CRYPTOGRAPHY

ENCRYPTION AND DECRYPTION

CONTENTS

INTRODUCTION:

- In today's modern world confidential and classified documents are difficult to be kept secret especially in computers.
- The passwords, our bank account details, and other confidential information requires to be kept secret form the external invaders.
- Here **CRYPTOGRAPHY** plays an important role.
- **CRYPTOGRAPHY:**
  - ✓ It is the art and science of creating secret writing via making them unreadable by scrambling the Symbols of the message in such a way that only the intended receiver can read it by knowing the method used to scramble the message.
  - ✓ It can be descrambled, if they know some secret (like a password) between them.

APPLICATIONS OF CRYPTOGRAPHY: -

- Cryptography makes secure web sites and electronic safe transmissions possible.
- Due to the large number of commercial transactions on the internet, cryptography is very vital key in ensuring the security of the transactions.
- Cryptography allows you to have confidence in your electronic transactions.
- Encryption is used in electronic transactions to protect data such as account numbers and transaction amounts, digital signatures replace handwritten signatures or credit card authorizations, and public-key encryption provides confidentiality.

SYSTEM REQUIREMENT:

HARDWARE:

1. Processor: Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz, 1801 Mhz, 4 Core(s), 8 Logical Processor(s).

SOFTWARE:

1. DEV C++ was used for coding and compiling.
2. OS Name: Microsoft Windows 10 Pro.

HEADER FILES: -

S.no	HEADER FILE	PURPOSE
1	#include<iostream>	Input and output commands
2	#include<fstream>	File handling
3	#include<string.h>	Character variables
4	#include<conio.h>	Clearing the screen

FUNCTIONS: -

S.no	FUCNTIONS	PURPOSE
1	int getFileSize(char*);	Read the size of file
2	char* caesar(char*);	Caesar Cipher encryption
3	char* transposition(char*);	Transposition encryption and Decryption
4	char* dcrcaesar(char*);	Caesar Cipher decryption
5	char* getPassword(char*);	To read password
6	char*maskPassword(char*);	To hide the password with *

NOTE:

- Here only two types of encryption and decryption algorithm is followed: -
  1) Transposition
  2) Caesar Cipher

PROJECT DESCRIPTION:

The project requirements are as follows:

PROGRAM SPECIFICATION:

1) Using the techniques presented during this semester to create a complete C++ program to emulate an Encryption/Decryption Machine.

The program will be capable of the following:

- Encrypt a file provided by the user
- Choose between two different encryption methods
- Decrypt a file by the user
- Choose between two different decryptions methods
- Decrypt without knowing the encryption method

2) The interface must be professional and fully intuitive to the user

3) The program will be menu driven.

4) The program will use a function(s) to define and implement each of the methods and store the original file and the encrypted/decrypted file.

5) In addition to using a function we have:

- Selection statements (if, if-else, switch)
- Loops (while, for, do-while)
- Standard Libraries (don't recreate the wheel)
- Functions
- Arrays

6) The two encryption/decryption methods are:

1. Transposition
2. Caesar Cipher

❼ TRANSPOSITION:

In this type of encryption, the sentence or data is encrypted in a laterally inverted way.

The positions of the letters get swapped from last to first position.

EX: ABCDE would be encrypted as EDCBA.

When a huge amount of data is encrypted in this way, it would be difficult to decode it without decrypting it.

⑦ CAESAR-CIPHER:

Here ASCII values are taken into consideration. The characters are replaced by 25 (as KEY in code is set to 25) characters after the entered character.

Briefly, the letters in the file are added 25 numbers ahead.

EX: If letter A is entered then Z is displayed.

ABCDE is encrypted as Z[\]^.

FLOW OF PROGRAM:

1. The name of the file to be encrypted is entered.
2. If file does not exist, the program terminates.
3. When file exists, the file is encrypted by one of the encryption methods.
4. The file can then be decrypted again in the program.
5. During execution of code, the original file is accessed.
6. When the file is encrypted, a <file_name>. enc type is created which consists encrypted data.
7. A file which is encrypted is decrypted only using a password.
8. A mismatch in password would terminate the code.

