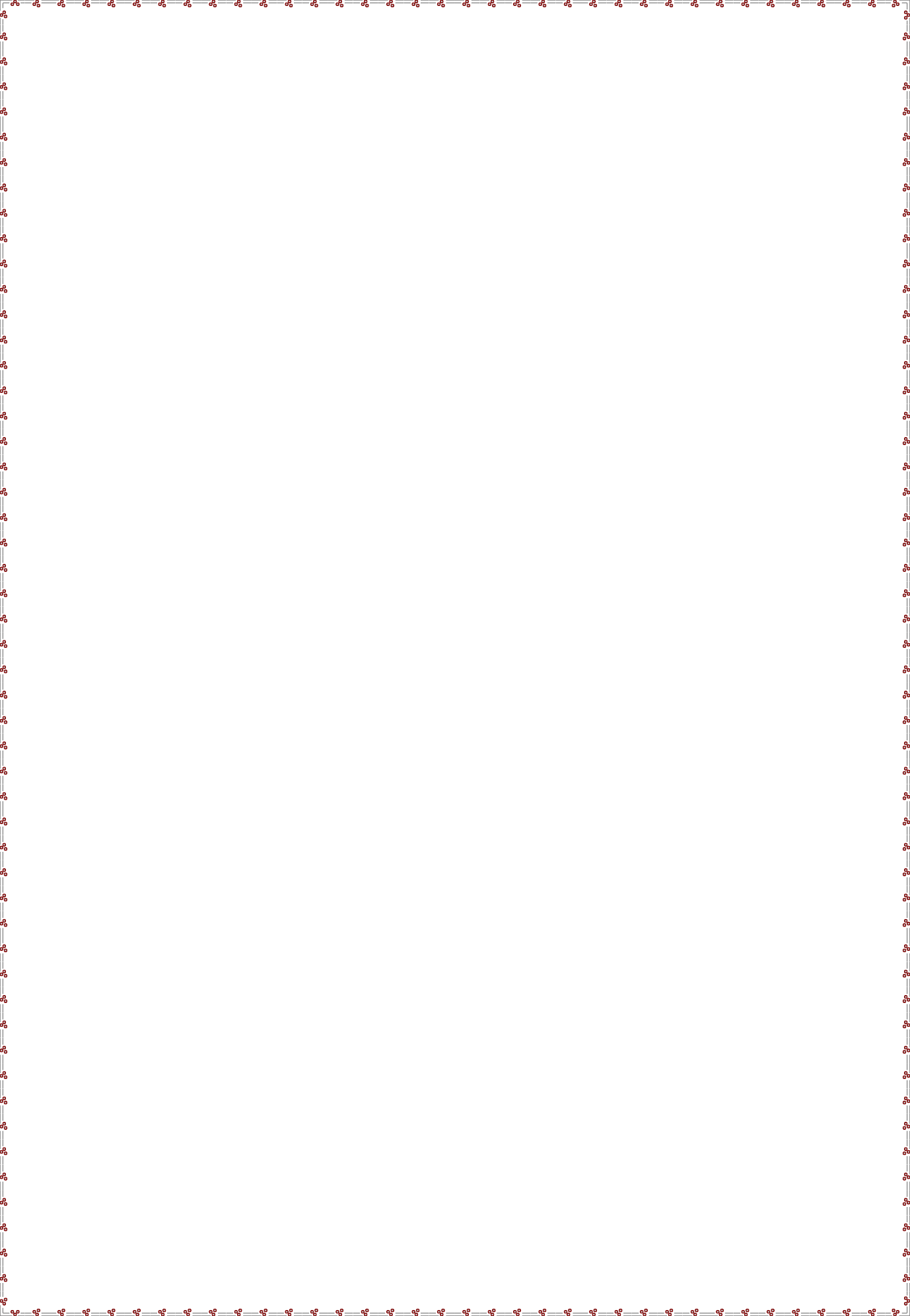
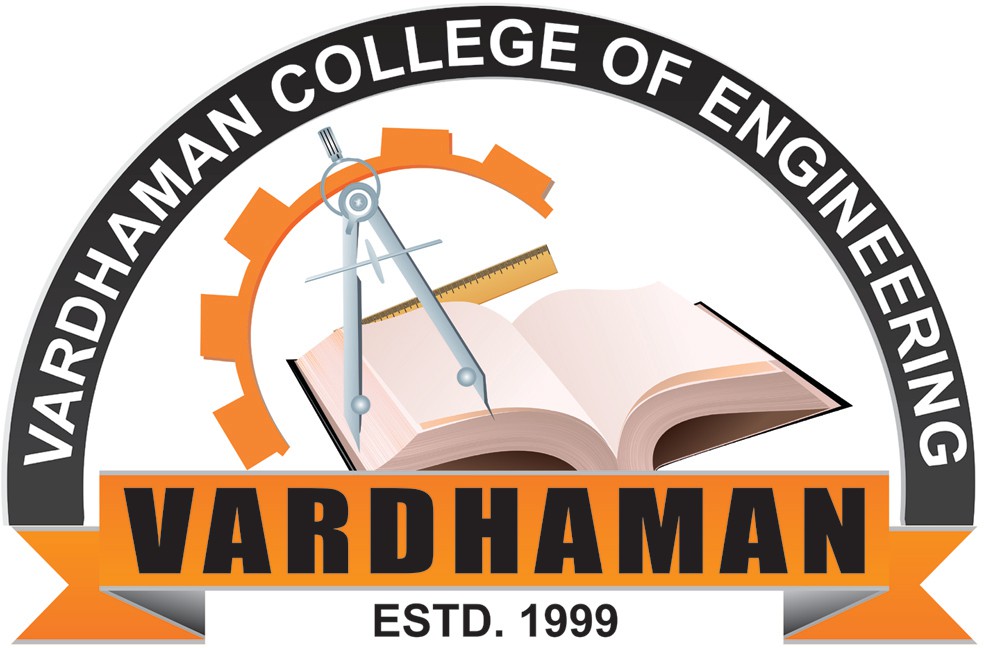
**A Course End Project Report**



**On**

# SPAM MESSAGE FILTERING

# ADVANCED DATA STRUCTURES LABORATORY (A8513)

Submitted in the Partial Fulfilment of the Requirements

for the Award of the Degree of

**Bachelor of Technology**

**IN**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

**K.Sathish 23885A0504**

**Under the Esteemed Guidance of**

**M.Naresh Goud**

**Assistant Professor**

**CSE**

**Department of Computer Science and Engineering**

# VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

(AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified

**Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India**

**January, 2024**

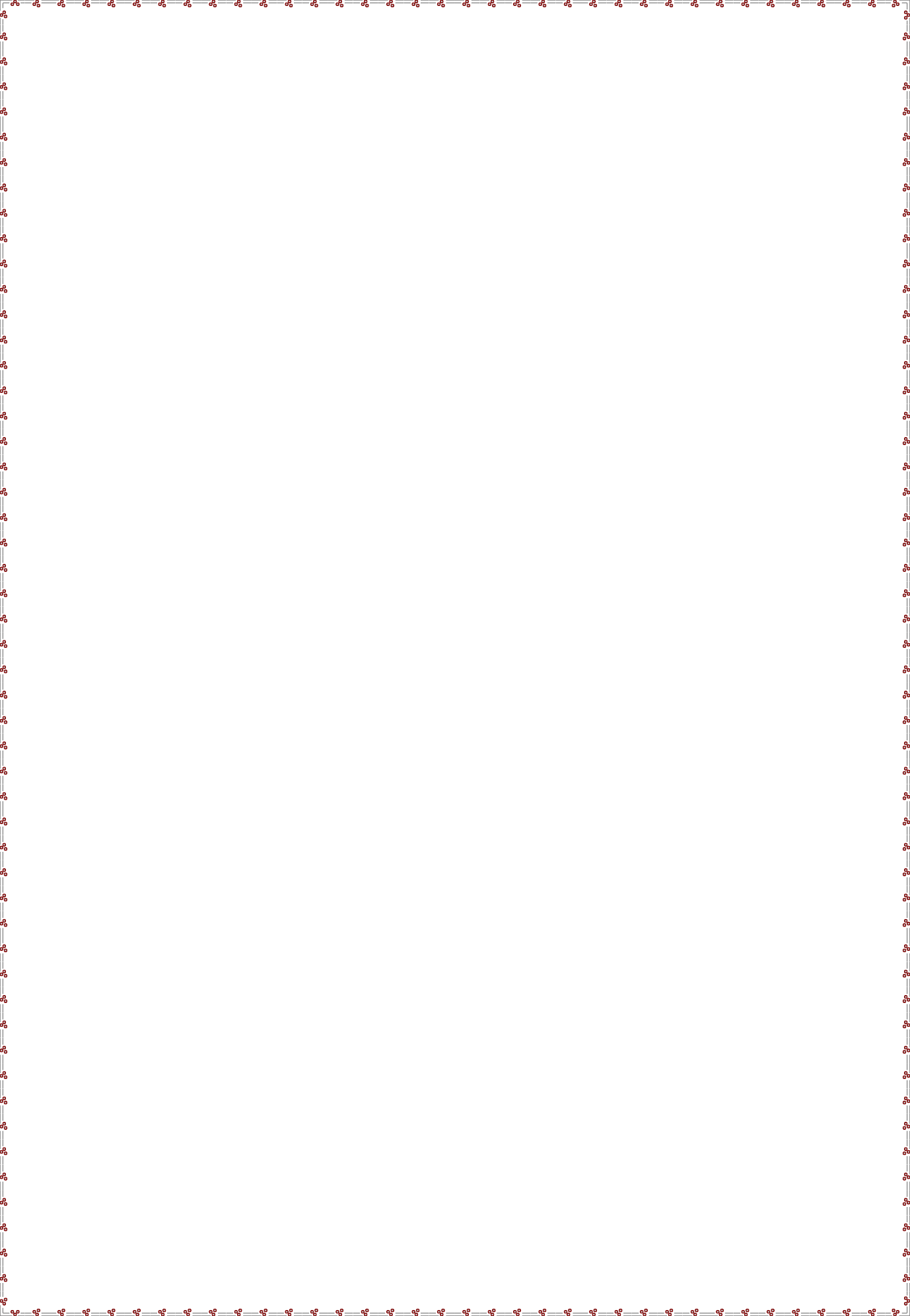
# VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

