|  |  |
| --- | --- |
| Name | Krish Srivastava |
| Register Number | AP21110010302 |
| Section | CSE - E |

**Lab-3**

**Decrypt the following cipher by finding its key length and key.**

**ULE PSO ENG LII WREBR RHLSMEWE XHH DFXTHJ GVOP LII PRKU SFIADI**

**int Procedure-for-key-length(Cipher-text c):**

**1. Look at the sequence of ciphertext characters c(1), c(1+t)...... for t in {1,2,3...} and tabulate the frequencies of characters q\_0,q\_1,....q\_25 for this sequence.**

**2. Then compute sum of (q\_i)^2. As t becomes the length of the key then we expect the sum close to 0.065. If we get such a t then return that t.**

#include <stdio.h>

#include <string.h>

#include <math.h>

void calculateFrequencies(const char \*text, int interval, int \*frequencies) {

int len = strlen(text);

for (int i = 0; i < len; i += interval) {

char c = text[i];

if (c >= 'A' && c <= 'Z') {

frequencies[c - 'A']++;

}

}

}

int findProbableKeyLength(const char \*cipherText) {

double targetSum = 0.065;

int bestLength = -1;

double bestDifference = 1000.0; // A large initial difference

int textLength = strlen(cipherText);

for (int interval = 1; interval <= textLength; interval++) {

int frequencies[26] = {0}; // Initialize frequencies array

calculateFrequencies(cipherText, interval, frequencies);

double sumOfSquares = 0;

for (int i = 0; i < 26; i++) {

sumOfSquares += pow(frequencies[i], 2);

}

sumOfSquares /= (textLength / interval);

double difference = fabs(sumOfSquares - targetSum);

if (difference < bestDifference) {

bestDifference = difference;

bestLength = interval;

}

}

return bestLength;

}

void decryptText(const char \*cipherText, const char \*key) {

int textLength = strlen(cipherText);

int keyLength = strlen(key);

printf("Decrypted Text:\n");

for (int i = 0; i < textLength; i++) {

if (cipherText[i] >= 'A' && cipherText[i] <= 'Z') {

char decryptedChar = (char)('A' + (cipherText[i] - key[i % keyLength] + 26) % 26);

printf("%c", decryptedChar);

} else {

printf("%c", cipherText[i]);

}

}

printf("\n");

}

int main() {

const char \*cipherText = "ULE PSO ENG LII WREBR RHLSMEWE XHH DFXTHJ GVOP LII PRKU SFIADI";

int probableKeyLength = findProbableKeyLength(cipherText);

printf("Probable Key Length: %d\n", probableKeyLength);

const char \*probableKey = "THE";

decryptText(cipherText, probableKey);

return 0;

}

**OUTPUT :**