IMPORTANT POINTS:

- There are two ways you can encrypt a file.
- The password set for decrypting the file secure unauthorized access to file.
- The Caesar-Cipher encryption uses all sorts of characters, numbers and special characters.
- The output screen is set to clear after each execution in program so that output is clearer to comprehend.
- The encrypted file is also saved to the device

SOURCE CODE:

```
 /************************** HEADER FILES *****************************/
#include<iostream>
#include<fstream>
#include<string.h>
#include<conio.h>
#define KEY 25                                      // Macro to store KEY for Encryption.
using namespace std;

int getFileSize(char*);
char* caesar(char*);
char* transposition(char*);
char* dcrcaesar(char*);
char* getPassword(char* );
char* maskPassword(char*);
/*******************************************************************/



/*****************************MAIN CODE*****************************/
int main()
{
        char fname[80];
        char encfile[80];
        char pw[80], choice = 'Y';
        char* encstring;
        char* dcrstring;
        int srcsize;
        int ch,ch1;
        ifstream src,drc;
        ofstream enc;
        cout<<"\n Enter File name : ";
        cin.getline(fname,80);


        do
        {
            system("CLS");
            strcpy(encfile,fname);
            strcat(encfile,".enc");
            src.open(fname);
```

```
/***************************MAIN MENU****************************/
            if(!src.good())
            {
                cout<<"\n File does not exists";
            }
                else
                {
                        srcsize = getFileSize(fname);
                        src.seekg(0,ios::beg);
                        char* str = new char[srcsize];
                        char* encstring = new char[srcsize];
                        char* dcrstring = new char[srcsize];
                        src.read((char*)str,srcsize);
                        cout<<"\n 1. Encryption\n 2. Decryption\n 3. Exit";
                        cout<<"\n Enter your choice : ";
                        cin>>ch;
                        char encmode;
/****************************************************************/

/*************************** ENCRYPTION ****************************/
                        switch(ch)
                        {
                         case 1:
                                cout<<"\n Enter password : ";
                                cin.ignore();
                                maskPassword(pw);
                                cout<<"\n\n 1. Transposition\n 2. Caesar Cipher";
                                cout<<"\n\n Enter your choice of encryption method : ";
                                cin>>ch1;

        /*************** ENCRYPTION METHODS ***************/
                                switch(ch1)                    //INNER SWITCH
                                {

                                case 1:
    /************** FUNCTIONS CALLED FOR ENCRYPTION ***************/

                                        encstring = transposition(str); //TRANSPOSITION
                                        encmode = 'T';
                                        break;
                                case 2:
                                        encstring = caesar(str);          //CAESAR CIPHER
                                        encmode = 'C';
                                        break;

                                default : cout<<"\n Enter proper choice (1/2)";
                                exit(1);
                                }
        /************************************************/
```