An autonomous institute affiliated to JNTUH

**Department of Computer Science and Engineering**

**CERTIFICATE**

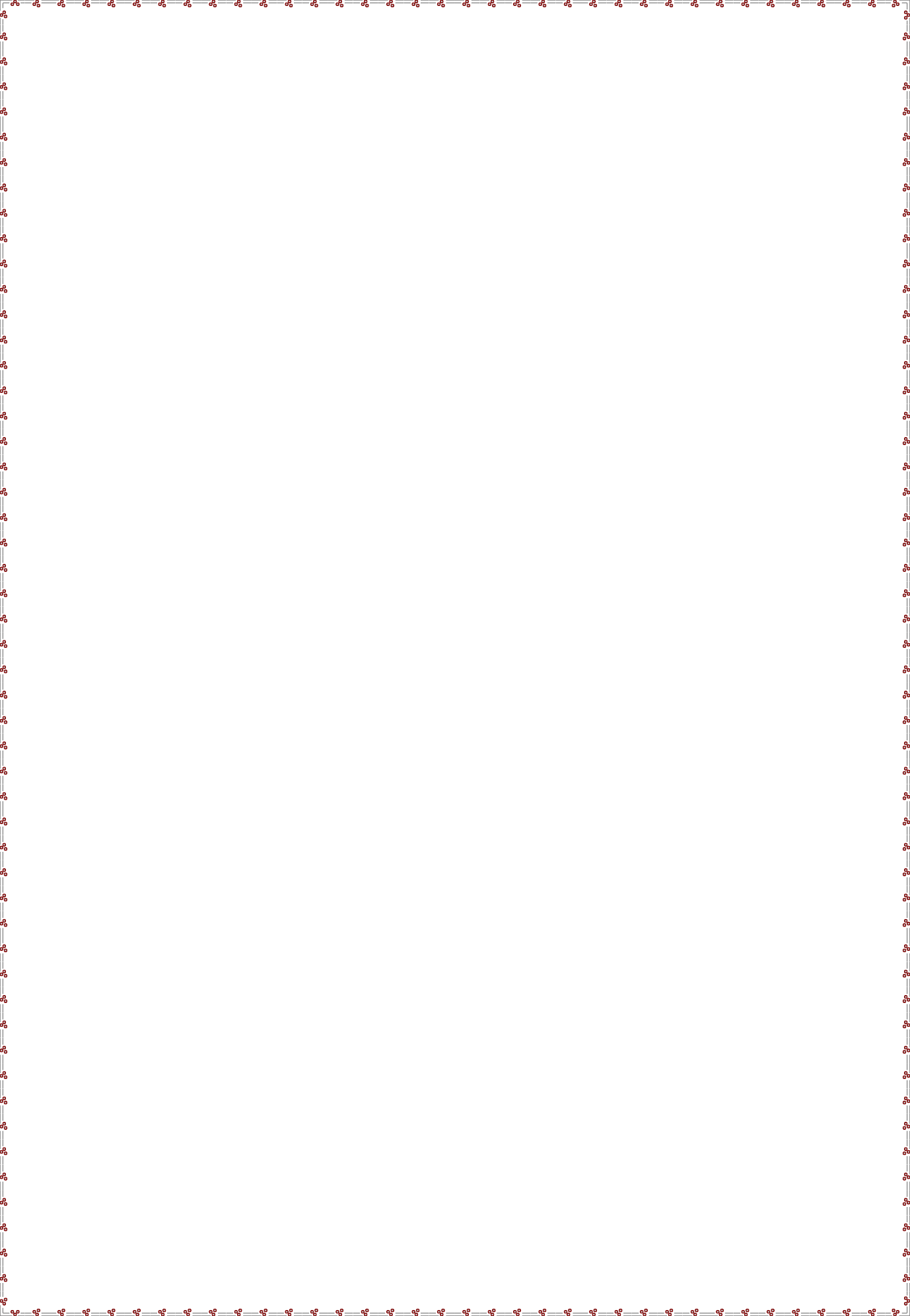
This is to certify that the Course End Project titled **“Spam Message Filtering” is** carried out by Mr./Ms.**K.Sathish**, Roll Number **23885A0504** towards **(A8512)-Advanced Data Structures Laboratory** course and submitted to **Department of Computer Science and Engineering**, in partial fulfilment of the requirements for the award of degree of **Bachelor of Technology** in **Department of Computer Science and Engineering** during the Academic year 2023-24.

**Name & Signature of the Instructor**

## M.Naresh Goud Assistant.ProfessorCSE

**Name & Signature of the HOD**

## Dr. Ramesh Karnati HOD, CSE



**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **M.Naresh Goud**, Professor, Department of Computer Science and Engineering, Vardhaman College of Engineering, for her able guidance and useful suggestions, which helped us in completing the design part of potential project in time.

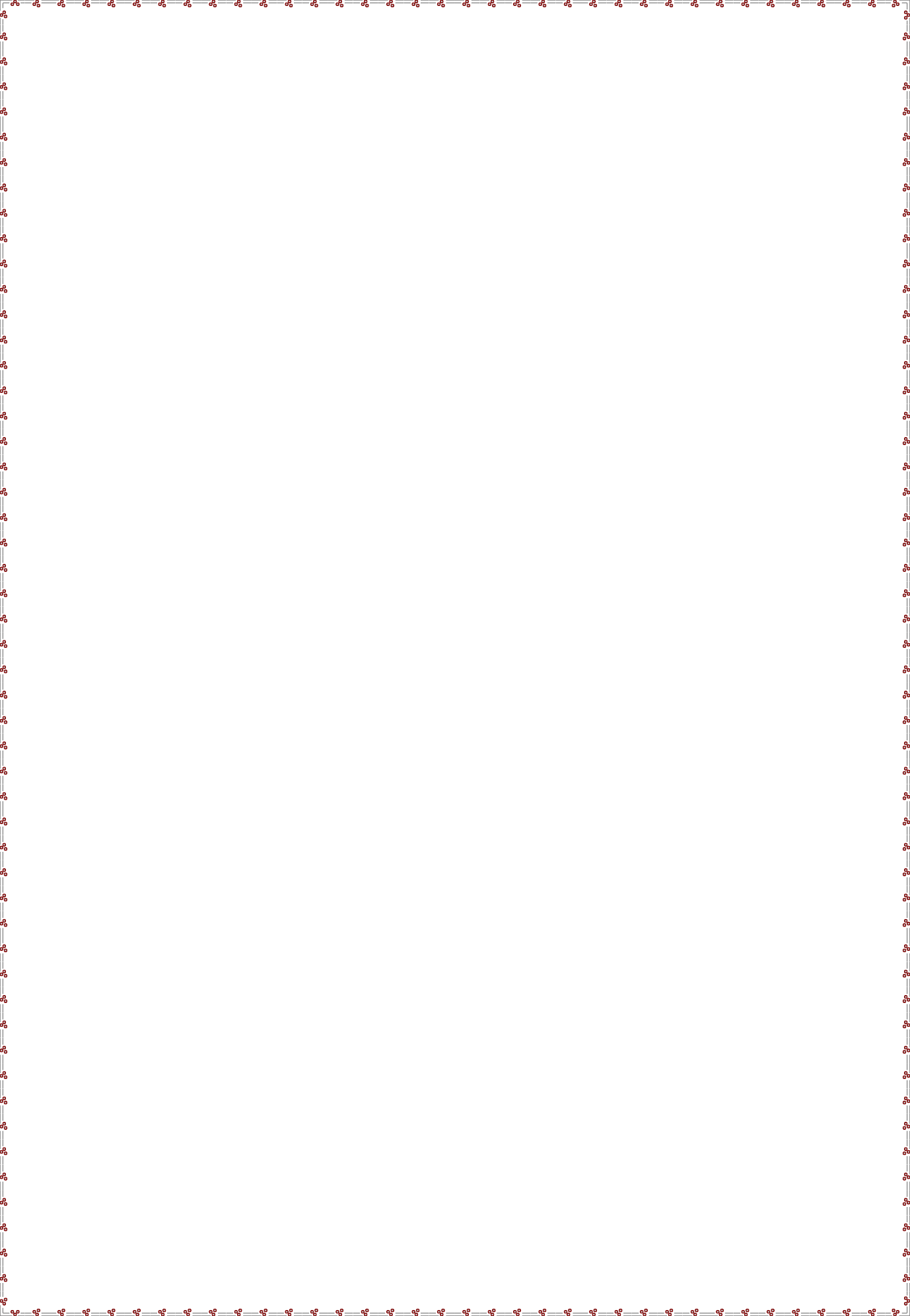
We particularly thankful to **Dr. Ramesh Karnati**, Associate Professor & Head, Department of Computer Science and Engineering for his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R.Ravindra**, for having provided all the facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science and Engineering for their valuable support and generous advice. Finally, thanks to all our friends and family members for their continuous support and enthusiastic help.

## K.Sathish – 23885A0504



***INDEX***

1. Introduction 5
2. Objective of the Project 5
3. Problem statement .......................................................................................................……….6
4. Software and hardware requirements ..........................................................................………..6
5. Project Description …….7
6. Code ….…8
7. Result(s) …...16
8. Conclusion and Future work ….....17
9. References..................................................................................................................... ……17

**Introduction:**

Implement a spam message filtering system in C that uses a hash table to store words and efficiently classify messages as spam or not based on the presence of certain keywords. This project will use a hash table to efficiently store and lookup words for spam classification.

**Objectives :**

The objective of a spam message filtering project in ads is to identify and filter out unwanted, irrelevant, or harmful messages from reaching users through advertising platforms. This project aims to enhance the user experience by ensuring that users are exposed to legitimate and meaningful advertisements while minimizing the impact of spam, scams, or misleading content.

