AURORA DEEMED TO BE UNIVERSITY
AURORA HIGHER EDUCATION AND RESEARCH ACADEMY

TASK 1

CHATBOT WITH RULE-BASED RESPONSES

Build a simple chatbot that responds to user inputs based on predefined rules. Use if-else statements or pattern matching techniques to identify user queries and provide appropriate responses. This will give you a basic understanding of natural language processing and conversation flow.

By
K.Charmi

CONTENT PAGE

INTRODUCTION

A Simple Chatbot is essentially a foundational interactive program designed to simulate dialogue between a user and a computer via predetermined responses. Developed using the C programming language, this project illustrates key programming principles, including the use of conditional statements, string manipulation, and functions. The chatbot offers a straightforward, text-based interface that enables users to input queries and receive corresponding replies, typically determined through keyword detection or basic rule-based logic. This serves as a valuable introductory exercise for understanding user input processing, control flow, and fundamental concepts in artificial intelligence.

ABSTRACT

Project Overview:

The chatbot interacts with users by responding to predefined inputs with relevant answers.
•The chatbot uses conditional logic if-else to handle common inputs like greetings, questions, and basic commands.
•It can perform simple tasks like answering questions, Your details, and evaluating math expressions like addition and subtraction.
•The chatbot ensures smooth user experience by recognizing common phrases and providing meaningful responses or fallback messages for unrecognized inputs.
•The conversation continues in a loop until the user types a command to exit (e.g., “exit”, “quit”, or “bye”).
•Allows users to restart the interaction without rerunning the program, maintaining a seamless chat session in the console.

Programming Concepts Used:

String Handling – To receive, compare, and interpret user inputs.
Conditionals – To implement rule-based responses and guide conversation flow.
Functions – To organize conversation logic into reusable modules for clarity and maintainability.
Input Validation – To manage unexpected or invalid inputs and maintain conversation stability.

This project focuses on strengthening core C programming skills through string processing, logic design, and modular function implementation. It enhances problem-solving abilities by simulating real-world interaction and reinforces structured, readable code practices.
The goal is to develop a simple, rule-based chatbot that is interactive, user-friendly, and serves as a stepping stone to more advanced natural language applications.

OBJECTIVES

The primary objectives of this project are:

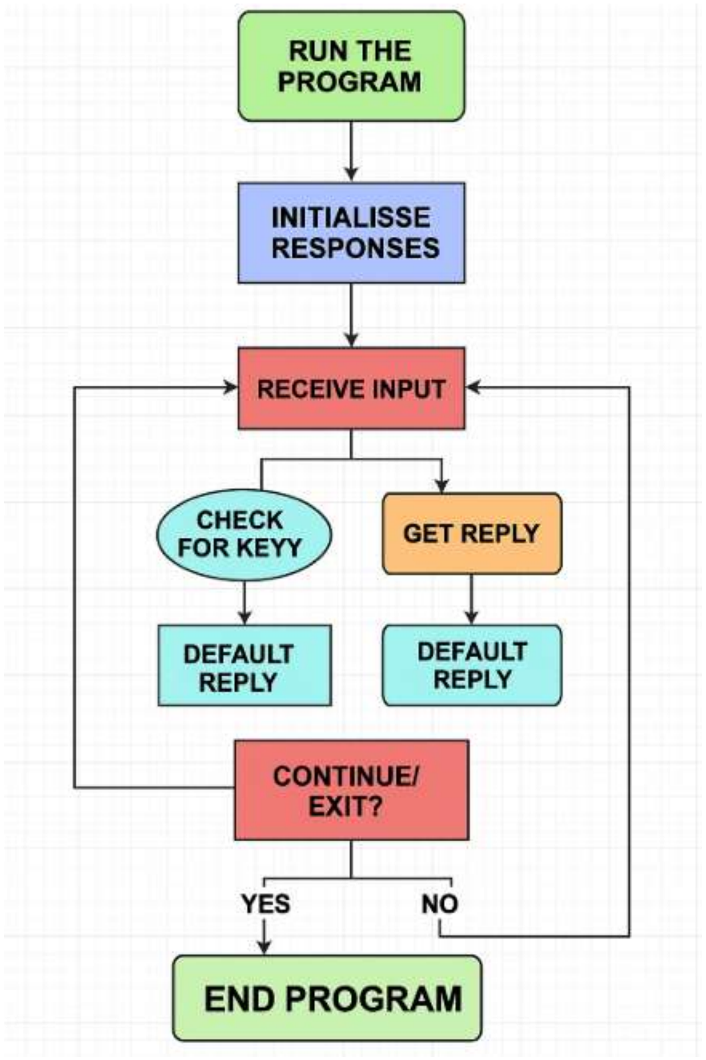
•To develop an interactive and user-friendly console-based chatbot that can respond to basic user inputs using predefined rules.

