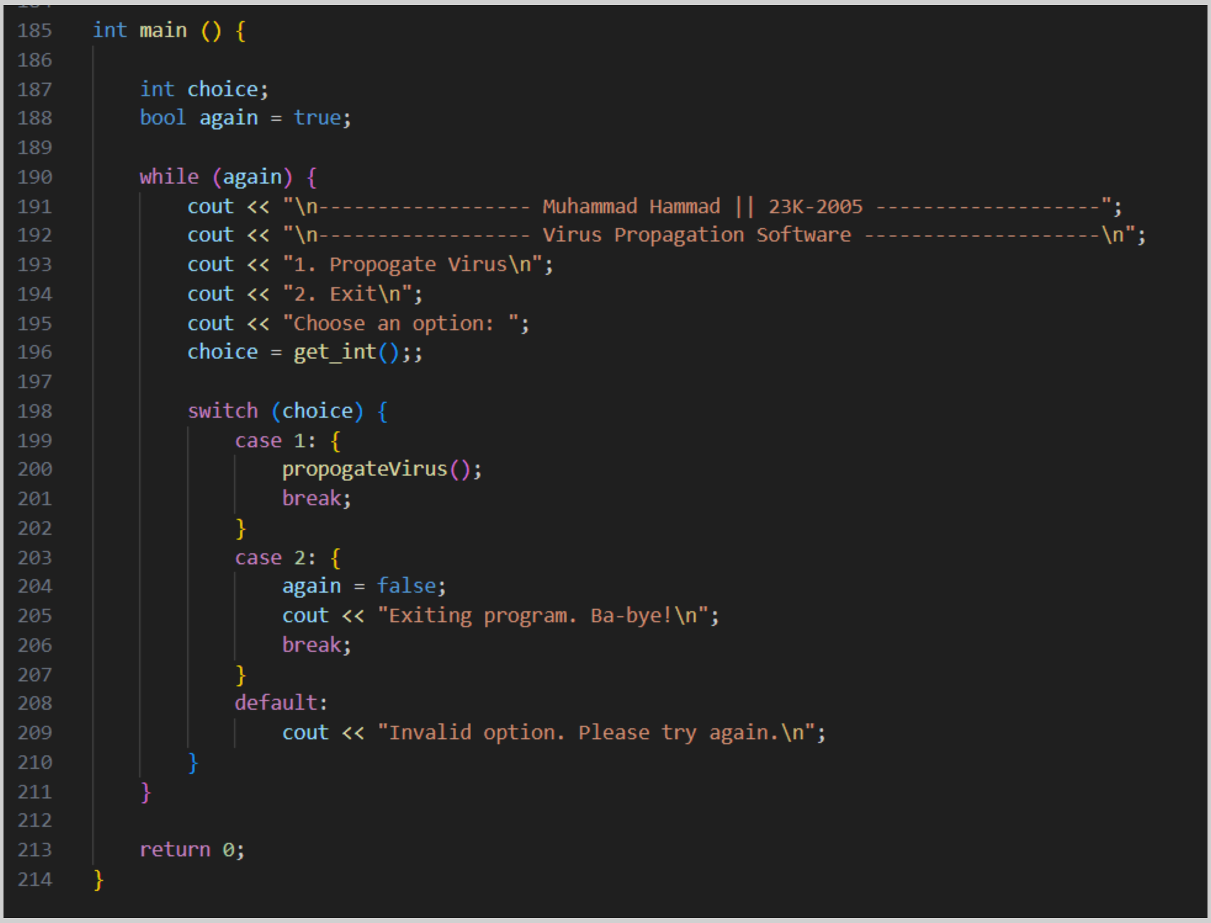


```cpp
void compareHashes (string hashes[]) {
    cout << "\nHash of Original Virus: " << hashes[0] << endl;

    for (int i=1; i<=numEncryptions; i++) {
        cout << "Hash of Encrypted Virus #" << i << ": " << hashes[i] << endl;
    }

    bool collision = false;
    for (int i=0; i<numEncryptions; i++) {
        for (int j=0; j<numEncryptions; j++) {
            if (hashes[j] != hashes[j+1]) {
                collision = false;
            }
            else {
                collision = true;
                break;
            }
        }
    }

    cout << "\nRESULT: ";
    if (!collision) cout << "All the hashes are distinct." << endl;
    else cout << "The hashes have collision." << endl;
}

int get_int() {
    int n;
    for (;;) {
        if (cin >> n) {
            return n;
        }
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Invalid entry. Please re-enter: ";
    }
}
```



```cpp
void propogateVirus () {
    string hashes[numEncryptions + 1];

    hashes[0] = generateOriginalVirusHash();


    for (int i=0; i<numEncryptions; i++) {
        cout << endl;
        string key = inputKey(i+1);
        string hash = generateHashOfEncryptedVirus(i+1, key);
        if (hash != "") hashes[i+1] = hash;
        else generateHashOfEncryptedVirus(i+1, key);
    }
    compareHashes(hashes);
}
```