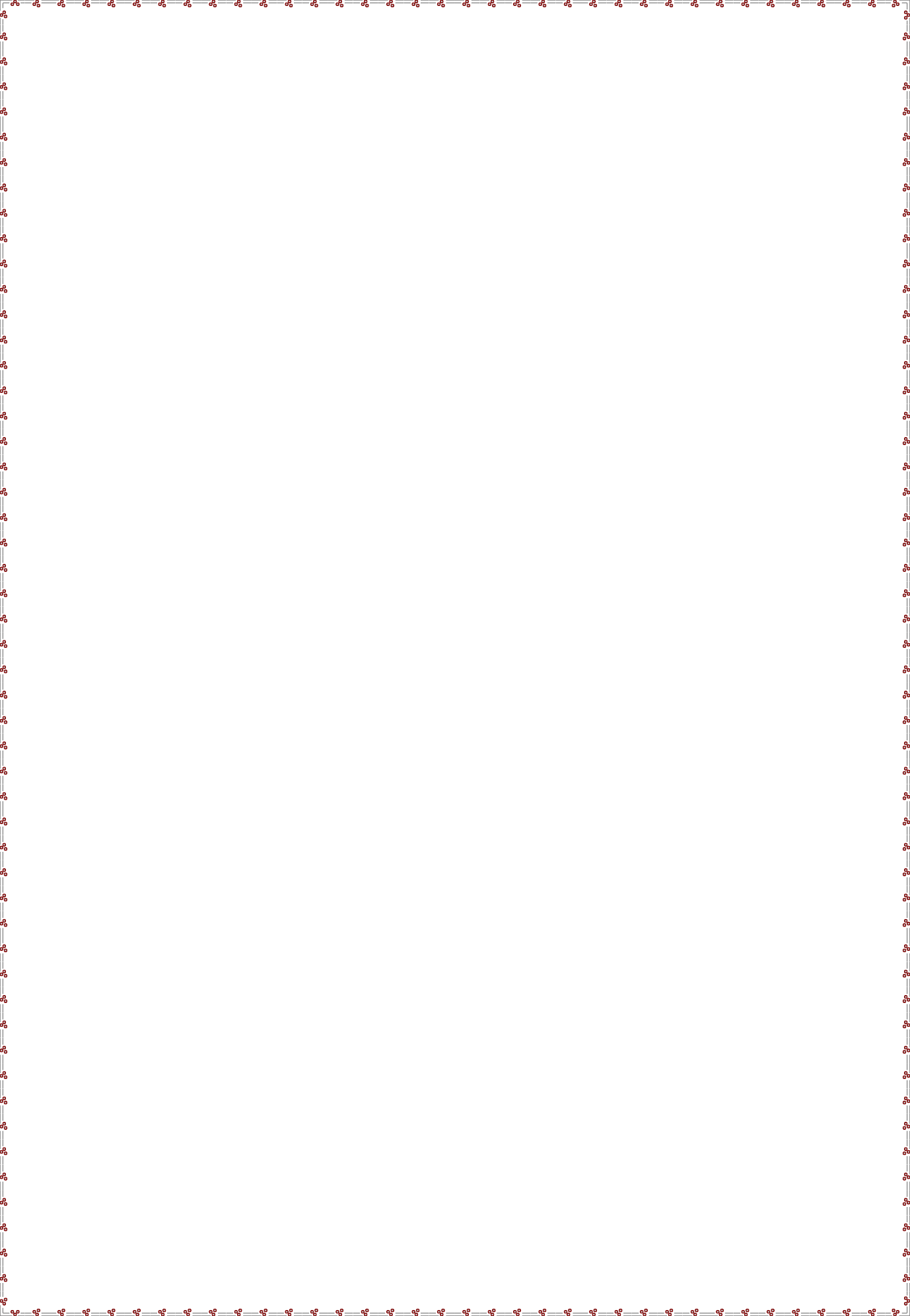
**Core Features:**

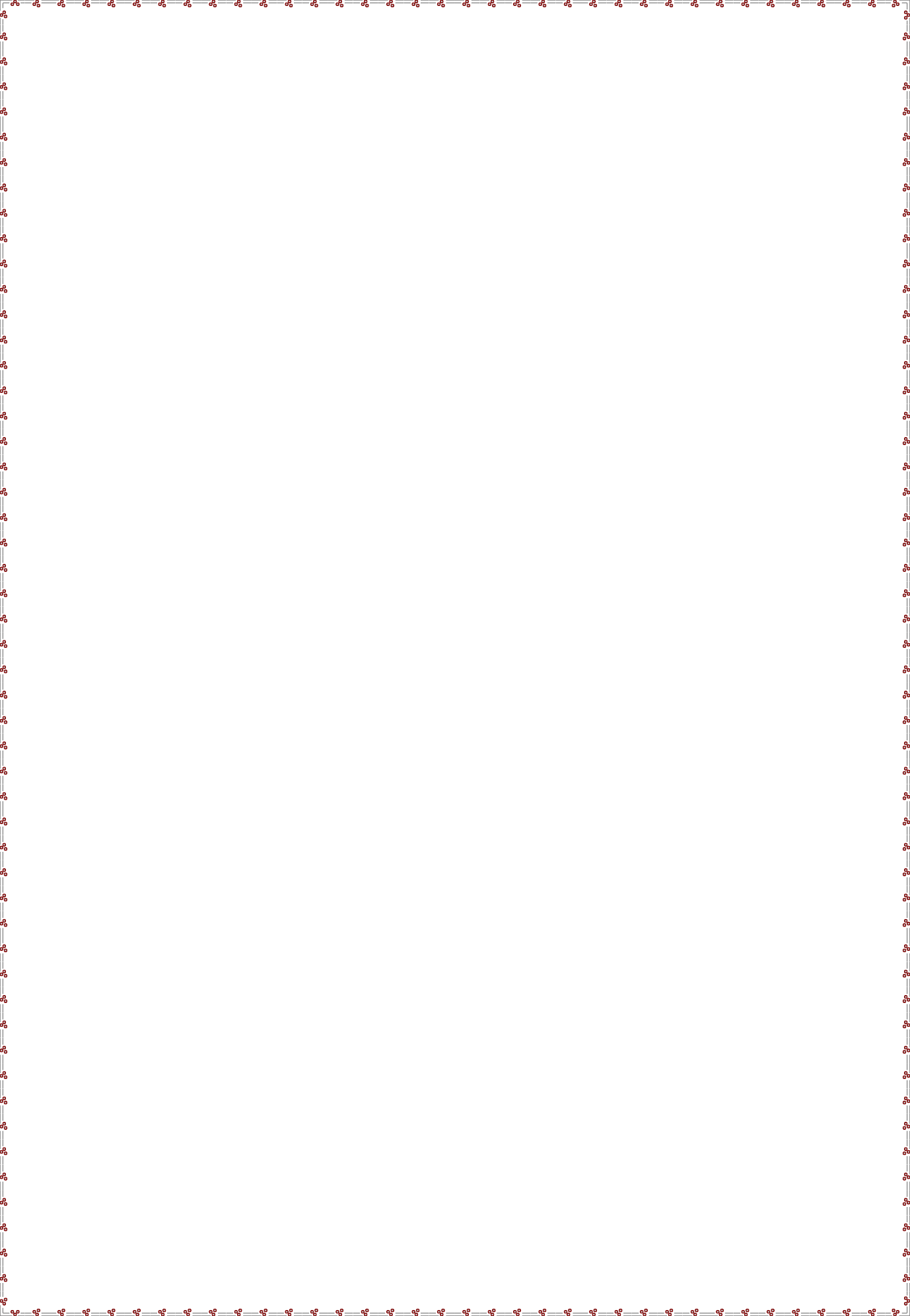
**Message Representation:** Define a data structure to represent SMS messages, including fields such as sender, timestamp, and message content.

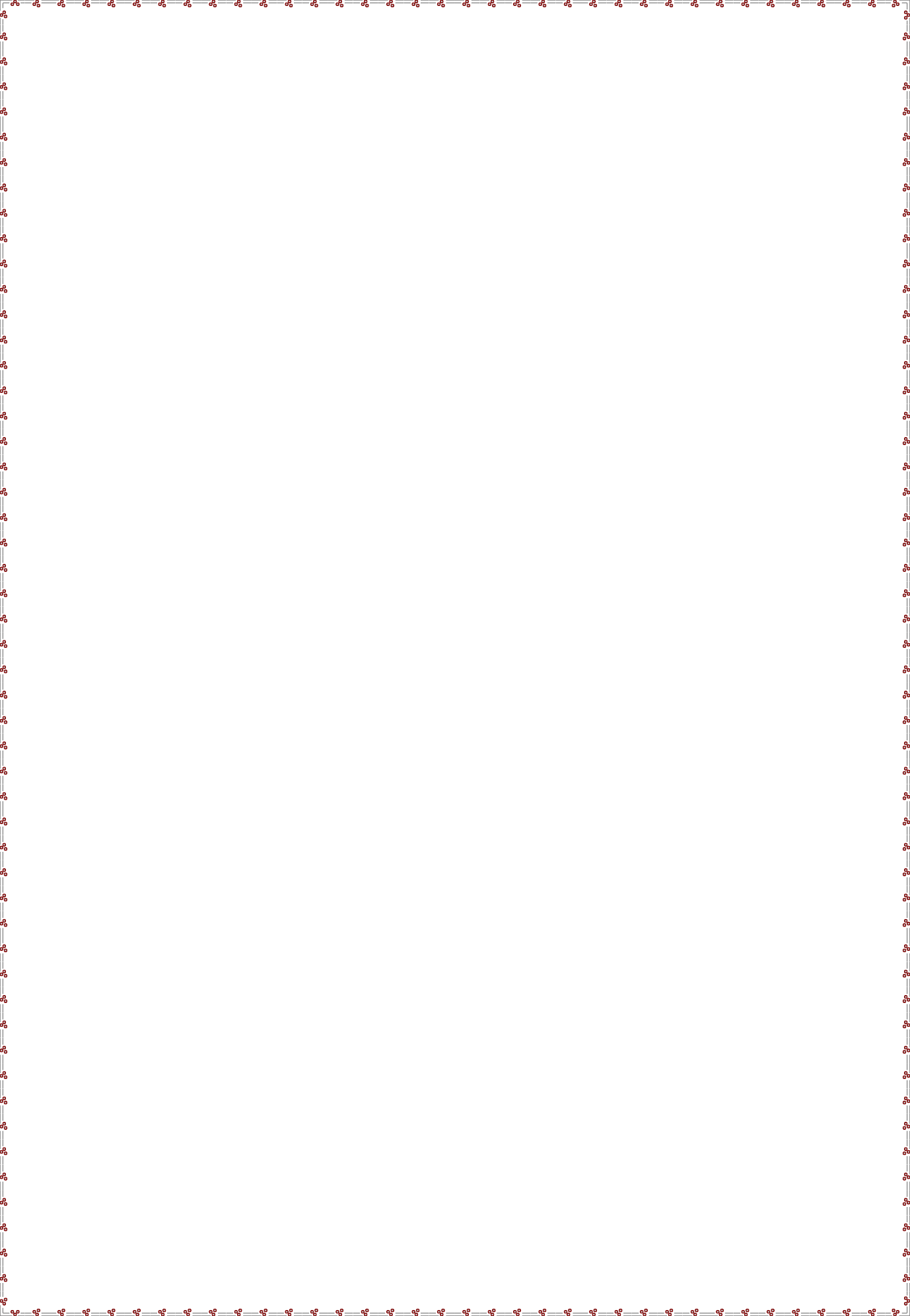
**Data Structures:** Choose appropriate data structures to store and manage SMS messages. Common choices include arrays, linked lists, or dynamic data structures based on the specific needs of your project.

**User Interface:** If applicable, design a simple user interface for users to interact with the application.

**Statistics and Reporting:** Keep track of statistics, such as the number of spam messages detected, false positives, and false negatives. Generate reports for users to review.

**Memory Management :** Implement proper memory allocation and deallocation, especially if dynamic memory is used.



**Problem statement :**

online advertising platform, you are tasked with developing a Spam Message Filtering module to enhance the user experience and maintain the integrity of the platform.

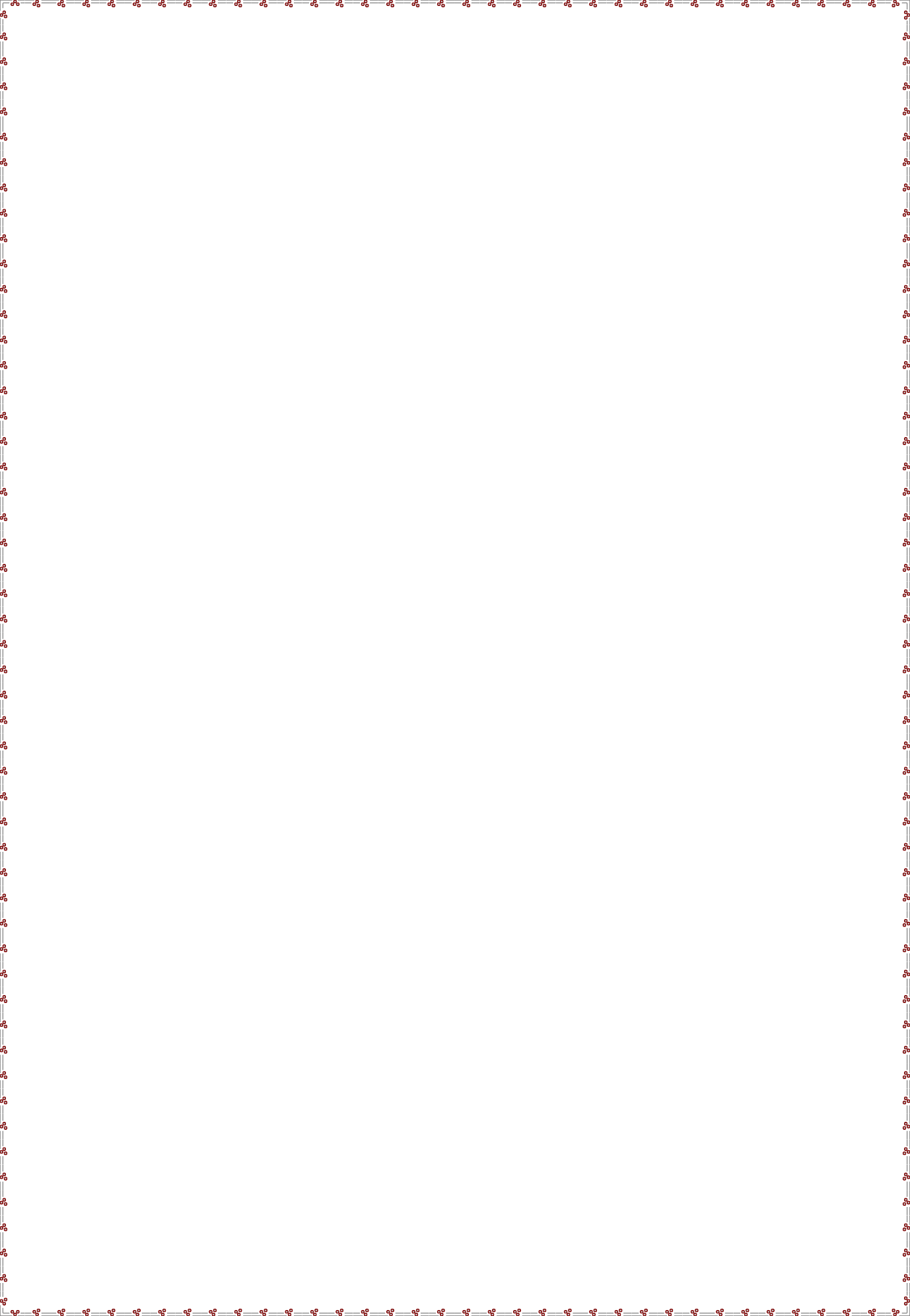
Create a model for spam message filtering in the C language.