•To boost fundamental C programming concepts, such as loops, conditional statements, string handling, and functions.

•To effectively handle user input and provide appropriate, logical, and timely responses, simulating natural conversation flow.

•To implement rule-based logic for recognizing keywords, responding to greetings, performing simple tasks like calculations, and managing unknown queries gracefully.

LOGIC


RUN THE PROGRAM
INITIALISSE RESPONSES
RECEIVE INPUT
CHECK FOR KEYY
GET REPLY
DEFAULT REPLY
DEFAULT REPLY
CONTINUE/ EXIT?
YES
NO
END PROGRAM

METHODOLOGY USED IN THE PROJECT

Algorithm Design:-

•Initialize a loop for continuous interaction.

•Take user input and normalize it (lowercase, trimmed).

•Match input with predefined responses using conditions.

•Output chatbot's response based on matching rule.

•Handle unknown queries with a default reply.

•Allow the user to type 'exit', 'quit', or 'bye' to stop the chatbot.

•Optionally include basic features like a calculator or time display.

Implementation in C (for chatbot)

•Use string variables and functions to process user input.
•Create a main interaction loop with if-else or switch-case .
•Define functions for:
❖ Calculator functionality.
❖ Providing responses for greeting, thanks, and general queries.

Input Handling & Validation

•Ensure input is captured correctly using input()
•Handle empty or unsupported queries with a generic fallback message.
•Ensure expressions in the calculator are evaluated safely

Chatbot Logic Implementation

•Use string matching to detect keywords like “hello”, “how are you”, “calculator”.
•Provide fixed responses based on matched phrases.
•Use functions to modularize logic (e.g., handle_calculator()).
•Exit the loop gracefully when user types "exit", "bye", etc.

