SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CRYPTOGRAPHY AND NETWORK SECURITY LABORATORY (7RCSL01)

Student Name: Pragati Shankar	USN: 1SI19CS090	Batch No: B2	Date: 06-12-2022

Evaluation:

Write Up (10 marks)	Clarity in concepts (10 marks)	Implementation and execution of the algorithms (10 marks)	Viva (05 marks)	Total (35 marks)

Sl.No	Name of the Faculty In-Charge	Signature
1.	H K Vedamurthy	
2.	Gururaj S P	

Question No: 1

Perform encryption and decryption using mono-alphabetic cipher. The program should support the following:

i. Construct an input file named plaintext.txt (consisting of 1000 alphabets, without any space or special characters)

ii. Compute key space (Permutation of set of all letters appeared in plaintext.txt: there are n! permutations of a set of n elements)

iii. Encrypt the characters of plaintext.txt using any one key from (ii) and store the corresponding ciphertext characters in ciphertext.txt

iv. Compute the frequency of occurrence of each alphabet in both plaintext.txt and ciphertext.txt and tabulate the results as follows

Frequency	Plaintext character	Ciphertext character
12.34	A	X
.	.	.
.	.	.

Monoalphabetic substitution cipher:

Select a Key randomly from 26! Key space and map from plain alphabet to cipher alphabet:

- Consider Plaintext P which contains sequence of characters.
- Consider the alphabet { a,b,c……..z}
- Consider a Initial Key which also contains only alphabets K= {a,b, …… z} to have the keyspace.
- Hence there will be 26! Keyspace.
- User has to generate randomly any one possible permutation of alphabets {a…z} say key K .
- Define each letter in the randomly generated key K against each letter of the plain text.
- Map from plain alphabet to cipher alphabet

CODE:

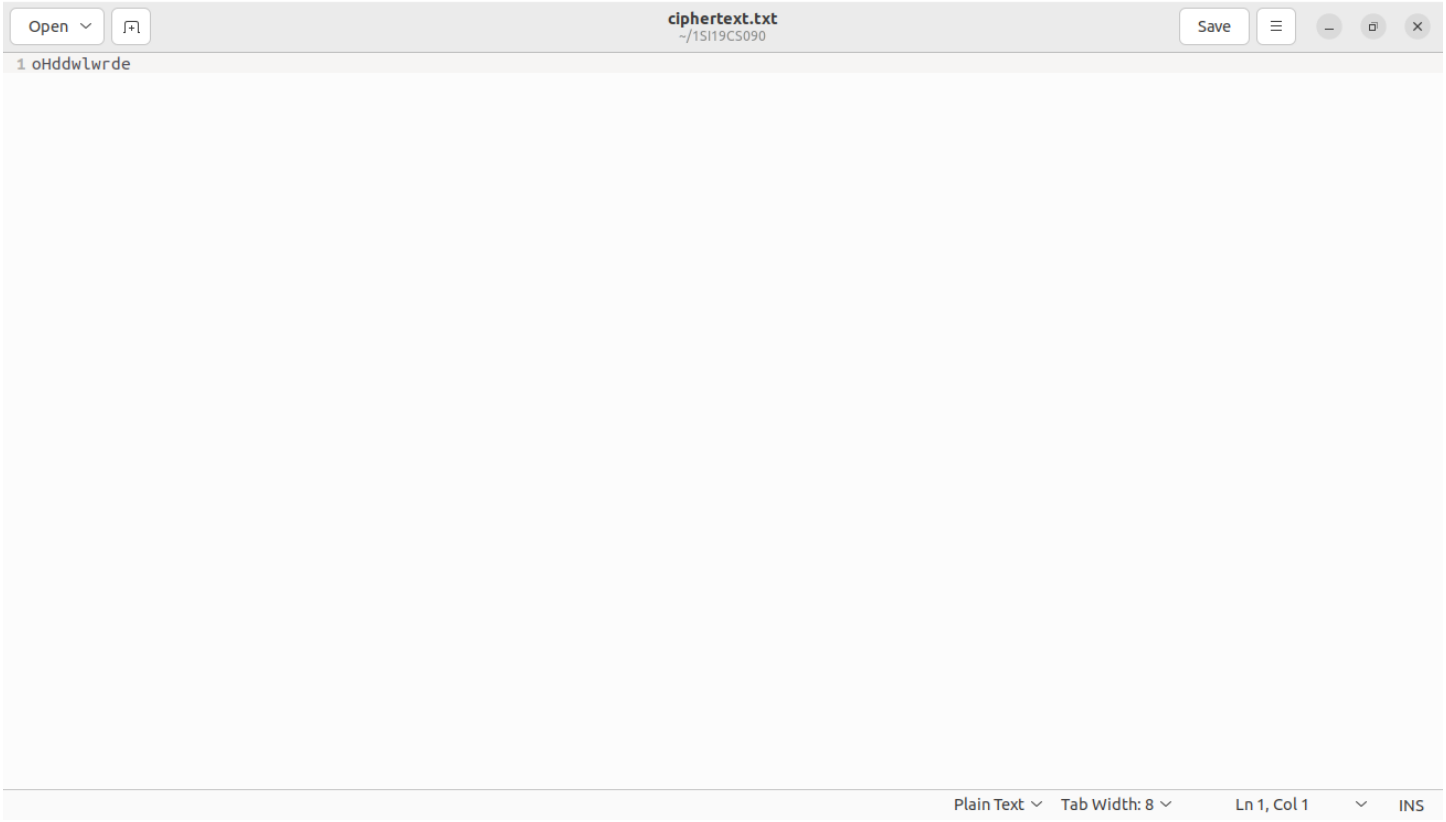
```
#include<bits/stdc++.h>
using namespace std;
char uniqtext[26];
string readPlainText() {
	ifstream fin;
	string ptext;
	fin.open("plaintext.txt");
	fin >> ptext;
	fin.close();
	return ptext;
}
void writecipherText(string ctext) {
	ofstream fout;
	fout.open("ciphertext.txt");
	fout << ctext;
	fout.close();
}
void permute(char a[], int l, int r, vector<string>& keyspace) {
	if(l == r)
	{
		keyspace.push_back(a);
	}
	else
	{
		for(int i = l; i <= r; i++)
		{
			swap(a[l], a[i]); //inbuilt swap function
			permute(a, l+1, r, keyspace);
			swap(a[l], a[i]);
		}
	}
}
vector<string> genKeySpace(string plaintext)
{
	set<char> uniqSet;
	vector<string> keyspace;
	int count = 0;
	for(int i=0; i < plaintext.length(); i++)
	{
		uniqSet.insert(plaintext[i]);
	}
	for(set<char>::iterator it = uniqSet.begin(); it != uniqSet.end(); it++)
	{
```

```
		uniqtext[count++] = (*it);
	}
	permute(uniqtext, 0, strlen(uniqtext)-1, keyspace);
	return keyspace;
}
string encrypt(string unique, string key)
{
	string plaintext = readPlainText();
	string ciphertext = "";

	for(int i=0; i < plaintext.length(); i++)
	{
		int idx = unique.find(plaintext[i]);
		ciphertext += key[idx];
	}
	return ciphertext;
}
void showFrequency(string pt , string ct)
{
	map<char , int> mPlain ;
	map<char , int> mCipher ;
	for(int i =0 ;i < pt.length() ; i++)
	{
		mPlain[pt[i]]++ ;
		mCipher[ct[i]]++ ;
	}
	cout<<"\nFrequency\t\tPlaintext Character\t\tCiphertext character" <<endl;
	cout<<"=========\t\t===================\t\t===================" <<endl;
	for(int i=0 ; i<pt.length() ; i++)
	{
		cout<< (float)mPlain[pt[i]]/pt.length() << "\t\t\t\t" << pt[i] << "\t\t\t\t" << ct[i] << endl ;
	}
}
int main()
{
	srand(time(NULL)) ;
	string plaintext = readPlainText() ;
	vector<string> keyspace = genKeySpace(plaintext);
	string key = keyspace[rand()%keyspace.size()];
	cout<<"Unique chars = \t" << uniqtext <<endl;
	cout<<"Chosen key = \t" << key <<endl;
	string ciphertext = encrypt(uniqtext , key) ;
	writecipherText(ciphertext);
	showFrequency(plaintext , ciphertext) ;
}
```

Output Screenshots:

```
user@linux-OptiPlex-5090: ~/1SI19CS090
user@linux-OptiPlex-5090:~/1SI19CS090$ g++ monoalphabetic.cpp
user@linux-OptiPlex-5090:~/1SI19CS090$ ./a.out
Unique chars =  Hdelorw
Chosen key =    oeHdwrl

Frequency               Plaintext Character             Ciphertext character
=========               ===================             ====================
0.1                             H                               o
0.1                             e                               H
0.3                             l                               d
0.3                             l                               d
0.2                             o                               w
0.1                             w                               l
0.2                             o                               w
0.1                             r                               r
0.3                             l                               d
0.1                             d                               e
user@linux-OptiPlex-5090:~/1SI19CS090$
```


Open
plaintext.txt
~/1SI19CS090
Save
1 Helloworld
Plain Text
Tab Width: 8
Ln 1, Col 11
INS


Open
ciphertext.txt
~/1SI19CS090
Save
1 oHddwlwrde
Plain Text
Tab Width: 8
Ln 1, Col 1
INS