**Software and hardware requirements :**

* Operating systems

Windows/mac/linux

* C compiler



**Project Description:**

Linked List Implementation: Tasks are stored and managed using a linked list data structure, ensuring dynamic task addition and removal without fixed-size constraints.

User Interaction: The command-line interface engages users through a simple menu-driven system, guiding them to perform desired operations.

Memory Management: Dynamic memory allocation is utilized for task creation and deallocation to ensure efficient memory usage.

**Main Components:**

**Message Representation:** Define a structure to represent a message. Include fields such as message content, sender, timestamp, etc.

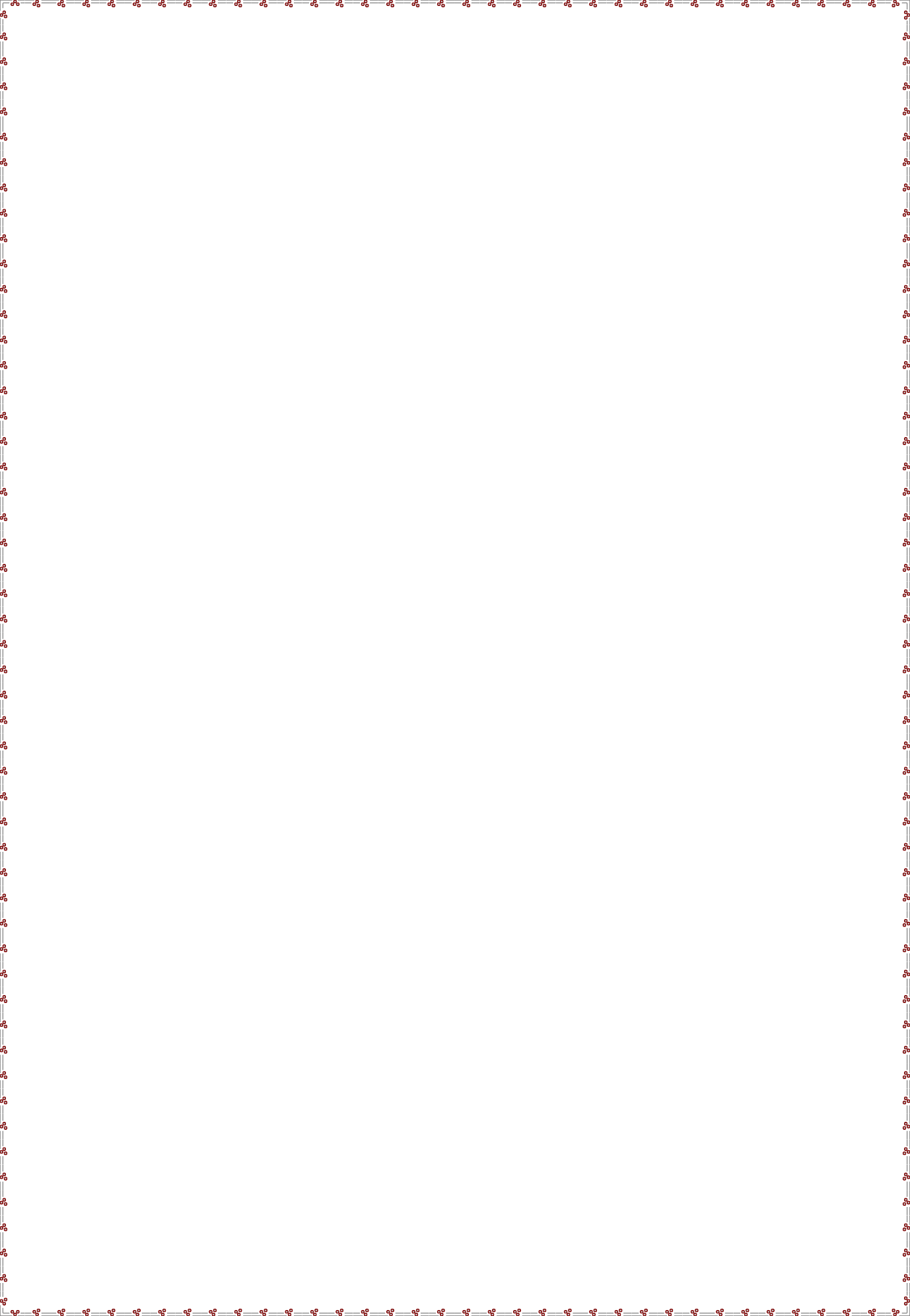
**Trie Implementation:** Implement a Trie data structure to efficiently store keywords for spam classification.

**Spam Filter Functions:** Implement functions to classify messages as spam or not based on the presence of keywords in the Trie.

**Input/Output Functions:** Create functions to read messages from a file or user input, read spam keywords from a file, and display the filtered messages.

**Filtering Logic:** Implement the main logic to filter messages using the spam filter functions and Trie.

**Main Function:** Combine all the components in the main function.



**Source Code:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_SENDER\_LENGTH 20

#define MAX\_TIMESTAMP\_LENGTH 20

#define MAX\_CONTENT\_LENGTH 200

// SMS structure

struct SMS {

char sender[MAX\_SENDER\_LENGTH];

char timestamp[MAX\_TIMESTAMP\_LENGTH];

char content[MAX\_CONTENT\_LENGTH];

};

// Node structure for linked list

struct Node {

struct SMS data;

struct Node\* next;

};

// Linked list structure

struct LinkedList {

struct Node\* head;

};

// Function to create a new node

struct Node\* createNode(const struct SMS\* sms) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

if (newNode != NULL) {

newNode->data = \*sms;

newNode->next = NULL;

}

return newNode;

}

// Function to insert a node at the end of the linked list

void insertNode(struct LinkedList\* list, const struct SMS\* sms) {

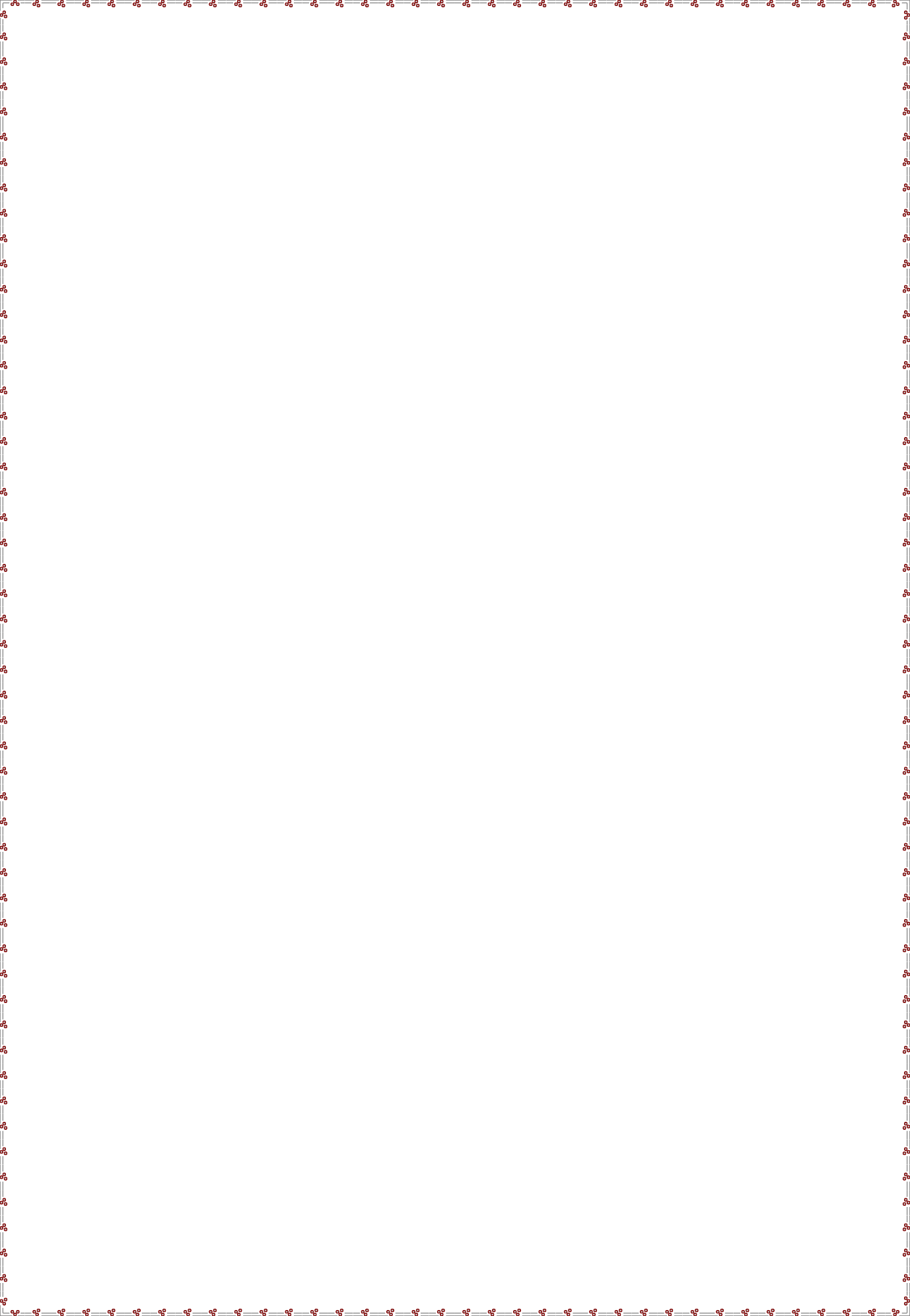
struct Node\* newNode = createNode(sms);

if (newNode == NULL) {

printf("Memory allocation error\n");

exit(EXIT\_FAILURE);

}

 if (list->head == NULL) {

list->head = newNode;

} else {

struct Node\* current = list->head;

while (current->next != NULL) {

current = current->next;

}

current->next = newNode;