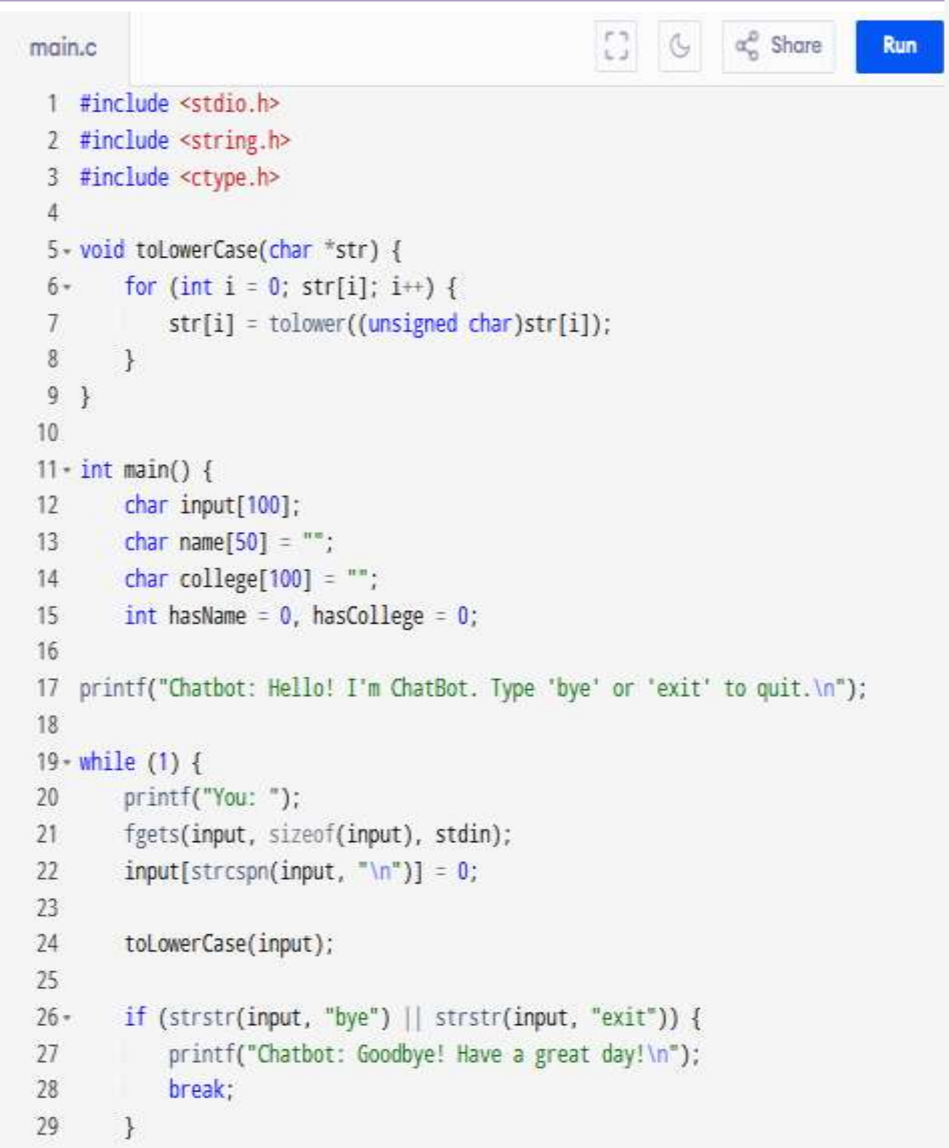
Testing and Debugging

•Test responses for each rule to ensure correct output.
•Input a variety of user queries (known and unknown) to check default handling.
•Validate calculator expressions with valid and invalid inputs.
•Test exit conditions and overall loop control.

