# I. Introduction

## 1. Purpose

This document specifies the requirements for a software application designed to process strings in a C# Windows-based application. The objective is to transform input strings, encode characters based on their ASCII values, and provide functionalities for sorting and retrieving character codes.

## 2. Scope

The project will deliver a winform application that operates efficiently within a Windows environment. It will cater to individual users needing string manipulation and encoding functionality, ensuring input validation and security.

# II. Description

## 1. Product

The product will include functionalities to encode input strings based on ASCII value shifts, sort strings alphabetically, and retrieve ASCII codes of input and output strings. The software will ensure accurate data validation and encapsulation.

## 2. Users

The primary users are developers, students, and individuals needing quick and reliable string manipulation tools.

## 3. Operational Environment

The application is designed to run on any standard Windows-based C# compiler with .NET Framework or later versions.

# III. System Features

## 1. Input Validation and Data Handling

The system validates input strings to ensure they contain only uppercase letters and limits length to 40 characters. Integer inputs are constrained to the range [-25, 25].

**The system needs to:**

* Prompt users for correct input if validation fails.
* Display error messages for invalid inputs.

## 2. Encoding Strings Description

This feature shifts each character in the input string based on a user-provided integer value, wrapping around the alphabet if required.

**The system needs to:**

* Replace characters accurately based on the ASCII value shift.
* Output a correctly encoded string.

## 3. Sorting Strings Alphabetically

The system sorts the input string alphabetically.

**The system needs to:**

* Generate a new, sorted string based on the original input string.

## 4. ASCII Code Retrieval

Provides arrays of ASCII values for characters in input and output strings.

**The system needs to:**

* Output ASCII values for input and encoded strings as integer arrays.

# IV. Demonstration

This section highlights screenshots and code snippets for each feature, providing visual and technical evidence of program execution, functionality, and underlying implementation logic.

## 1. Program User Interface

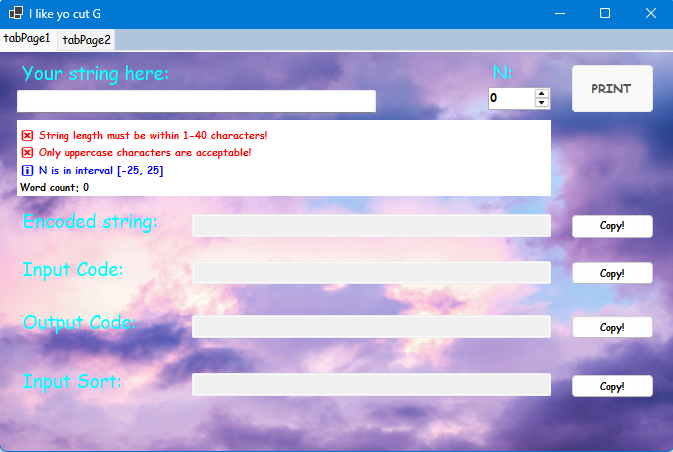


Figure 1 – The user interface of the program

The "String Encoder App" features a user-friendly interface with input fields for strings and numeric shifts, validation guidelines, and output displays for encoded strings, ASCII codes, and sorted strings. Its design ensures clear instructions and efficient functionality, with labeled sections and intuitive controls, offering a seamless experience for string manipulation tasks.

## 2. Validation

a) Code snippet

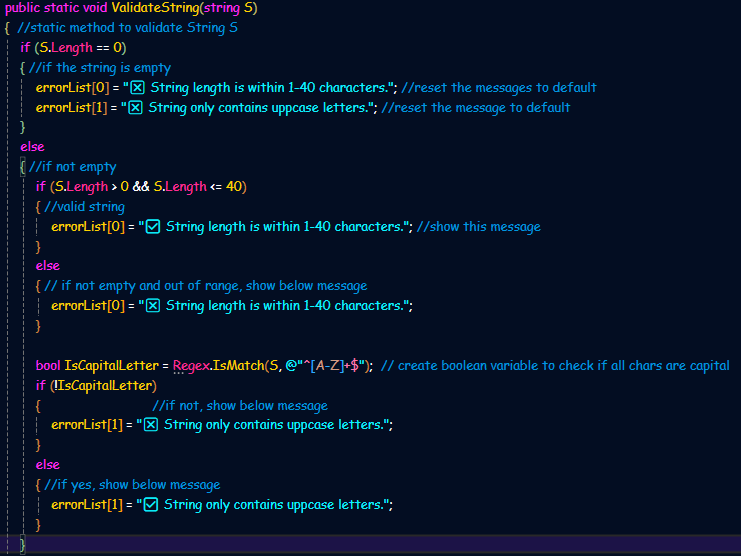


Figure 2 - Code snippet for Validation class

This class validates input strings, ensuring they contain only uppercase letters and do not exceed a maximum length of 40 characters.