Muhammad Hammad (23K-2005)

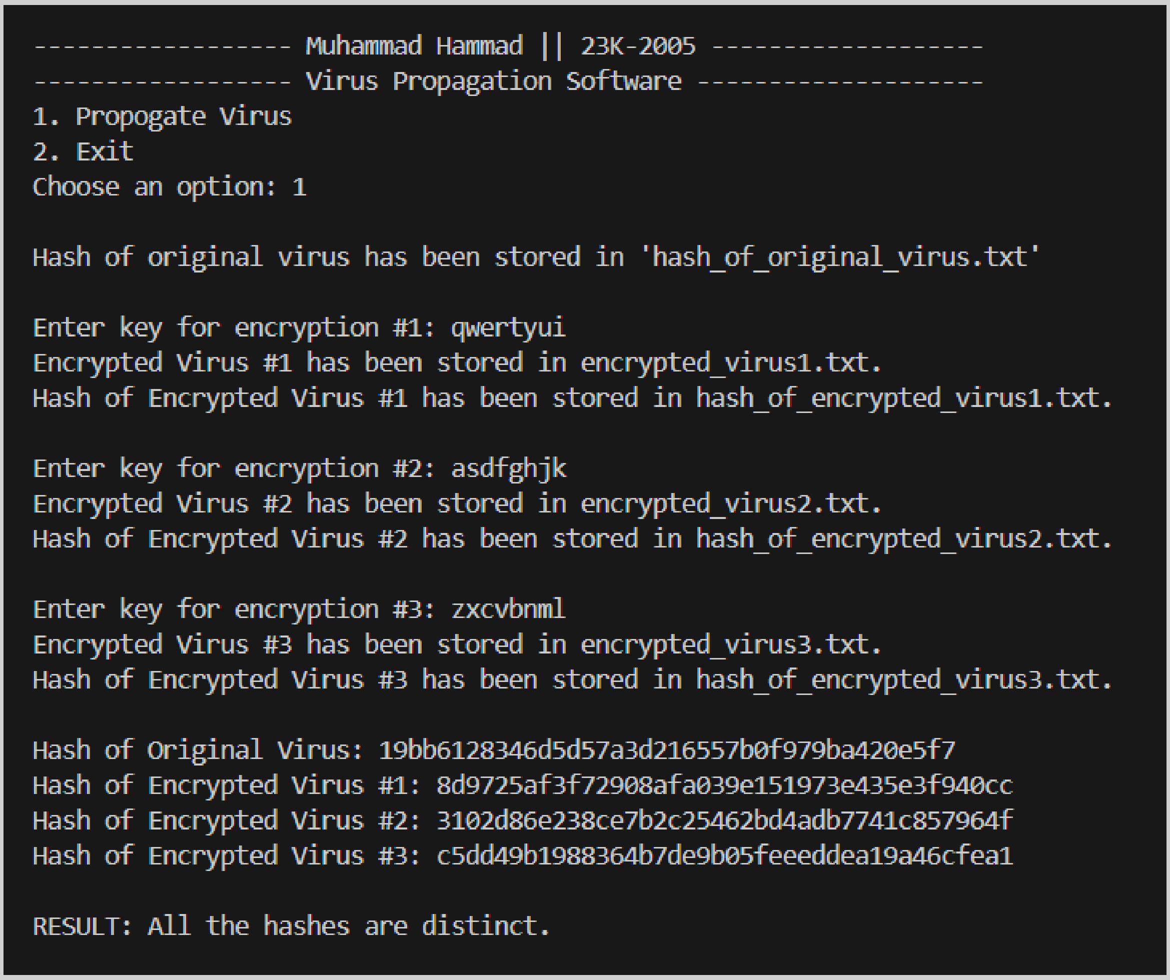


```cpp
int main () {

    int choice;
    bool again = true;

    while (again) {
        cout << "\n------------------ Muhammad Hammad || 23K-2005 -------------------";
        cout << "\n------------------ Virus Propagation Software --------------------\n";
        cout << "1. Propogate Virus\n";
        cout << "2. Exit\n";
        cout << "Choose an option: ";
        choice = get_int();;

        switch (choice) {
            case 1: {
                propogateVirus();
                break;
            }
            case 2: {
                again = false;
                cout << "Exiting program. Ba-bye!\n";
                break;
            }
            default:
                cout << "Invalid option. Please try again.\n";
        }
    }

    return 0;
}
```

