

# A PROJECT REPORT ON SQUARE CODE ENCODER

*SUBMITTED BY*  
**SURYANSH BANSAL (23/EC/209)**



**DELHI TECHNOLOGICAL UNIVERSITY**

(FORMERLY DELHI COLLEGE OF ENGINEERING)

BAWANA ROAD, DELHI – 110042

NOVEMBER 2024

## **CANDIDATE’S DECLARATION**

I, **Suryansh Bansal (23/EC/209)**, student of 1st year B. TECH ECE declare that the Project Report titled “**SQUARE CODE ENCODER**” which is submitted by me to Department of Electronics and Communication, Delhi Technological University, Delhi, is original and not copied from any source. This work has not previously formed the basis for the award of any Degree, Diploma, Fellowship or other similar title or recognition.

**Place:** DTU, Delhi

**Date:** 05/11/2023

# INTRODUCTION

The **Square Code Encoder** project is designed to encode a given input text using a classic encryption method that rearranges the characters in a square matrix format and reads them vertically. This encoding is a simple yet effective way of scrambling messages, making them less readable without reversing the process.

The encoder normalizes the input by removing non-alphanumeric characters and converting the text to lowercase. It then calculates the smallest possible square or rectangle dimensions required for encoding and reads the message column-wise to produce the output.

## OBJECTIVES

The main objectives of this project are:

- To create an application that encodes input text using the square code method.
- To implement a user-friendly C program that demonstrates simple text processing.
- To apply string and mathematical operations in C programming effectively.

# ALGORITHM

- **Step 1:** Start the application.
- **Step 2:** Take input text from the user.
- **Step 3:** Normalize the input by removing non-alphanumeric characters and converting it to lowercase.
- **Step 4:** Determine the number of rows and columns for the encoding matrix.
- **Step 5:** Fill the matrix with the normalized text row-wise.
- **Step 6:** Read and collect the characters column-wise to generate the encoded output.
- **Step 7:** Print the encoded message.
- **Step 8:** End the program.

## CODE IMPLEMENTATION

```

#include <stdio.h>
#include <math.h>

int stringLength(const char *str)
{
    int length = 0;

    while (str[length] != '\0')
    {
        length++;
    }

    return length;
}

void normalizeText(const char *input, char *normalized)
{
    int index = 0;

    for (int i = 0; input[i] != '\0'; i++)
    {
        if ((input[i] >= 'A' && input[i] <= 'Z') ||
            (input[i] >= 'a' && input[i] <= 'z') ||
            (input[i] >= '0' && input[i] <= '9'))
        {
            if (input[i] >= 'A' && input[i] <= 'Z')
            {
                normalized[index++] = input[i] + 32;
            }
            else
            {
                normalized[index++] = input[i];
            }
        }
    }

    normalized[index] = '\0';
}
```

```

void encodeSquareCode(const char *normalized, char *encoded)
{
    int length = strlen(normalized);
    int columns = (int)ceil(sqrt(length));
    int rows = (length + columns - 1) / columns;

    int k = 0;

    for (int c = 0; c < columns; c++)
    {
        for (int r = 0; r < rows; r++)
        {
            int pos = r * columns + c;
            if (pos < length)
            {
                encoded[k++] = normalized[pos];
            }
        }
        encoded[k++] = ' ';
    }
    encoded[k - 1] = '\0';
}

int main()
{
    char input[256];
    printf("Enter the input text: ");
    fgets(input, sizeof(input), stdin);

    for (int i = 0; input[i] != '\0'; i++)
    {
        if (input[i] == '\n')
        {
            input[i] = '\0';
            break;
        }
    }

    char normalized[256];
    normalizeText(input, normalized);

    char encoded[512];
    encodeSquareCode(normalized, encoded);

    printf("Encoded message: %s\n", encoded);

    return 0;
}

```

## RESULTS AND OUTPUT

Upon running the program, users can input a string, and the program will output the encoded message as per the square code method.

### Sample Outputs:

```
Enter the input text: Build your own dreams or someone else will hire you to build theirs
Encoded message: brmoiod uosnlut iwoelth lnrehoe ddslibi yrosrur oemeeis uaewyl
PS C:\Users\91999\OneDrive\Desktop\SURYANSH BANSAL\DTU\CO PROJECT 2>
```

```
Enter the input text: 123456789
Encoded message: 147 258 369
PS C:\Users\91999\OneDrive\Desktop\SURYANSH BANSAL\DTU\CO PROJECT 2>
```

## CONCLUSION

The Square Code Encoder project is an essential tool for anyone interested in learning and applying text encryption techniques. By encoding a given string into a format that rearranges the characters in a grid and reads them column by column, the project demonstrates how simple mathematical operations and text manipulation can be effectively implemented in C programming.

This project provides valuable practical expertise in the following areas:

- String and Text Processing: Understanding how to traverse and manipulate strings to achieve the desired transformation.
- Mathematical Computations in C: Applying mathematical functions such as `'ceil()'` and `'sqrt()'` to determine optimal grid dimensions for encoding.
- Developing User-Friendly Console Applications: Building a program that accepts user input, processes it, and displays results in an easily understandable format.

- **Problem-Solving Skills:** Tackling the challenge of encoding messages using logical and structured approaches to create an efficient algorithm.

Possible enhancements for future versions of this project could include:

- **Expanding Encoding Options:** Introducing alternative encoding patterns, such as spiral or zigzag arrangements, to make the encryption more complex.
- **Decoding Capability:** Adding functionality to decode an encoded message back to its original format.
- **Graphical User Interface (GUI):** Implementing a simple GUI to make the application more user-friendly and accessible to non-programmers.
- **Support for Special Characters:** Extending the program to handle various character sets and languages.
- **Performance Optimization:** Enhancing the code to process larger input strings more efficiently by optimizing space and time complexity.

For developers and enthusiasts who wish to understand and implement character encoding techniques, the Square Code Encoder offers a comprehensive starting point. The project is simple yet showcases how logic, mathematical computation, and C programming can be combined to create practical and educational applications.