*/******************** FUNCTIONS CALLED FOR ENCRYPTION ******************/*

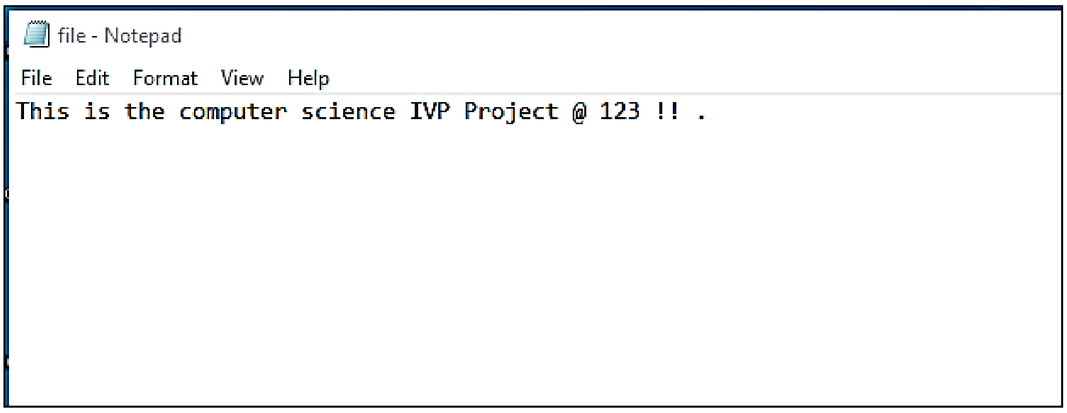
```
                    enc.open(encfile);
                    enc.seekp(0,ios::beg);
                    enc.write((char*)&encmode,1);
                    enc.write((char*)"`",1);
                    enc.write((char*)&pw,strlen(pw));
                    enc.write((char*)"`",1);
                    enc.write((char*)encstring,srcsize);
                    enc.flush();

                cout<<"\n The encrypted version of your file is :\n\n "<<encstring;
                cout<<"\n\n "<<fname<<" encrypted successfully as "<<encfile;
                break;
/****************************************************************/

    /******************DECRYPTION METHODS******************/

            case 2:
                cout<<"\n Enter password : ";
                cin.ignore();
                maskPassword(pw);
                cout<<"\n Enter file name to decrypt : ";
                gets(encfile);
                drc.open(encfile);

                  if(!drc.good())
                  {
                    cout<<"\n File does not exists";
                   }
                    else
                    {
                        srcsize = getFileSize(encfile);
                        drc.seekg(0,ios::beg);
                        char* str = new char[srcsize];
                        char* encstring = new char[srcsize];
                        char* dcrstring = new char[srcsize];
                        drc.read((char*)str,srcsize);
                        if(!strcmp(pw,getPassword(str)))
                        {
                          int i,j;
                          for(i=(strlen(pw)+3),j=0; str[i]!='\0'; i++,j++)
                         {
                                encstring[j] = str[i];
                         }
                                encstring[j] = '\0';
                    }
                else
                {
                        cout<<"\n Wrong Password.";
                        exit(1);
                    }
/******************** FUNCTIONS CALLED FOR DECRYPTION ******************/

                    if(str[0] == 'T')
                    {

                cout<<"\n The decrypted text is : "<<"\n"<<transposition(encstring)<<endl;
                                //DECRYPTION FOR TRANSPOSITION
                     }
                        else if(str[0] == 'C')
                        {

                cout<<"\n The decrypted text is :"<<"\n"<<dcrcaesar(encstring)<<endl;
                                //DECRYPTION FOR CAESAR CIPHER
                        }
                }

                cout<<"\n\n "<<encfile<<" has been decrypted successfully";
                drc.close();
                break;
/******************************************************************/

                case 3 :
                    cout<<" \n **** PROGRAM TERMINATED!! ****";
                    exit(1);
                    default :       cout<<"\n Enter correct choice(1-3)";
                                    exit(1);
                }
            }
                cout<<"\n\n Do you want to continue (y/n) : ";
                cin>>choice;
                src.close();
                cin.ignore();
            }while(choice == 'y' || choice == 'Y');
            return 0;
}

/******************************************************************/
```

```
/**********************CASEAR-CIPHER ENCRYPTION***********************/
char* caesar(char* str)
{
	int i,encsize;
	encsize = strlen(str);
	char* enstr = new char[encsize];
	for(i=0;i<encsize;i++)
	{
		enstr[i] = (str[i] + KEY);
	}
	enstr[i] = '\0';
	return enstr;
}
/*****************************************************************/

/**************TRANSPOSITION ENCRYPTION AND DECRYPTION*****************/
char* transposition(char* str)
{
	int i,j,encsize;
	char temp;
	encsize = strlen(str);
	char* enstr1 = new char[encsize];
	enstr1 = str;
	for(i=0,j=encsize-1;i<j;i++,j--)
	{
		temp = enstr1[i];
		enstr1[i] = enstr1[j];
		enstr1[j] = temp;
	}
	return enstr1;
}
/*****************************************************************/

/**********************CASEAR-CIPHER       DECRYPTION**********************/
char* dcrcaesar(char* encstring)
{
	int var,i,dcrsize;
	dcrsize = strlen(encstring);
	char* dcrstr = new char[dcrsize];
	for(i=0;i<dcrsize;i++)
	{
		dcrstr[i] = (encstring[i] - KEY);
	}
	dcrstr[i] = '\0';
	return dcrstr;
}
/*****************************************************************/
```

```
/***************************SIZE OF SOUCE FILE**************************/
int getFileSize(char* fname) // To get size of source file.
{
        int filesize;
        ifstream src(fname);
        src.seekg(0,ios::end);
        filesize = src.tellg();
        return filesize;
}
/*****************************************************************/


    /********************** GET PASSWORD**********************/
            char* getPassword(char* str)
            {
                    int i,j;
                    char* pw = new char[80];
                    for(i=2,j=0;str[i]!='`';i++,j++)
                    {
                            pw[j] = str[i];
                    }
                    pw[j] = '\0';
                    return pw;
            }
     /***************************************************/


    /******************HIDE PASSWORD********************/
            char* maskPassword(char* pw) // To hide password.
            {
                    int i=0;
                    char ch;

                    do
                    {
                            ch = getch();
                            if(ch == 13)
                            break;
                            cout<<"*";
                            pw[i] = ch;
                            i++;
                    }while(ch != 13);
                    return pw;
            }
    /***************************************************/


/***************************END OF CODE****************************/
```