b) Demonstration

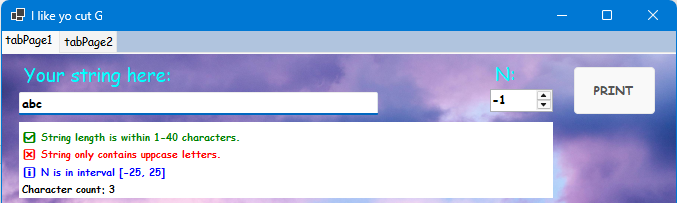


Figure 3 - Input string contains lowercase letter(s)

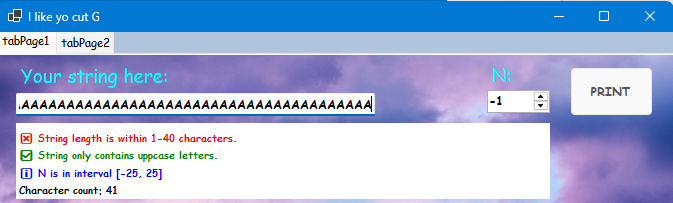


Figure 4 - Input string length exceeding 40 characters

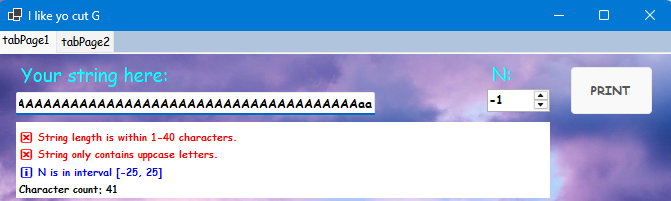


Figure 5 - Both conditions unsatisfied

## 3. Encode string feature

a) Code snippet

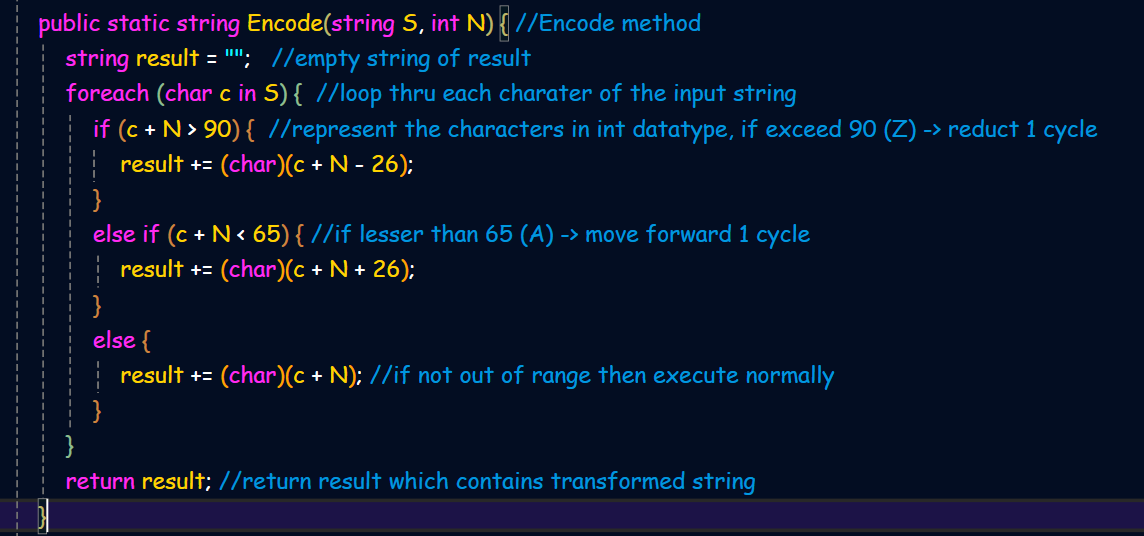


Figure 6 -Code snippet for Encode feature

When the user inputs a valid string and an "N" value (default: 0), the system processes the data and displays the output string upon clicking "PRINT."

