Project Report

Spring 2025

Name: Rakshith
Project: Homework 4 - GeoPin

Note: As you write each section, try to be as clear and detailed as possible. Your goal is to communicate your thought process and work clearly. Don't worry if you faced challenges or made mistakes; discussing these is a valuable part of learning and shows your problem-solving skills! Remember, there's no single 'right' way to do these tasks, so be creative and honest in your responses.

Problem Statement (2-3 Paragraphs):

The goal of the program was to take a 5 digit zip code and convert it into a postnet barcode format. This system makes mail sorting faster by encoding digits as patterns. Each barcode also includes a check digit that makes sure the sum of all digits is divisible by 10.

The program asks the user to enter a ZIP code, calculates the check digit, and prints the full barcode with frame bars on both ends. It also shows the zip code and calculated check digit.

Error handling is included to check that the input is exactly 5 digits. If the user enters an invalid ZIP code, the program prints an error message.

Design (1-3 Paragraphs):

The program is divided into two modules: main.py and barcode.py. Main handles user input and the flow of the program whereas the barcode function has helper functions for validation, the check digit calculations, and barcode creations.

The separation made the code cleaner and easier to understand. Each function performs a single task. The main file imports the functions it needs to keep everything simple. I had a dictionary and a list to store encoding for the digits.

The solution is efficient because it uses direct lookups from list instead of long conditionals. If I did it again, I might add a prompt to where it can detect all fake zip codes and print an error message.

Testing (1-2 Paragraphs + screenshots of 3 test cases):

I tested the program by running it multiple times with different types of inputs such as normal, and irregular to test all cases. I tested regular 5 digit codes like 72712 and 01961. The special case I used was 123 which is too short to be a zip code. My program worked as intended in all cases.

Conclusion (1 paragraph)

It met all requirements and produced correct barcodes for the working zip codes. I learned how to separate codes into modules, define clear functions, and handle input error. The project was successful because it matched the test cases and followed the guidelines. In future projects, I would improve the user experience by allowing more than one inputs without having to rerun the program.