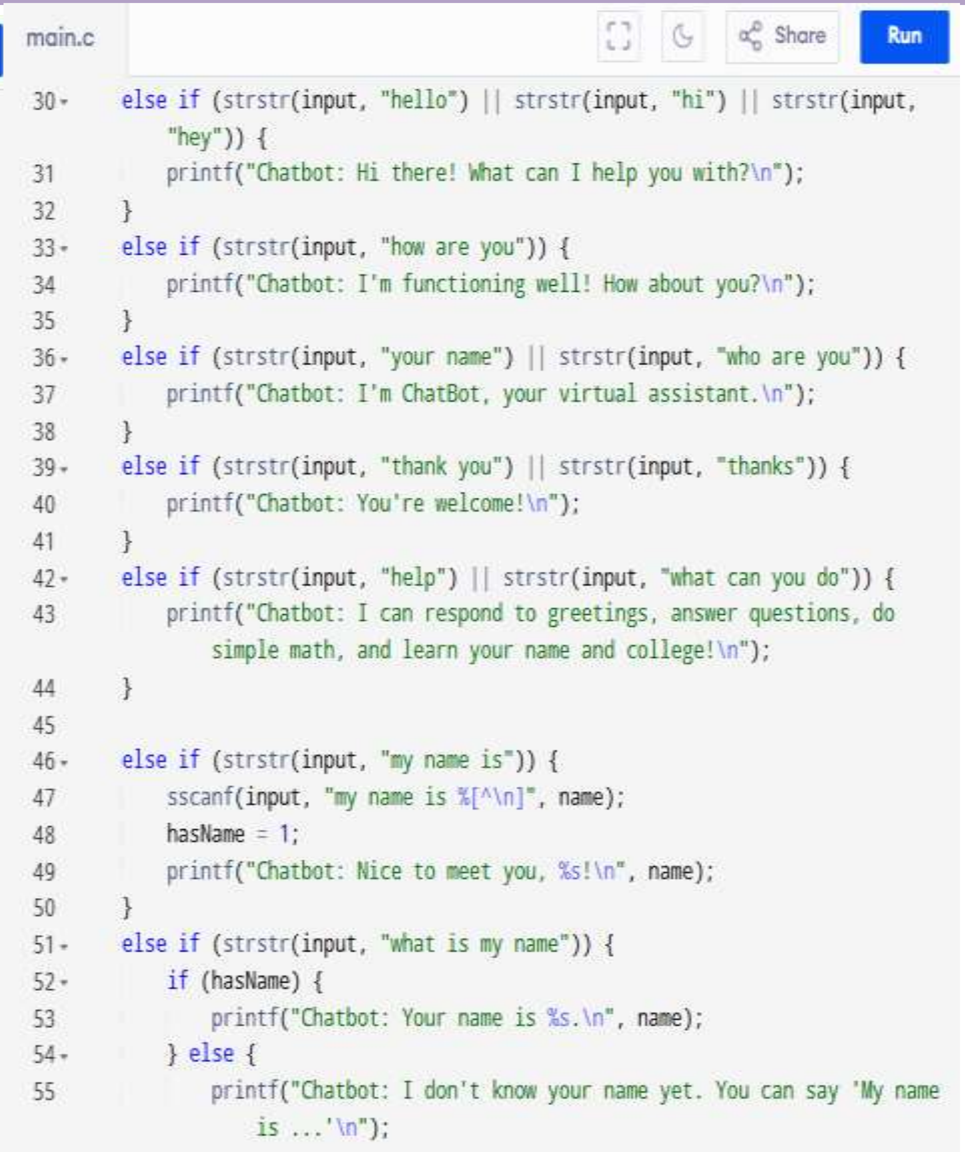
SOFTWARE REQUIREMENTS

1. **Operating System**
   - Windows
2. **Programming Language**
   - C Language
3. **Compiler**
   - Programiz (Online Compiler)
4. **System Requirements**
   - Processor: Any modern CPU (Intel, AMD, ARM)
   - RAM: At least 512MB
   - Storage: Less than 10MB