}

}

// Function to check if an SMS is spam

int isSpam(const struct SMS\* sms) {

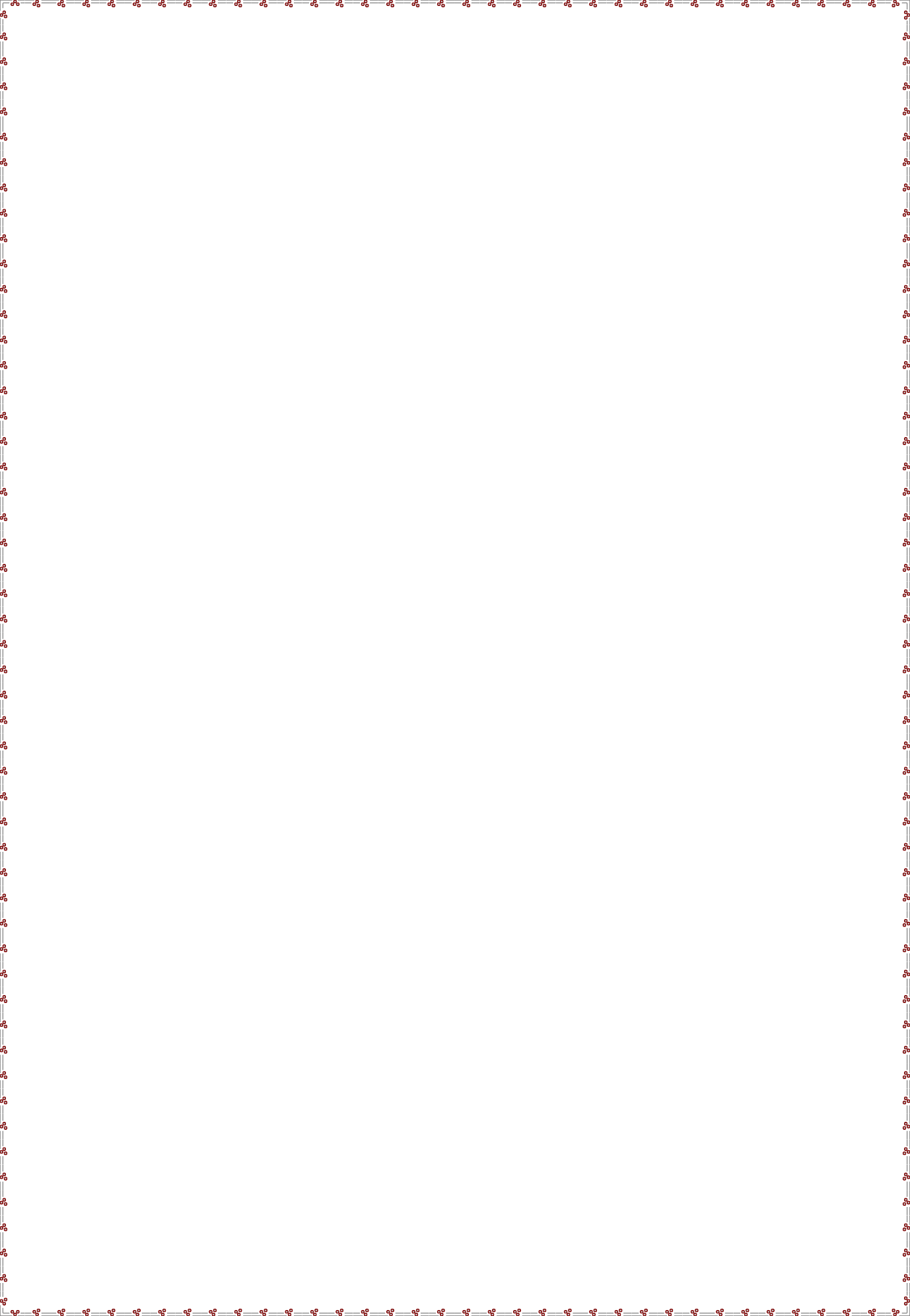
// Implement your spam filtering logic here

// For simplicity, this example considers any message containing "spam" as spam

const char\* keyword = "spam";

return (strstr(sms->content, keyword) != NULL);

}

// Function to filter spam messages and create a new linked list

struct LinkedList\* filterSpam(struct LinkedList\* allMessages) {

struct LinkedList\* filteredMessages = (struct LinkedList\*)malloc(sizeof(struct LinkedList));

if (filteredMessages == NULL) {

printf("Memory allocation error\n");

exit(EXIT\_FAILURE);

}

filteredMessages->head = NULL;

struct Node\* current = allMessages->head;

while (current != NULL) {

if (!isSpam(&current->data)) {

insertNode(filteredMessages, &current->data);

}

current = current->next;

}

return filteredMessages;

}

// Function to display SMS messages

void displayMessages(struct LinkedList\* messages) {

struct Node\* current = messages->head;

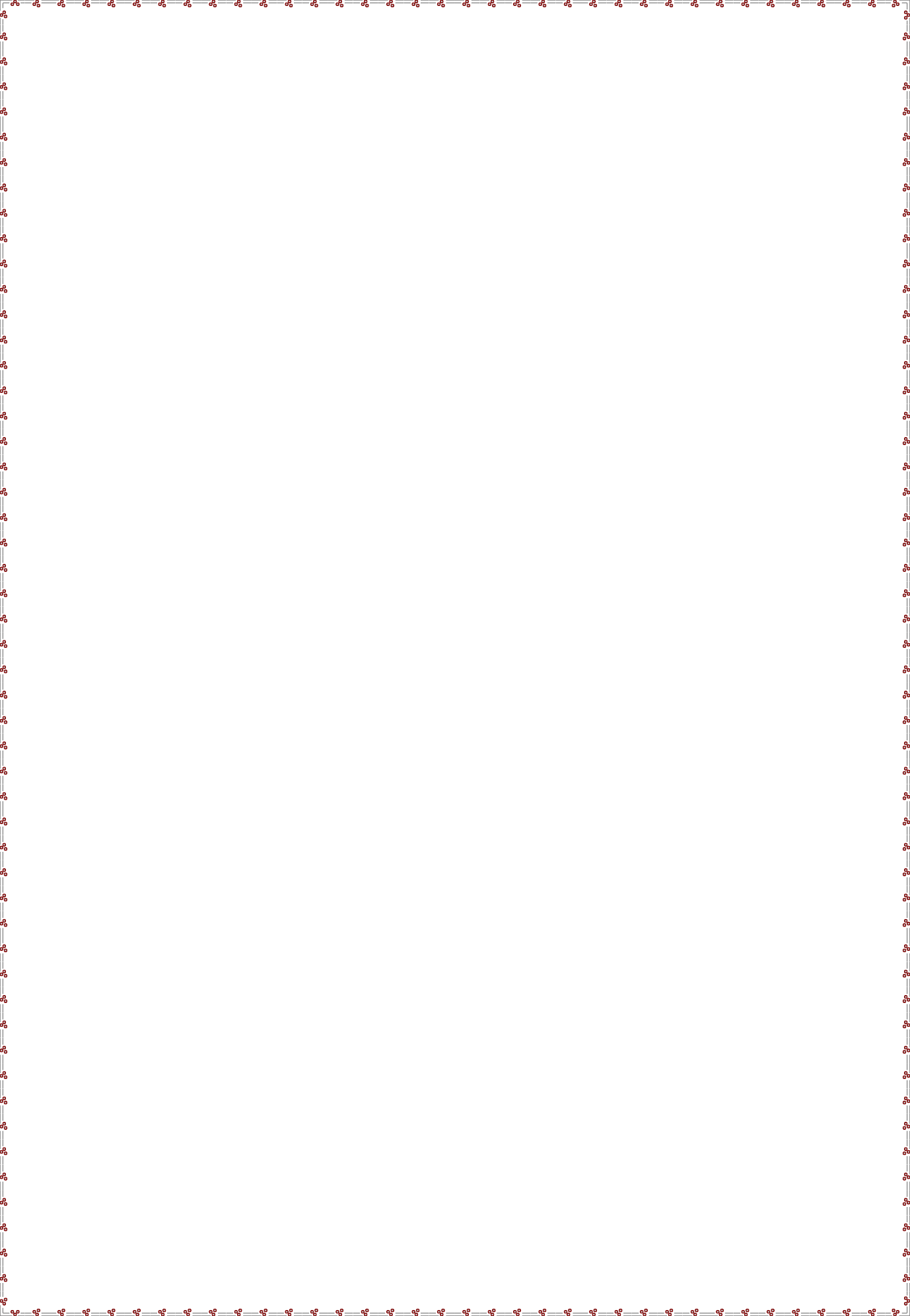
while (current != NULL) {

printf("Sender: %s\nTimestamp: %s\nContent: %s\n\n", current->data.sender, current->data.timestamp, current->data.content);

current = current->next;

}

}



// Function to free memory used by the linked list

void freeLinkedList(struct LinkedList\* list) {

struct Node\* current = list->head;

struct Node\* next;

while (current != NULL) {

next = current->next;

free(current);

current = next;

}

list->head = NULL;

}

// Function to read SMS messages from a file and populate a linked list

struct LinkedList\* readMessagesFromFile(const char\* filename) {

struct LinkedList\* allMessages = (struct LinkedList\*)malloc(sizeof(struct LinkedList));

if (allMessages == NULL) {

printf("Memory allocation error\n");

exit(EXIT\_FAILURE);

}

allMessages->head = NULL;

FILE\* file = fopen(filename, "r");

if (file == NULL) {

printf("Error opening file: %s\n", filename);

exit(EXIT\_FAILURE);

}

struct SMS sms;

while (fscanf(file, "%s %s %[^\n]", sms.sender, sms.timestamp, sms.content) == 3) {

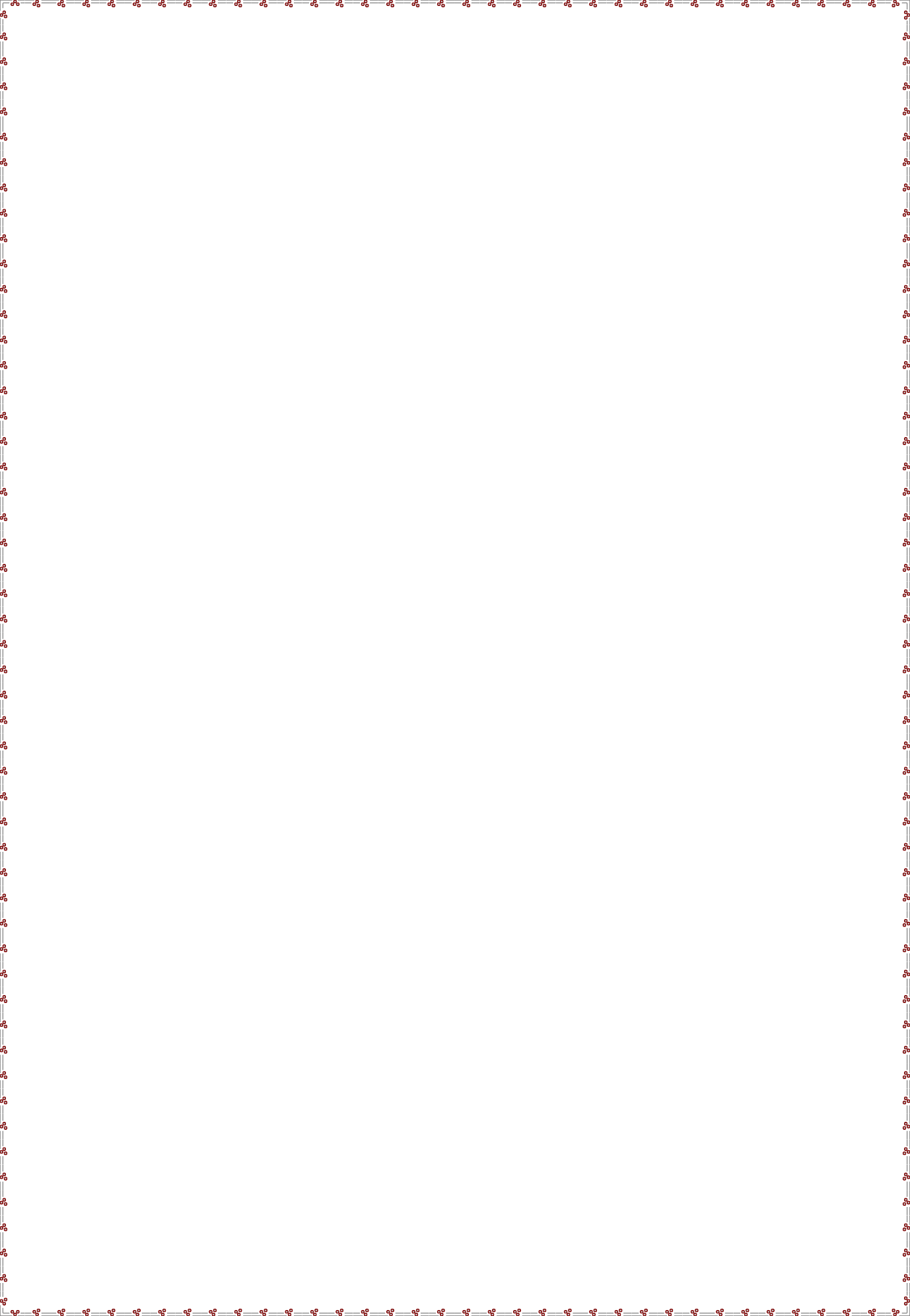
insertNode(allMessages, &sms);