Muhammad Hammad (23K-2005)

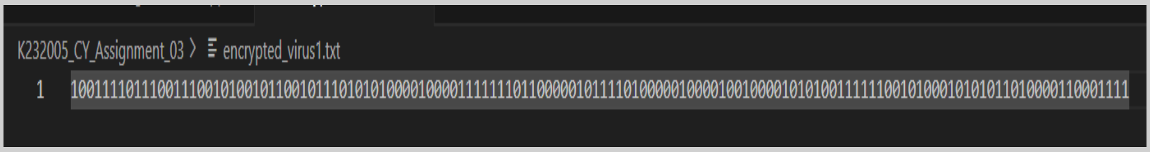
## Output

### Case 01: Distinct Hashes

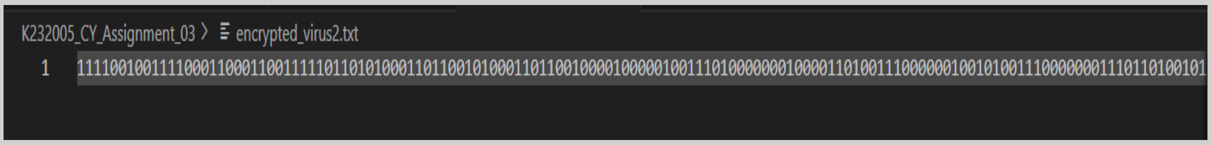


```
------------------ Muhammad Hammad || 23K-2005 --------------------
------------------ Virus Propagation Software --------------------
1. Propogate Virus
2. Exit
Choose an option: 1

Hash of original virus has been stored in 'hash_of_original_virus.txt'

Enter key for encryption #1: qwertyui
Encrypted Virus #1 has been stored in encrypted_virus1.txt.
Hash of Encrypted Virus #1 has been stored in hash_of_encrypted_virus1.txt.

Enter key for encryption #2: asdfghjk
Encrypted Virus #2 has been stored in encrypted_virus2.txt.
Hash of Encrypted Virus #2 has been stored in hash_of_encrypted_virus2.txt.

Enter key for encryption #3: zxcvbnml
Encrypted Virus #3 has been stored in encrypted_virus3.txt.
Hash of Encrypted Virus #3 has been stored in hash_of_encrypted_virus3.txt.

Hash of Original Virus: 19bb6128346d5d57a3d216557b0f979ba420e5f7
Hash of Encrypted Virus #1: 8d9725af3f72908afa039e151973e435e3f940cc
Hash of Encrypted Virus #2: 3102d86e238ce7b2c25462bd4adb7741c857964f
Hash of Encrypted Virus #3: c5dd49b1988364b7de9b05feeeddea19a46cfea1

RESULT: All the hashes are distinct.
```

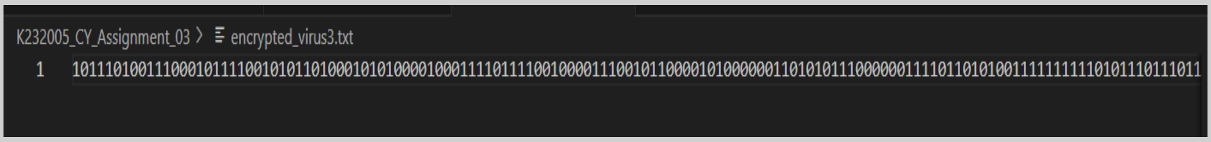
### Encrypted Virus Files



```
K232005_CY_Assignment_03 > encrypted_virus1.txt
1001111011100111001010010110010111010101000010000111111101100000101111010000010000100100001010100111111001010001010101101000011000 1111
```