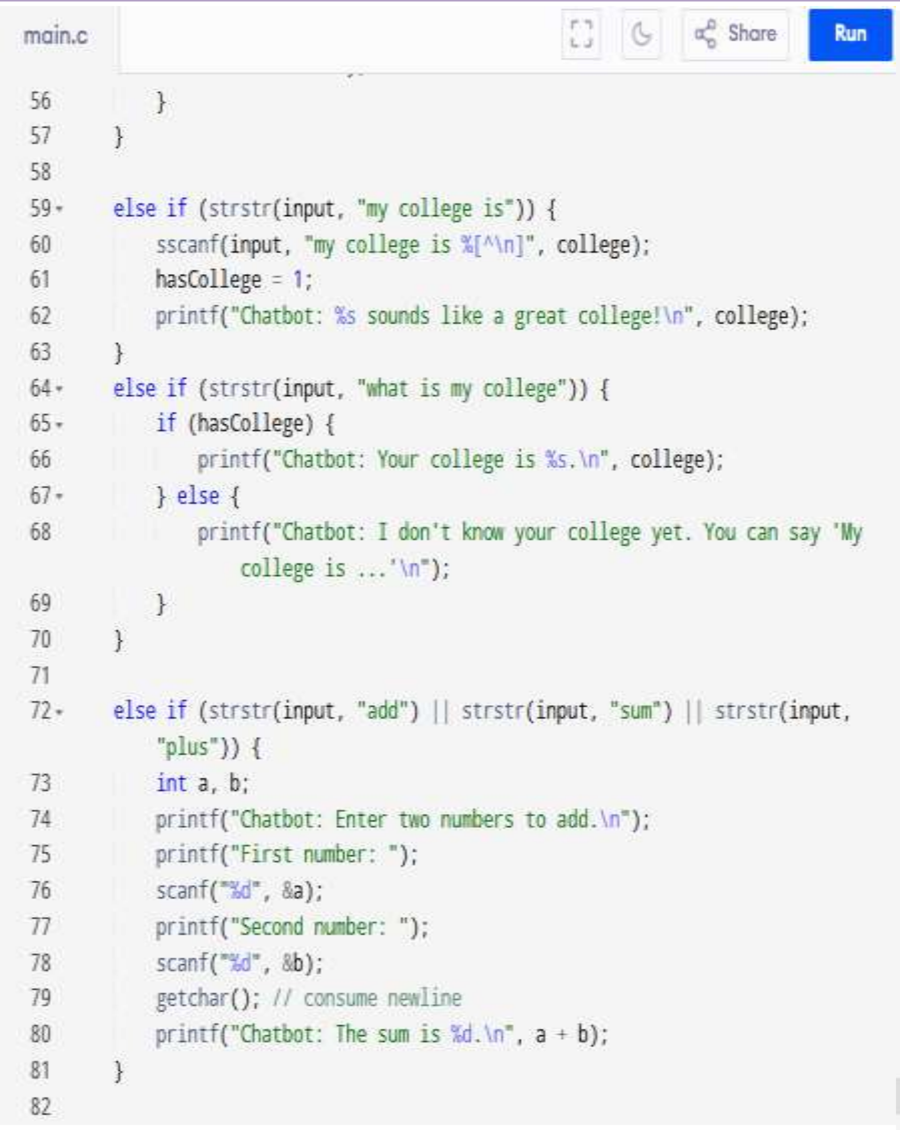
EXECUTION SCREENSHOTS



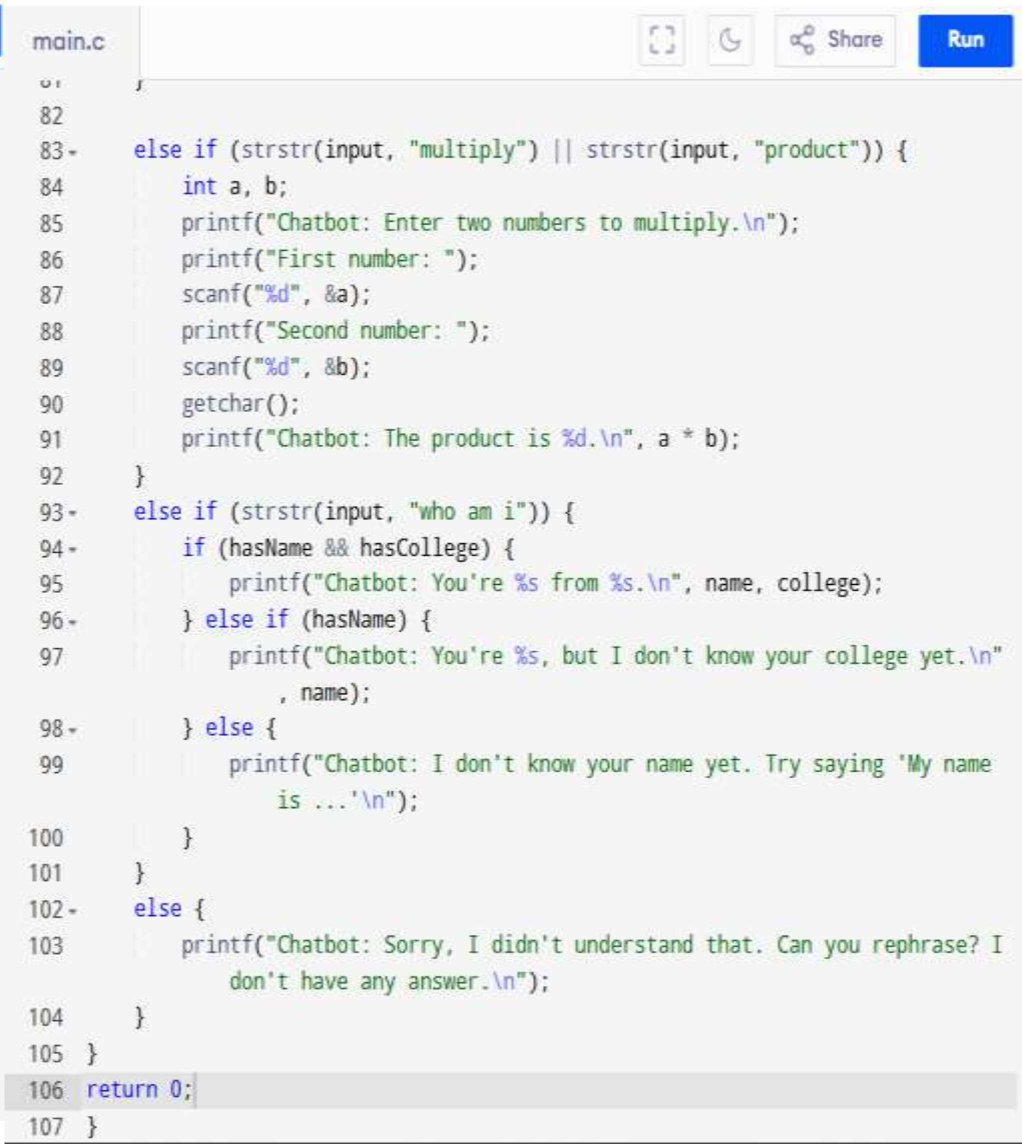
```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>

void toLowerCase(char *str) {
    for (int i = 0; str[i]; i++) {
        str[i] = tolower((unsigned char)str[i]);
    }
}

int main() {
    char input[100];
    char name[50] = "";
    char college[100] = "";
    int hasName = 0, hasCollege = 0;

printf("Chatbot: Hello! I'm ChatBot. Type 'bye' or 'exit' to quit.\n");

while (1) {
    printf("You: ");
    fgets(input, sizeof(input), stdin);
    input[strcspn(input, "\n")] = 0;

    toLowerCase(input);

    if (strstr(input, "bye") || strstr(input, "exit")) {
        printf("Chatbot: Goodbye! Have a great day!\n");
        break;
    }
```



```c
    else if (strstr(input, "hello") || strstr(input, "hi") || strstr(input,
        "hey")) {
        printf("Chatbot: Hi there! What can I help you with?\n");
    }
    else if (strstr(input, "how are you")) {
        printf("Chatbot: I'm functioning well! How about you?\n");
    }
    else if (strstr(input, "your name") || strstr(input, "who are you")) {
        printf("Chatbot: I'm ChatBot, your virtual assistant.\n");
    }
    else if (strstr(input, "thank you") || strstr(input, "thanks")) {
        printf("Chatbot: You're welcome!\n");
    }
    else if (strstr(input, "help") || strstr(input, "what can you do")) {
        printf("Chatbot: I can respond to greetings, answer questions, do
            simple math, and learn your name and college!\n");
    }

    else if (strstr(input, "my name is")) {
        sscanf(input, "my name is %[^\n]", name);
        hasName = 1;
        printf("Chatbot: Nice to meet you, %s!\n", name);
    }
    else if (strstr(input, "what is my name")) {
        if (hasName) {
            printf("Chatbot: Your name is %s.\n", name);
        } else {
            printf("Chatbot: I don't know your name yet. You can say 'My name
                is ...'\n");
```