}

fclose(file);

return allMessages;

}



int main() {

const char\* filename = "sms\_messages.txt";

// Step 1: Read SMS messages from a file

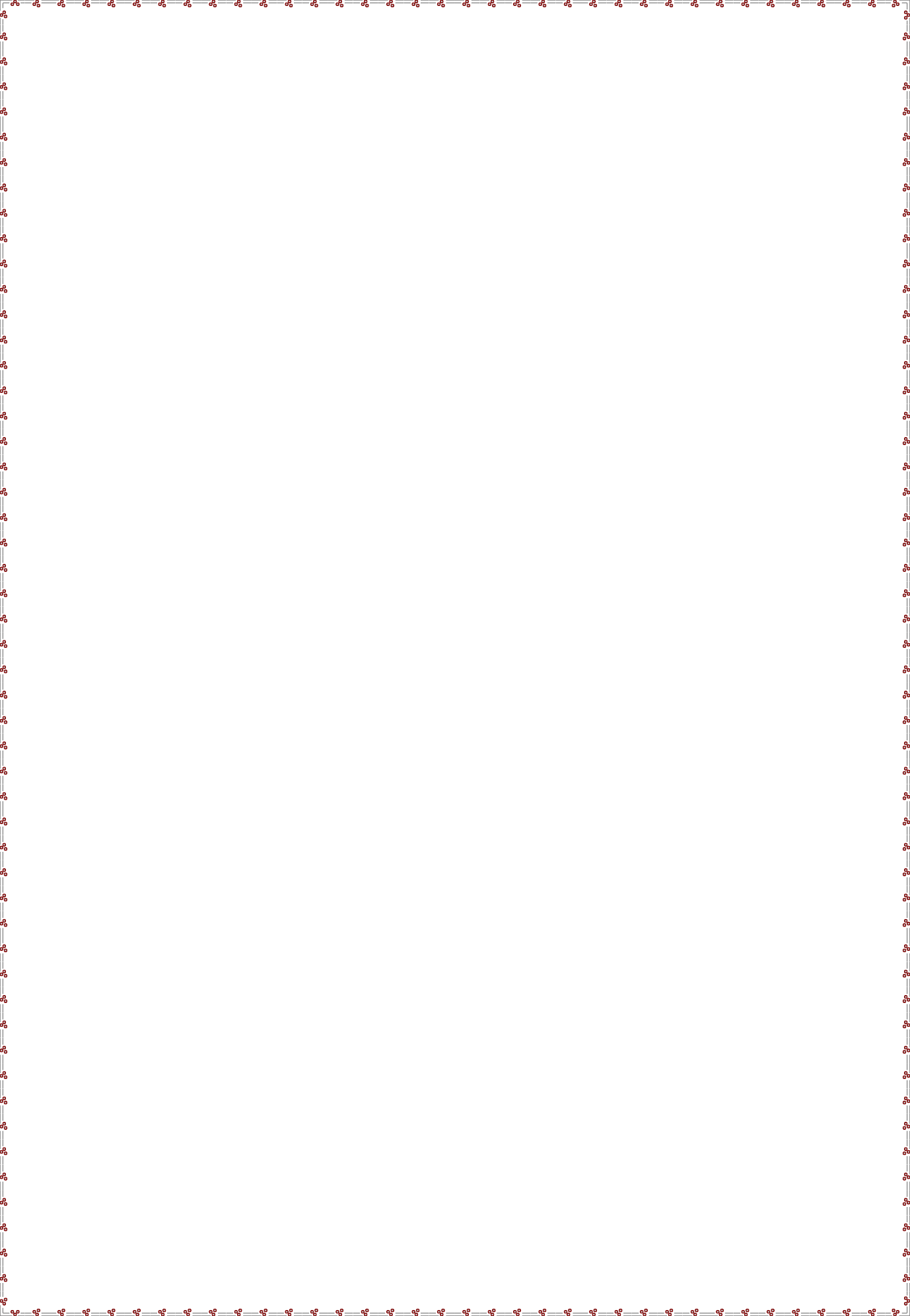
struct LinkedList\* allMessages = readMessagesFromFile(filename);

// Step 2: Filter out spam messages

struct LinkedList\* filteredMessages = filterSpam(allMessages);

// Step 3: Display the filtered messages

displayMessages(filteredMessages);

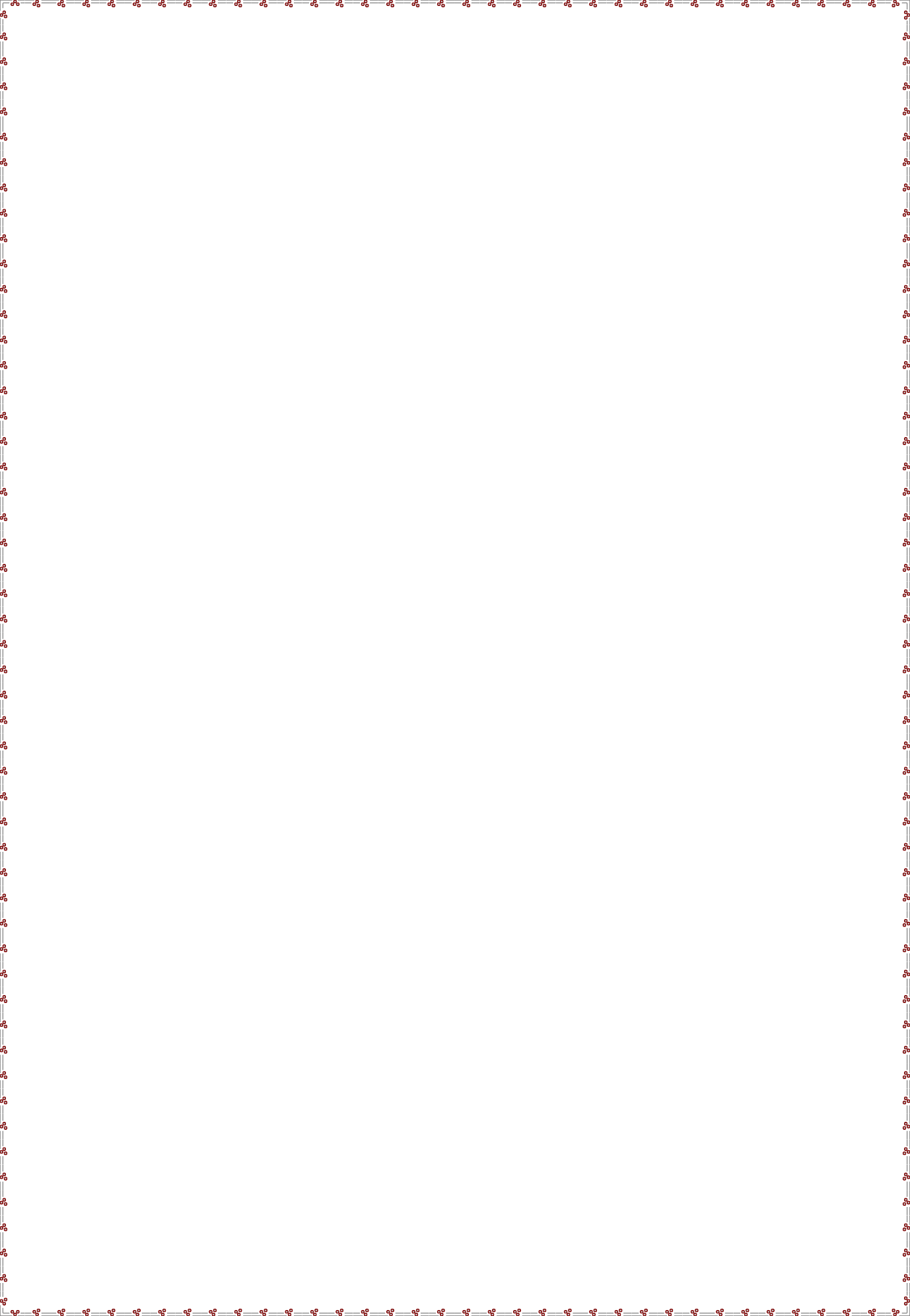
 // Step 4: Clean up memory

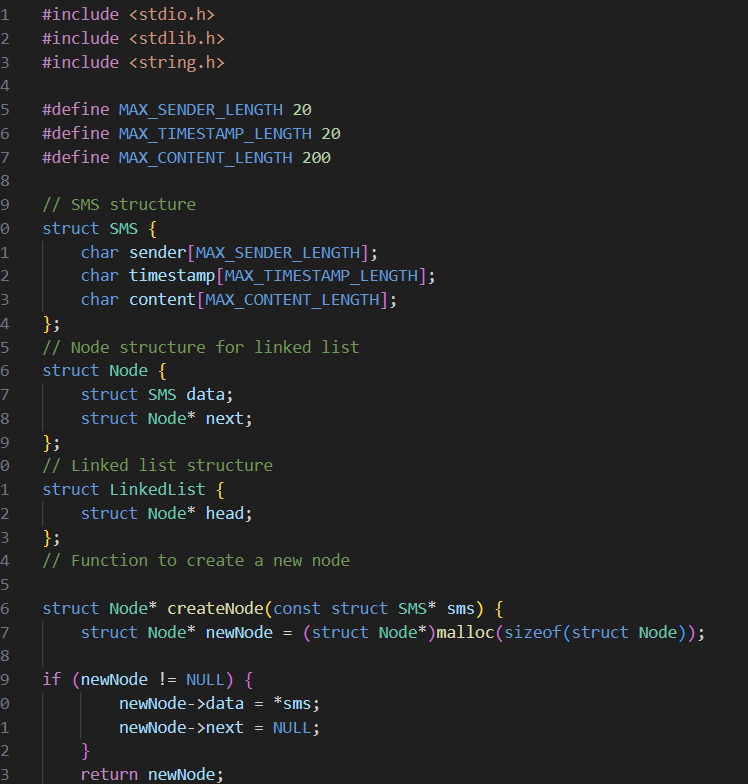
freeLinkedList(allMessages);