b) Demonstration

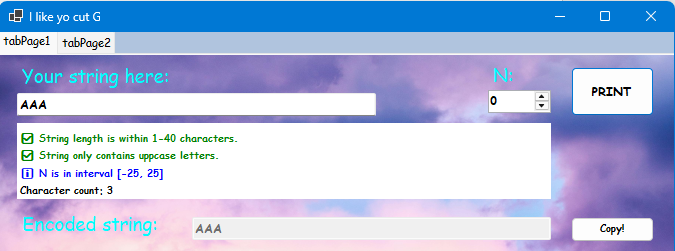


Figure 7 - Input string is "AAA" and N is 0

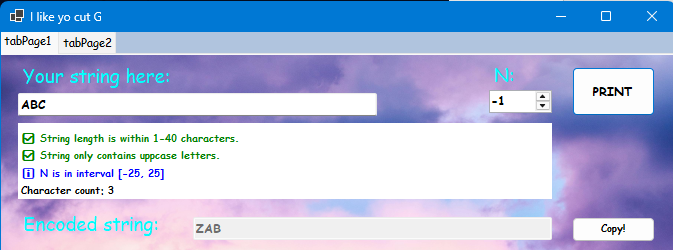


Figure 8 - Input string is “ABC” and N is -1

The system wraps alphabetically during encoding if characters exceed 'Z' or 'A,' ensuring proper cyclical character transformation in outputs.

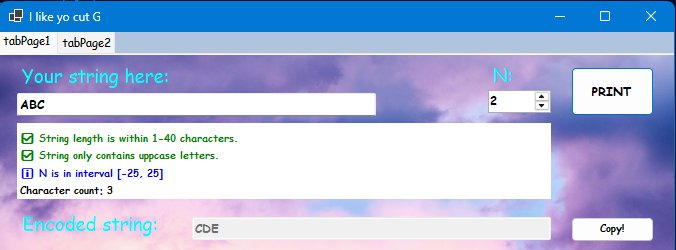


Figure 9 - Input string is "ABC" and N is 2

## 4. InputCode and OutputCode feature

a) Code snippet

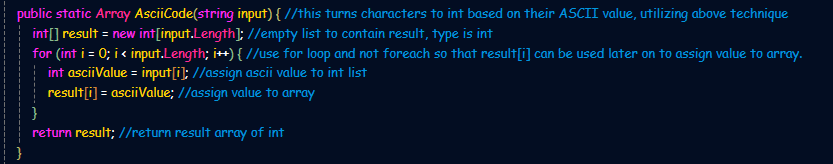


Figure 10 - Code snippet for InputCode and OutputCode

b) Demonstration

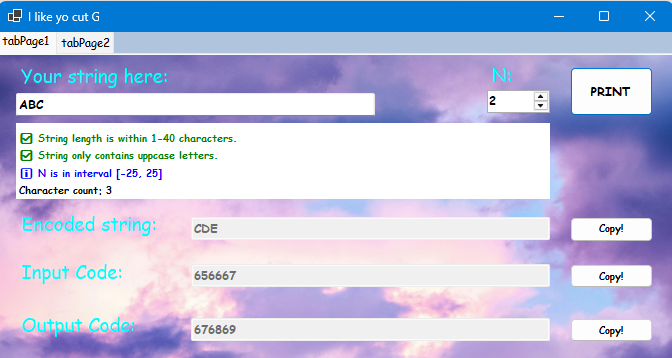


Figure 11 – Demonstration

## 5. InputSort feature

a) Code snippet

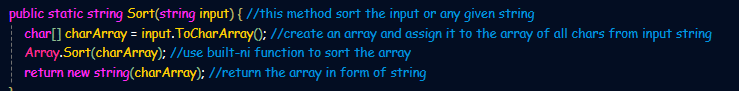


Figure 12 - Code snippet for Sorting feature

b) Demonstration

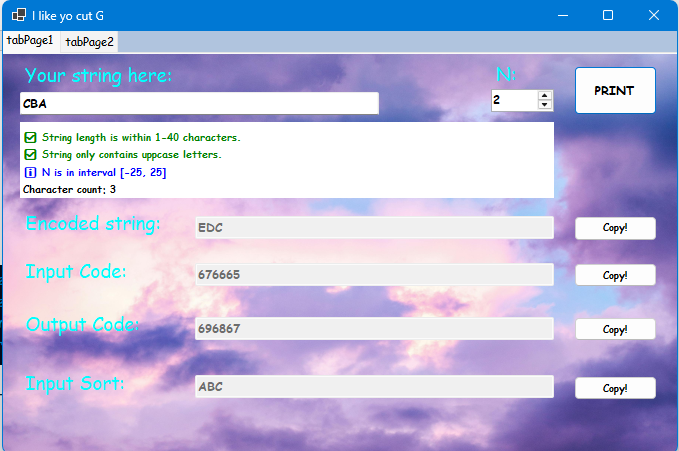


Figure 13 - Input string is "CBA" and when sorted, being "ABC"