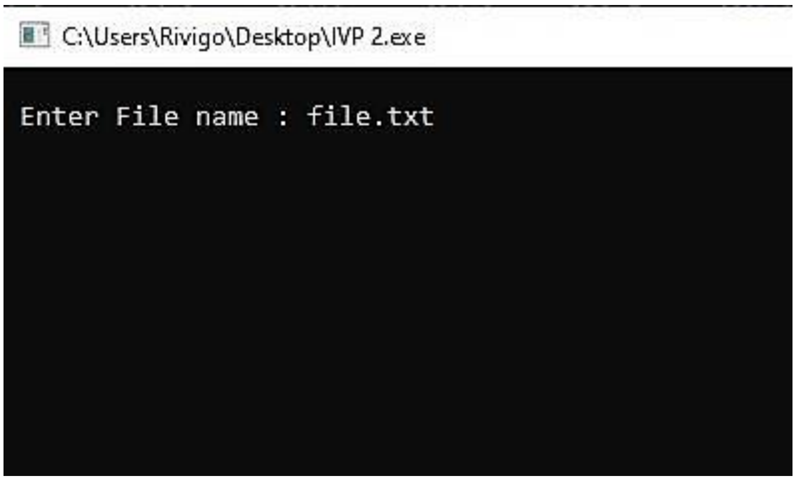
OUTPUTS:

1) FILE: A File created for encryption.



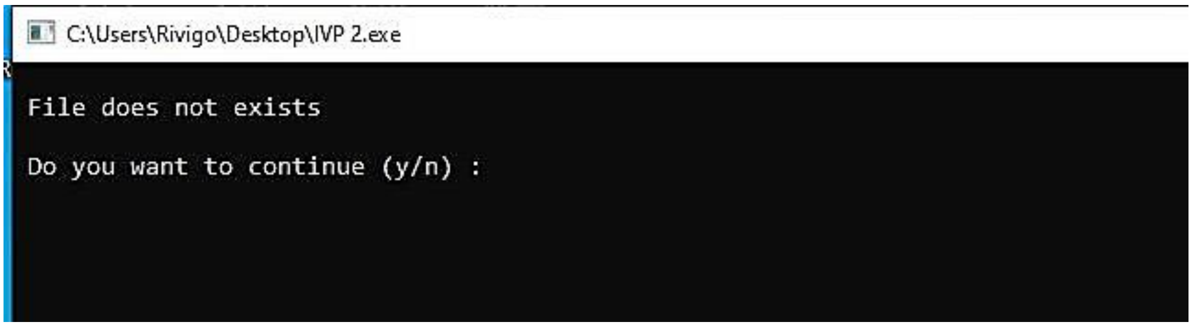
```
This is the computer science IVP Project @ 123 !! .
```

2) File name is entered



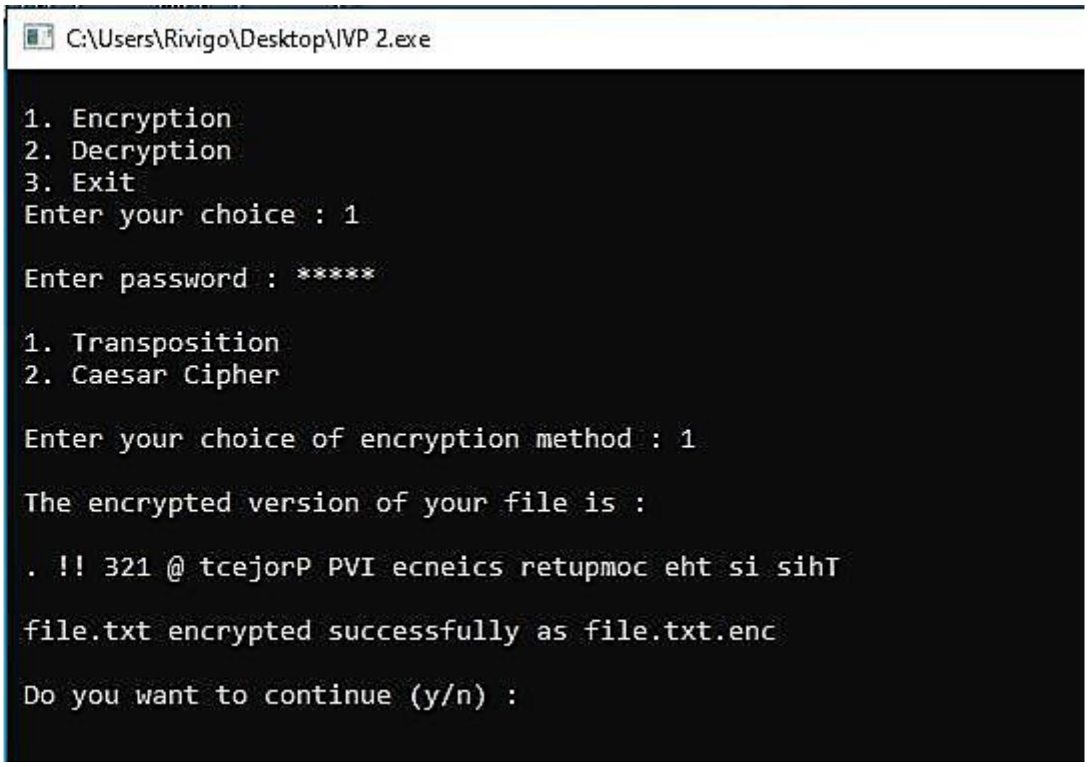
```
Enter File name : file.txt
```

3) If file is not created



```
File does not exists

Do you want to continue (y/n) :
```