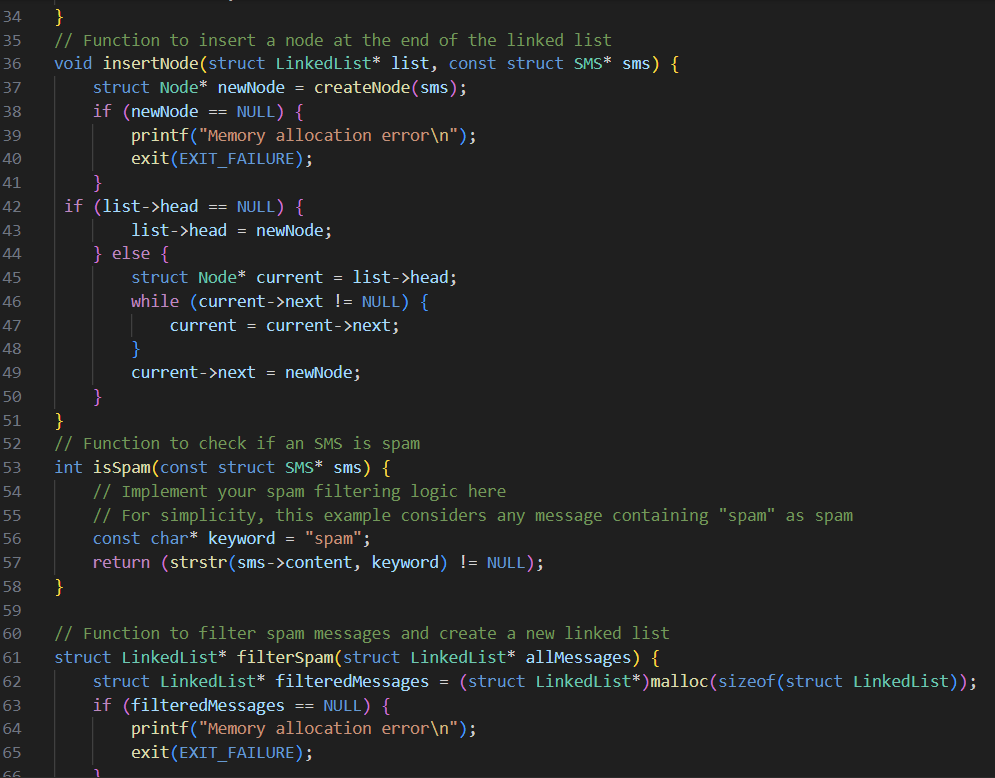
freeLinkedList(filteredMessages);

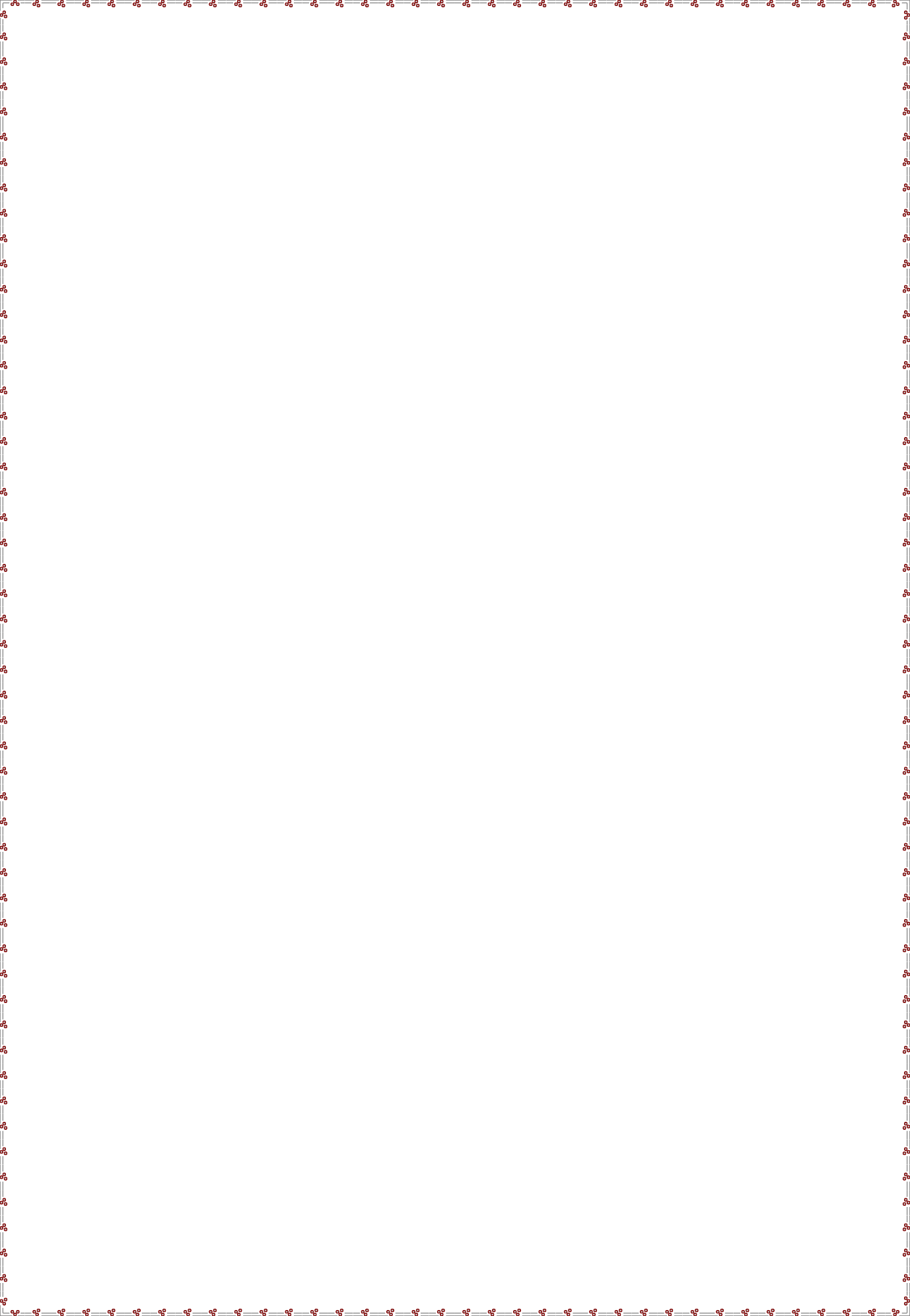
return 0;