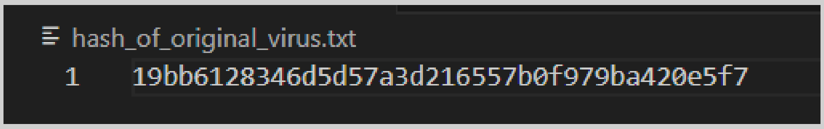
```
K232005_CY_Assignment_03 > encrypted_virus2.txt
111100100111100011000110011111011010100011011001010001101100100001000001001110100000000100001101001110000000100101001110000000111011010010 1
```



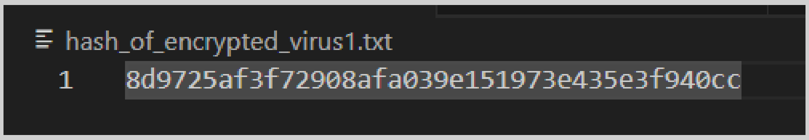
```
K232005_CY_Assignment_03 > encrypted_virus3.txt
101110100111000101111001010110100010101000010001111011110010000111001011000010100000001101010111000000111101101010011111111110101110111011
```

Muhammad Hammad (23K-2005)

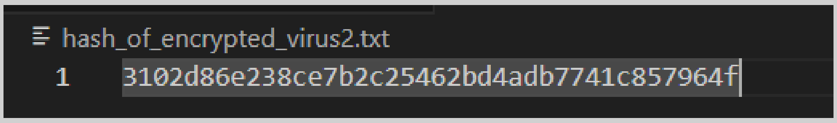
## Hashes of Encrypted Viruses



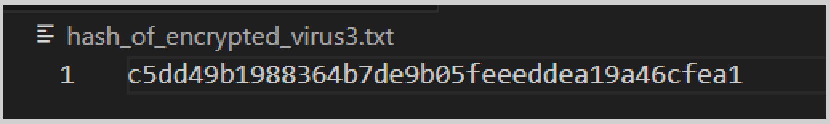
```
hash_of_original_virus.txt
19bb6128346d5d57a3d216557b0f979ba420e5f7
```



```
hash_of_encrypted_virus1.txt
8d9725af3f72908afa039e151973e435e3f940cc
```