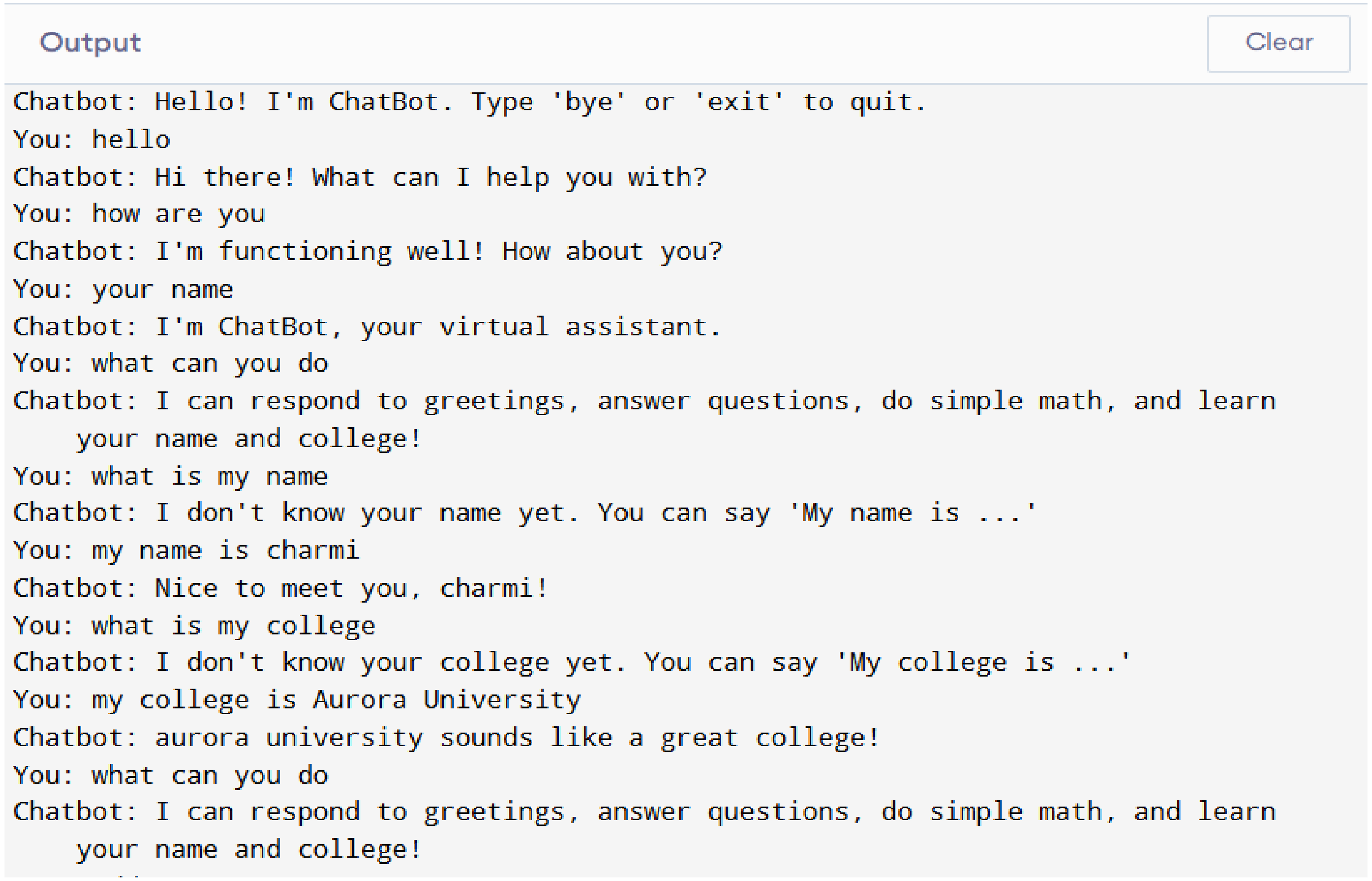


```c
        }
    }

    else if (strstr(input, "my college is")) {
        sscanf(input, "my college is %[^\n]", college);
        hasCollege = 1;
        printf("Chatbot: %s sounds like a great college!\n", college);
    }
    else if (strstr(input, "what is my college")) {
        if (hasCollege) {
            printf("Chatbot: Your college is %s.\n", college);
        } else {
            printf("Chatbot: I don't know your college yet. You can say 'My
                college is ...'\n");
        }
    }

    else if (strstr(input, "add") || strstr(input, "sum") || strstr(input,
        "plus")) {
        int a, b;
        printf("Chatbot: Enter two numbers to add.\n");
        printf("First number: ");
        scanf("%d", &a);
        printf("Second number: ");
        scanf("%d", &b);
        getchar(); // consume newline
        printf("Chatbot: The sum is %d.\n", a + b);
    }