}

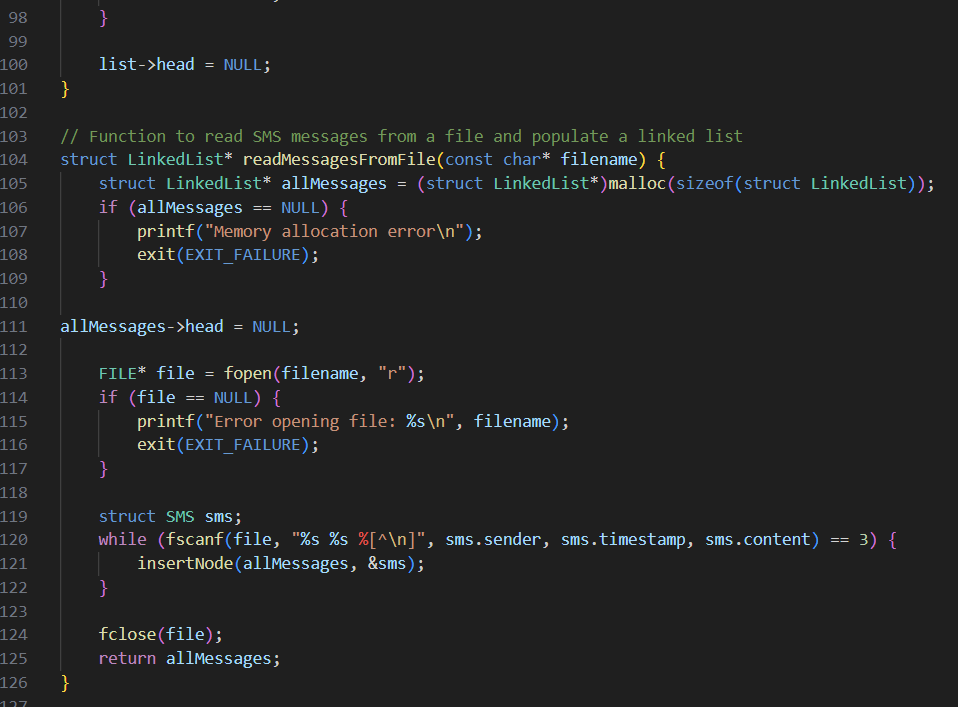


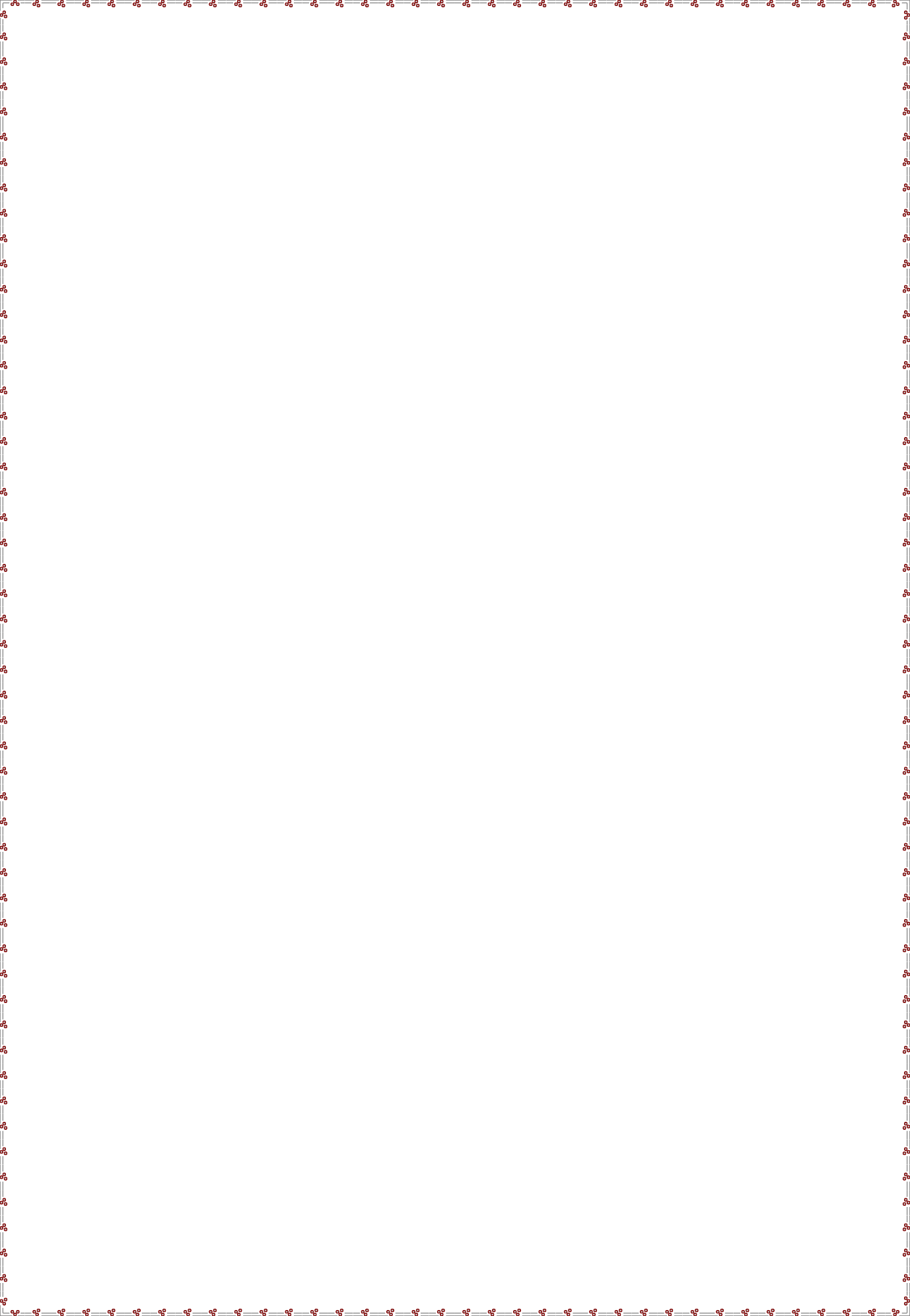
****

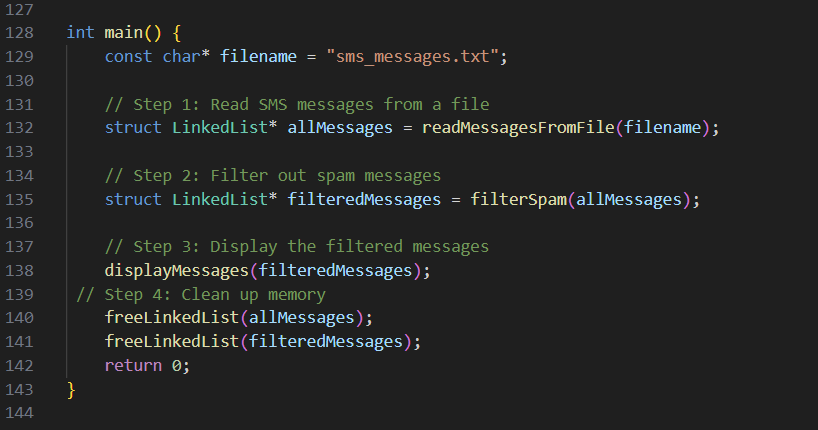
****



****



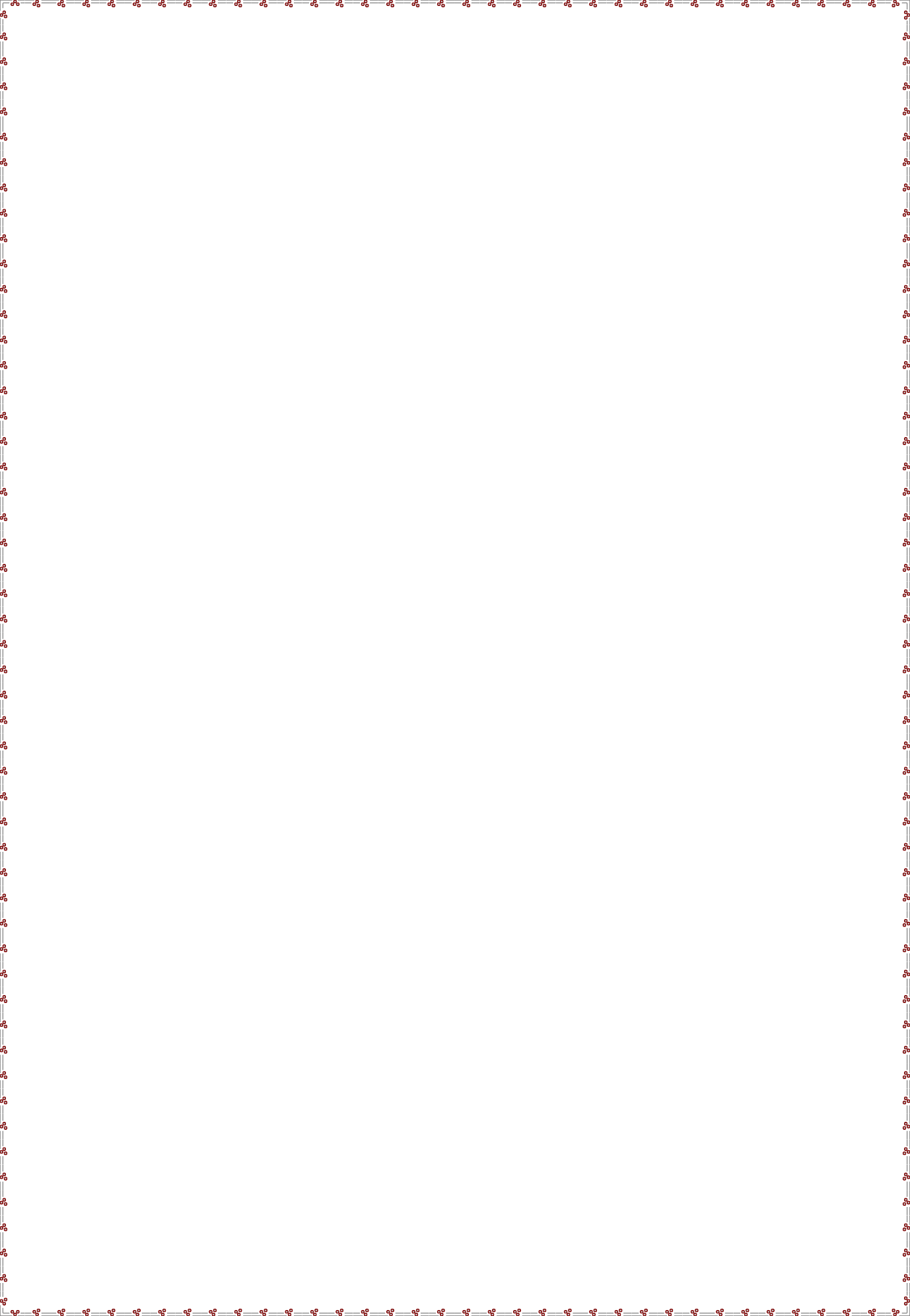


****

**NOTE :-**

The program will read SMS messages from the "sms\_messages.txt" file, filter out spam messages, and display the filtered messages on the console.

Make sure the "sms\_messages.txt" file is present in the same directory as your C file

**OutPut:**

Sender: John

Timestamp: 2023-01-01 12:30:45

Content: Hello, how are you?

Sender: Alice

Timestamp: 2023-01-02 09:15:20

Content: Urgent: Claim your prize now!

Sender: Bob

Timestamp: 2023-01-03 15:45:00

Content: Meeting at 4 PM in the conference room.

...

Filtered Messages:

Sender: John

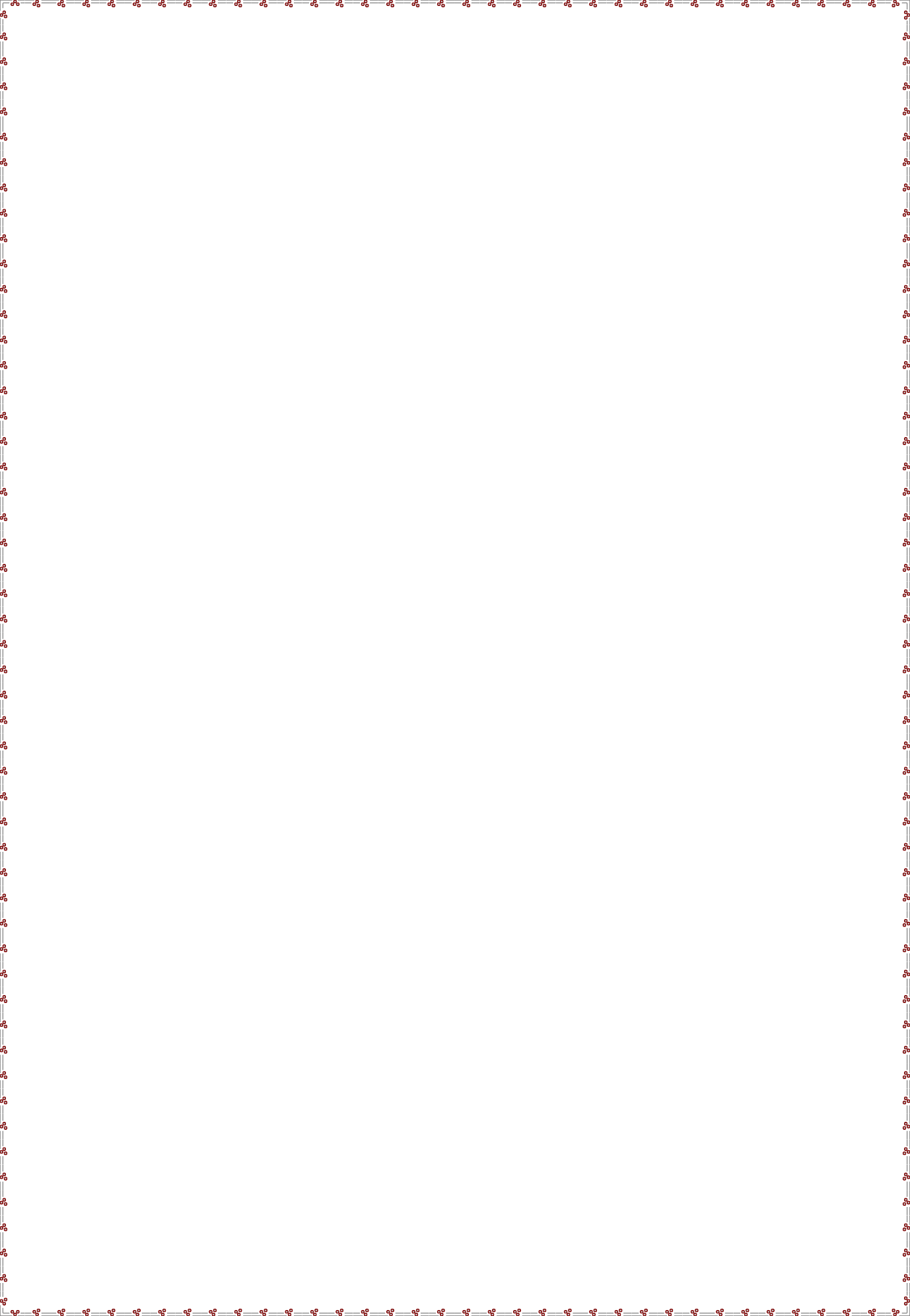
Timestamp: 2023-01-01 12:30:45

Content: Hello, how are you?

Sender: Bob

Timestamp: 2023-01-03 15:45:00

Content: Meeting at 4 PM in the conference room.

**Conclusion:**

In conclusion, The number of people using mobile devices increasing day by day. SMS (short message service) is a text message service available in smartphones as well as basic phones. So, the traffic of SMS increased drastically. The spam messages also increased. The hackers try to send spam messages for their financial or business benefits like market growth, lottery ticket information, credit card information, etc. So, spam classification has special attention. In this paper, we applied various machine learning and deep learning techniques for SMS spam detection. we used a dataset to train the machine learning and deep learning models like LSTM and NB. The SMS spam collection data set is used for testing the method. The dataset is split into two categories for training and testing the research. Our experimental results have shown that our NB model outperforms previous models in spam detection with an accuracy of good. This improves security of the mobile.

**References :**

<https://www.studocu.com/in/document/shri-mata-vaishno-devi-university/high-performance-computing-cse-3223/spam-message-detection-report/41537619>

<https://github.com/topics/spam-detection?l=c%2B%2B>