****

**MINI PROJECT**

**ON**

**ROT13 CRYPTOGRAPHY**

**Submitted By: Madhura Bhat**

**SF ID:104873**

**Table of Contents:**

|  |  |  |
| --- | --- | --- |
| **SL No.** | **Contents** | **Page No.** |
| 1 | Problem Statement | 3 |
| 2 | Description | 3-4 |
| 3 | Requirements | 5 |
| 4 | Test Plan | 5-6 |
| 5 | Test Case | 6-7 |
| 6 | Expected Final Outcome | 7 |

**ROT13 CRYPTOGRAPHY**

**Problem Statement :**

To replace the existing character with a character that lies 13 position from it and the same action is used for both encryption and decryption.

**ROT13** has been described as the "Usenet equivalent of a magazine printing the answer to a quiz upside down". ... In **rot13** encryption each character is replaced by the character that's 13-letters away.

**Description :**

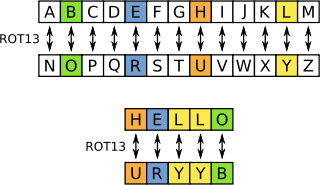
ROT13 is a cryptographic method using a 13-step shift. ROT13 is not intended to be used where secrecy is of any concern-the use of a constant shift means that the encryption effectively has no key and decryption requires no more knowledge than the fact that ROT13 is in use. Even without this knowledge, the algorithm is easily broken through frequency analysis.

ROT13 is used in online forums as a means of hiding spoilers, punchliners, puzzle solution, and offensive materials. Email addresses are also sometimes encoded with ROT13 to hide from less sophisticated spam bots. ROT13 is supported as a built-in feature to news reading software.

Applying ROT13 to a piece of text merely requires examining its alphabetic characters and replacing each one by the letter 13 places further along in the alphabet, wrapping back to the beginning if necessary. A becomes N, B becomes O, and so on up to M, which becomes Z, then the sequence continues at the beginning of the alphabet: N becomes A, O becomes B, and so on to Z, which becomes M. Only those letters which occur in the English alphabet are affected; numbers, symbols, whitespace, and all other characters are left unchanged. Because there are 26 letters in the English alphabet and 26 = 2 × 13, the ROT13 function is its own inverse.



Two successive applications of ROT13 restore the original text. The transformation can be done using a lookup table, shown below.



**Requirements:**

* **User Requirement:** Encrypt and Decrypt Data with ROT13 logic.
* **Technical Requirement**:

1. Operating system: Windows 7,8 or 10
2. Editor: DEV C++
3. Programming Language: C

Dev-C++ is a free full-featured integrated development environment (IDE) distributed under the GNU General Public License for programming in C and C++. It is written in Delphi.

It is bundled with, and uses, the MinGW or TDM-GCC 64bit port of the GCC as its compiler. Dev-C++ can also be used in combination with Cygwin or any other GCC-based compiler.

Dev-C++ is generally considered a Windows-only program, but there are attempts to create a Linux version: header files and path delimiters are switchable between platforms.

**Test Plan :**

**Description:** This provides details on how the testing process will be conducted for a given project.

**Test Objectives:** The following will ensure that the project under test is error free.

1.To check if data to be encrypted using ROT13 is taken.

2.To check whether ROT13 is applicable for capital characters.

3.To check whether ROT13 is applicable for small characters.

4.To check the encryption and decryption one should enter characters to be encrypted first and then enter the character to be decrypted.

5.check if the original data retains the same after the process.

**Test Cases :**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Test Cases** | **Expected outcome** | **Status** |
| 1 | Test for characters entered. | If character is entered then it should be shifter by 13 otherwise it won’t give anything. | Pass/Fail |
| 2 | Test for capital characters in encryption | Displays encrypted output for capital characters. | Pass/Fail |
| 3 | Test for small characters in encryption | Displays encrypted output for small characters. | Pass/Fail |
| 4 | Test for capital characters in decryption | Displays decrypted output for capital characters. | Pass/Fail |
| 5 | Test for small characters in decryption | Displays decrypted output for small characters | Pass/Fail |
| 6 | Test whether output is same as input at the end | Displays character encrypted and character decrypted | Pass/Fail |

**Expected Final Outcome:**

The data to be encrypted is given. The characters are then rotated by 13 position to encrypt the data and the same is displayed. To decrypt the data of encrypted one the user should give the data which must be decrypted. The decrypted output is then displayed. Check whether the data entered for encryption is same as the data which is displayed after decryption. Here same logic is used for both encryption and decryption.