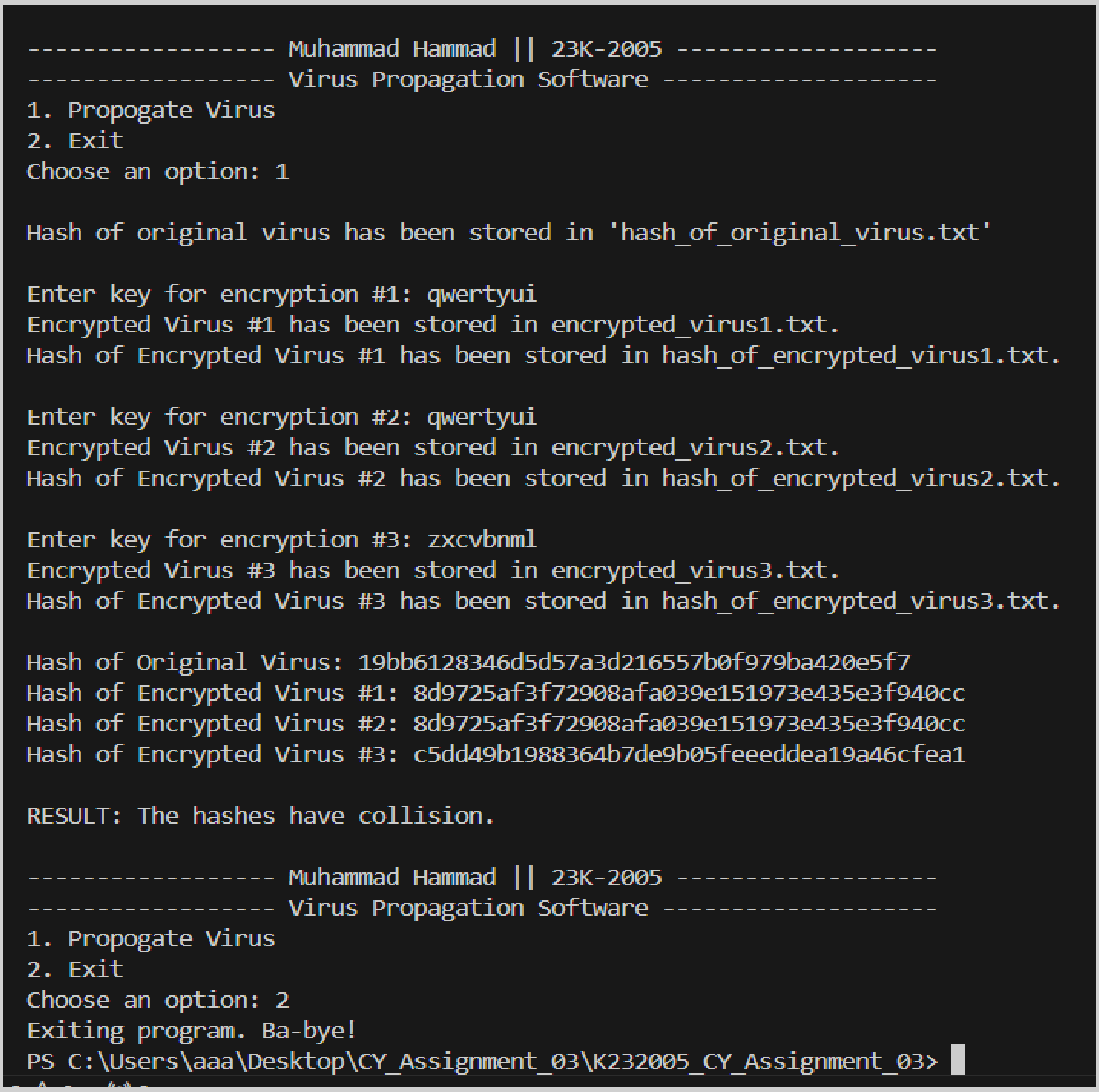


```
hash_of_encrypted_virus2.txt
3102d86e238ce7b2c25462bd4adb7741c857964f
```



```
hash_of_encrypted_virus3.txt
c5dd49b1988364b7de9b05feeeddea19a46cfea1
```

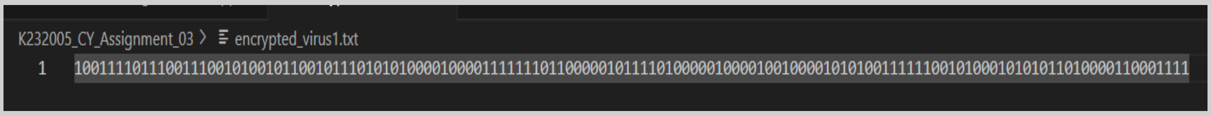
## Case 02: Same Hashes (Collision)



```
------------------ Muhammad Hammad || 23K-2005 -------------------
------------------ Virus Propagation Software --------------------
1. Propogate Virus
2. Exit
Choose an option: 1

Hash of original virus has been stored in 'hash_of_original_virus.txt'

Enter key for encryption #1: qwertyui
Encrypted Virus #1 has been stored in encrypted_virus1.txt.
Hash of Encrypted Virus #1 has been stored in hash_of_encrypted_virus1.txt.

Enter key for encryption #2: qwertyui
Encrypted Virus #2 has been stored in encrypted_virus2.txt.
Hash of Encrypted Virus #2 has been stored in hash_of_encrypted_virus2.txt.

Enter key for encryption #3: zxcvbnml
Encrypted Virus #3 has been stored in encrypted_virus3.txt.
Hash of Encrypted Virus #3 has been stored in hash_of_encrypted_virus3.txt.

Hash of Original Virus: 19bb6128346d5d57a3d216557b0f979ba420e5f7
Hash of Encrypted Virus #1: 8d9725af3f72908afa039e151973e435e3f940cc
Hash of Encrypted Virus #2: 8d9725af3f72908afa039e151973e435e3f940cc
Hash of Encrypted Virus #3: c5dd49b1988364b7de9b05feeeddea19a46cfea1

RESULT: The hashes have collision.

------------------ Muhammad Hammad || 23K-2005 -------------------
------------------ Virus Propagation Software --------------------
1. Propogate Virus
2. Exit
Choose an option: 2
Exiting program. Ba-bye!
PS C:\Users\aaa\Desktop\CY_Assignment_03\K232005_CY_Assignment_03>
```