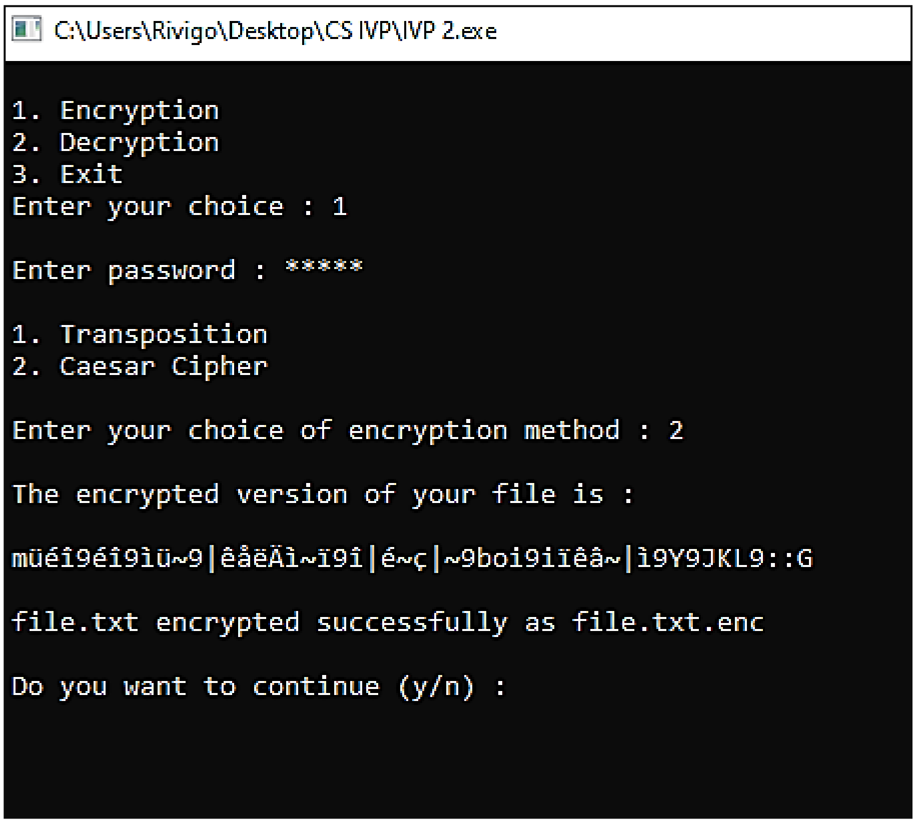
4) File encrypted through Transposition method:



C:\Users\Rivigo\Desktop\IVP 2.exe

```
1. Encryption
2. Decryption
3. Exit
Enter your choice : 1

Enter password : *****

1. Transposition
2. Caesar Cipher

Enter your choice of encryption method : 1

The encrypted version of your file is :

. !! 321 @ tcejorP PVI ecneics retupmoc eht si sihT

file.txt encrypted successfully as file.txt.enc

Do you want to continue (y/n) :
```

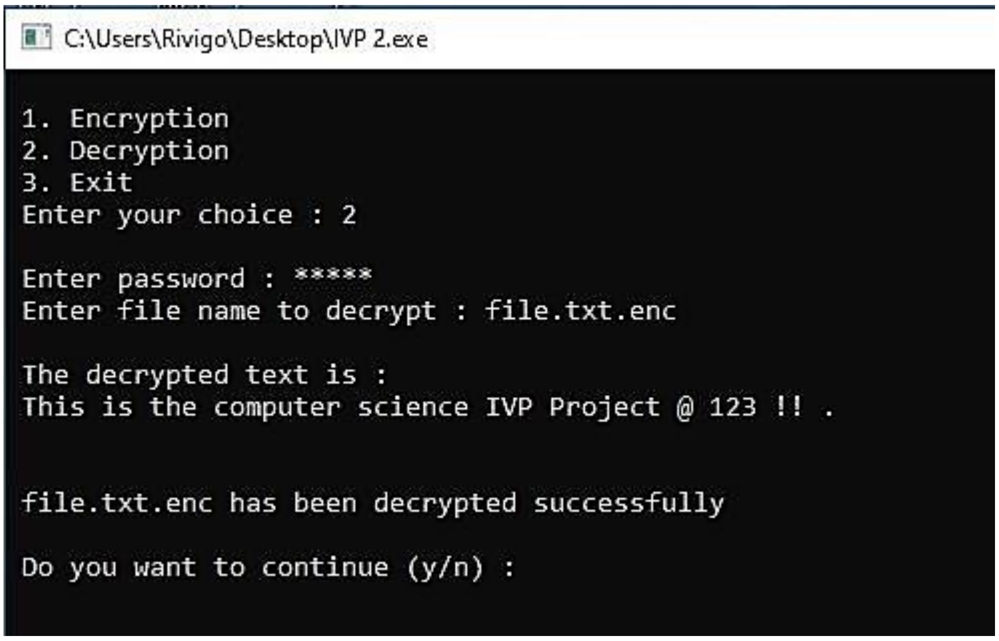
5) File encrypted through Caesar Cipher method:



C:\Users\Rivigo\Desktop\CS IVP\IVP 2.exe

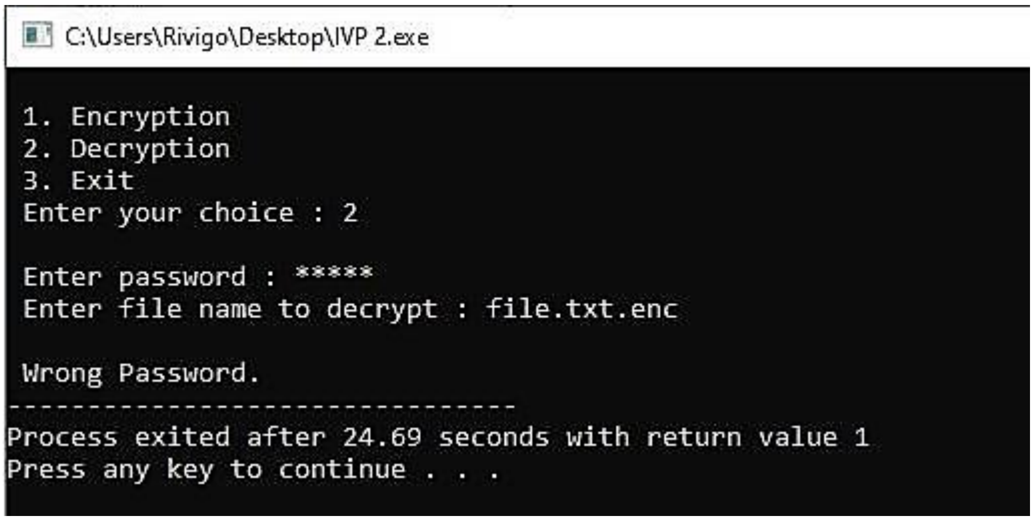
```
1. Encryption
2. Decryption
3. Exit
Enter your choice : 1

Enter password : *****

1. Transposition
2. Caesar Cipher

Enter your choice of encryption method : 2

The encrypted version of your file is :

müéî9éî9ìü~9|êåëÄì~ï9î|é~ç|~9boi9iïêâ~|ì9Y9JKL9::G

file.txt encrypted successfully as file.txt.enc

Do you want to continue (y/n) :
```

6) Decryption of the file:



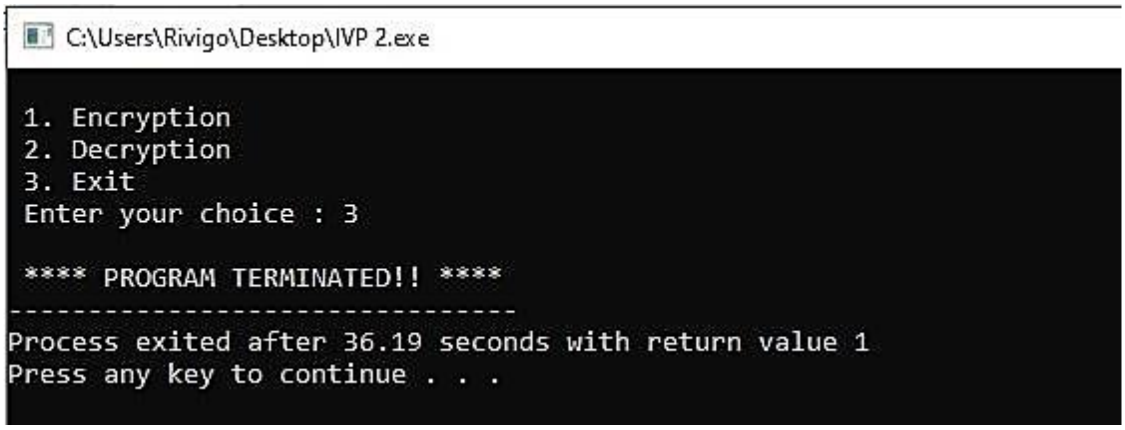
```
C:\Users\Rivigo\Desktop\IVP 2.exe

1. Encryption
2. Decryption
3. Exit
Enter your choice : 2

Enter password : *****
Enter file name to decrypt : file.txt.enc

The decrypted text is :
This is the computer science IVP Project @ 123 !! .


file.txt.enc has been decrypted successfully

Do you want to continue (y/n) :
```

7) Wrong password



```
C:\Users\Rivigo\Desktop\IVP 2.exe

 1. Encryption
 2. Decryption
 3. Exit
 Enter your choice : 2

 Enter password : *****
 Enter file name to decrypt : file.txt.enc

 Wrong Password.
--------------------------------
Process exited after 24.69 seconds with return value 1
Press any key to continue . . .
```

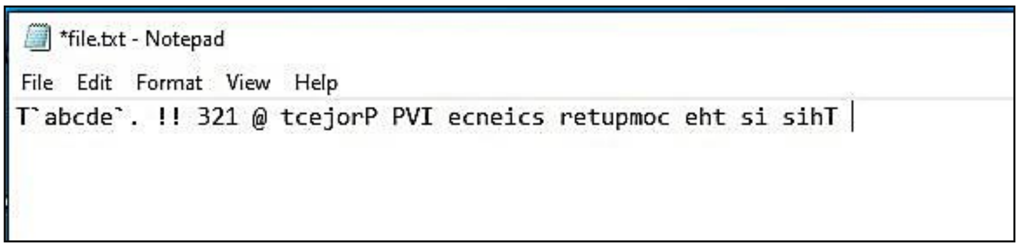
8) EXIT



```
C:\Users\Rivigo\Desktop\IVP 2.exe

 1. Encryption
 2. Decryption
 3. Exit
 Enter your choice : 3

 **** PROGRAM TERMINATED!! ****
--------------------------------
Process exited after 36.19 seconds with return value 1
Press any key to continue . . .
```

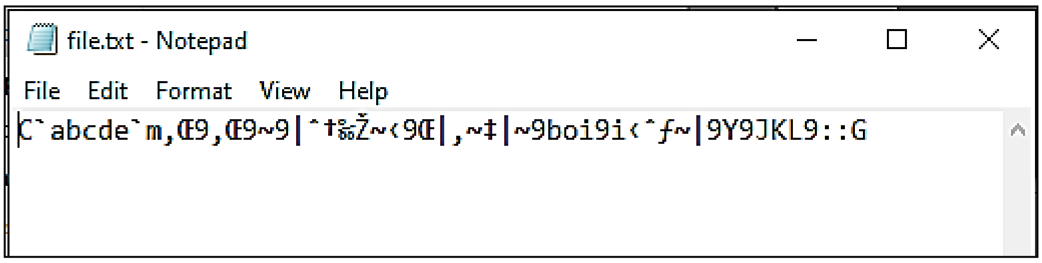
9) File after encryption through:

i) Transposition



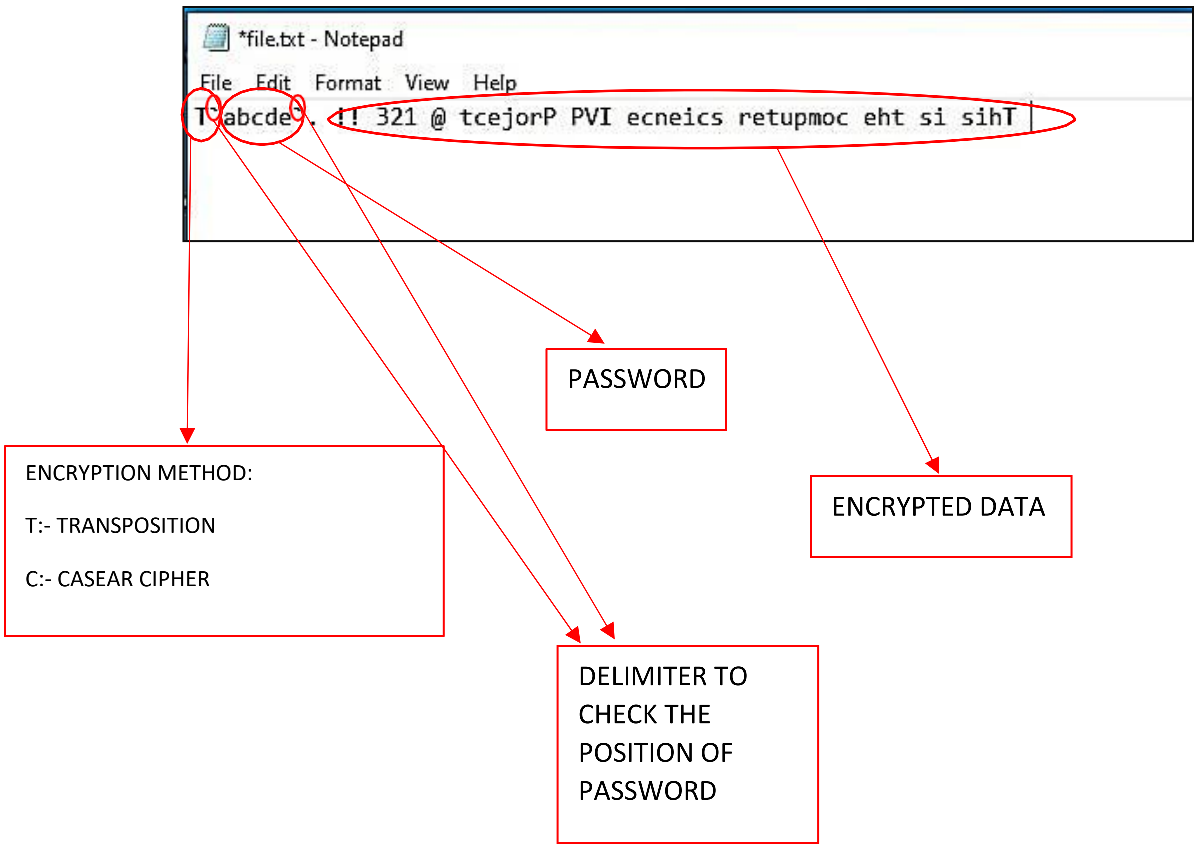
```
T`abcde`. !! 321 @ tcejorP PVI ecneics retupmoc eht si sihT
```

ii) Caesar Cipher



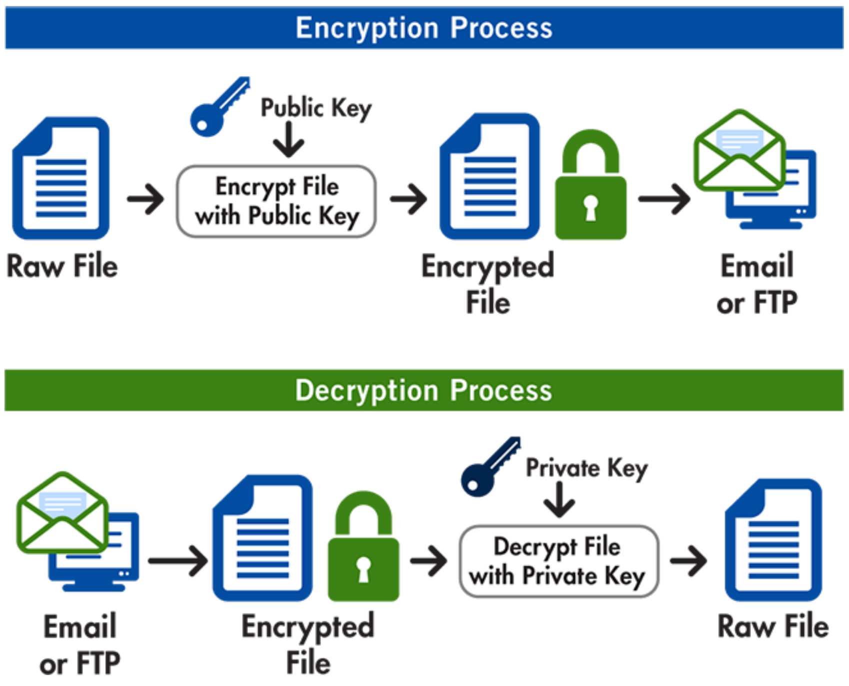
```
C`abcde`m,Œ9,Œ9~9|ˆ†‰Ž~‹9Œ|,~‡|~9boi9i‹ˆƒ~|9Y9JKL9::G
```

IN THESE FILES: -



```
T`abcde`. !! 321 @ tcejorP PVI ecneics retupmoc eht si sihT
```

WORKING OF CODE:



BIBLIOGRAPHY:

www.google.com
www.quora.com
en.wikipedia.org
books.gigatux.nl
www.networksorcery.com
www.cplusplus.com
www.c++reference.com