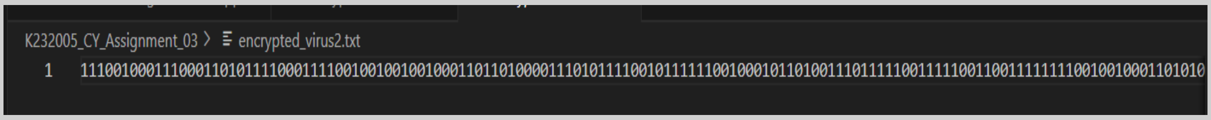
Muhammad Hammad (23K-2005)

## Encrypted Virus Files



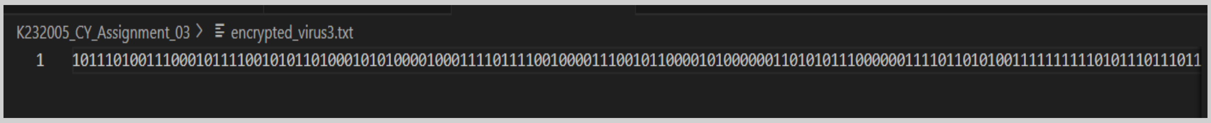
K232005_CY_Assignment_03 > encrypted_virus1.txt

```
1001111011100111001010010110010111010101000010000111111101100000101111010000010000100100001010100111111001010001010101101000011000 1111
```



K232005_CY_Assignment_03 > encrypted_virus2.txt

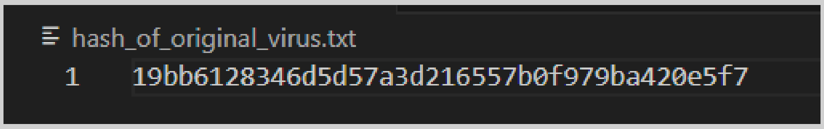
```
11100100011100011010111100011110010010010010001101101000011101011110010111111001000101101001110111110011111001100111111110010010001101010
```



K232005_CY_Assignment_03 > encrypted_virus3.txt

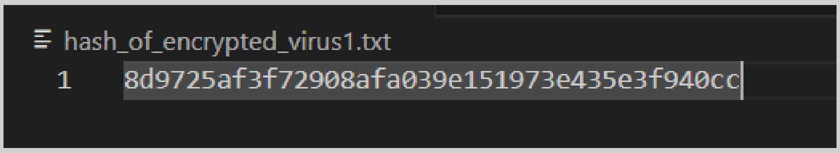
```
10111010011100010111100101011010001010100001000111101111001000011100101100001010000001101010111000000111101101010011111111110101110111011
```

## Hashes of Encrypted Viruses



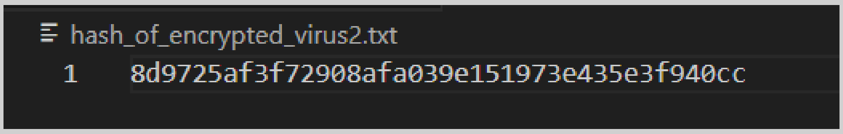
hash_of_original_virus.txt

```
19bb6128346d5d57a3d216557b0f979ba420e5f7
```



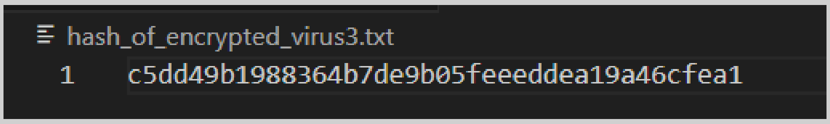
hash_of_encrypted_virus1.txt

```
8d9725af3f72908afa039e151973e435e3f940cc
```



hash_of_encrypted_virus2.txt

```
8d9725af3f72908afa039e151973e435e3f940cc
```



hash_of_encrypted_virus3.txt

```
c5dd49b1988364b7de9b05feeeddea19a46cfea1
```