```

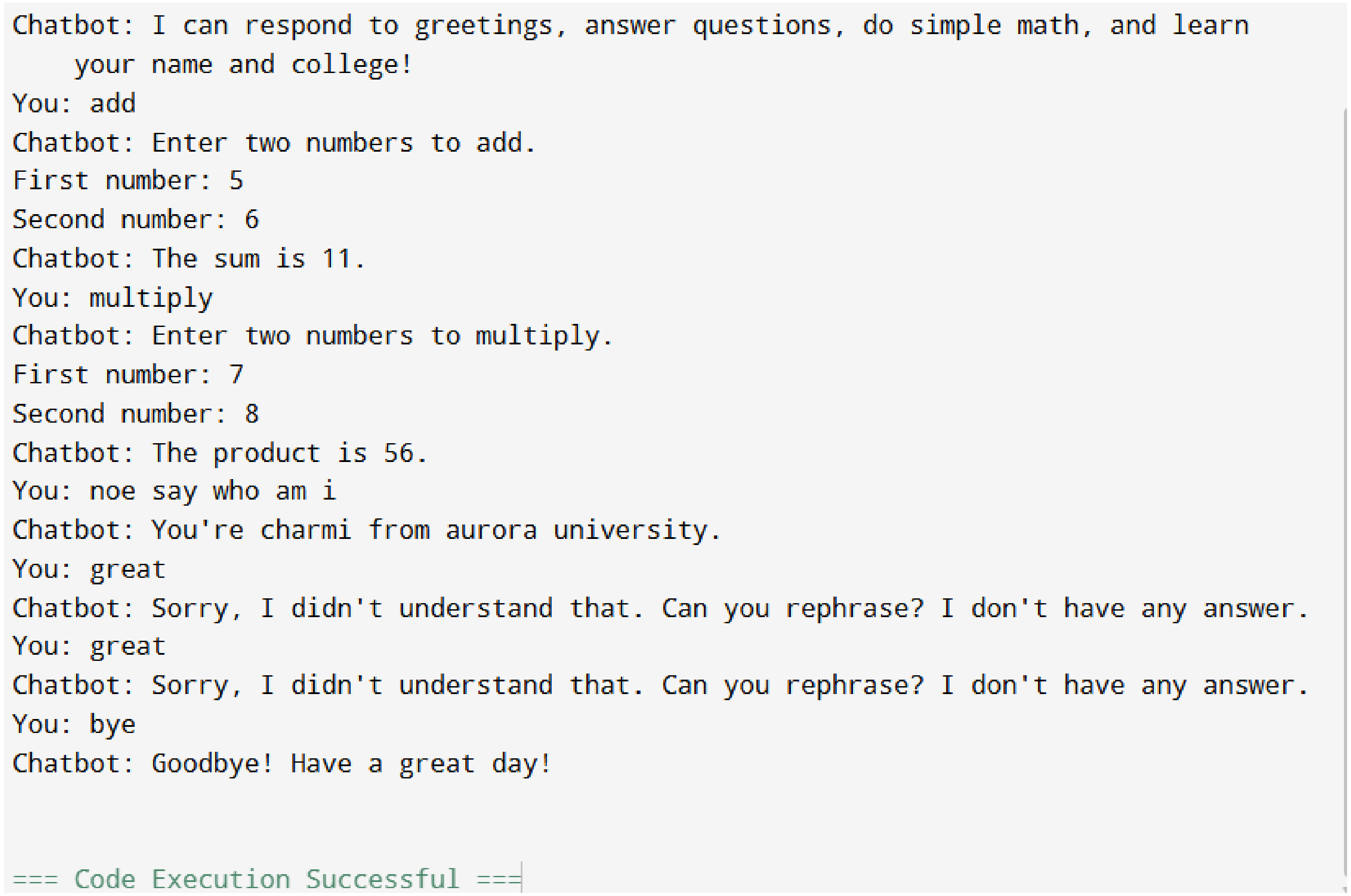


```c

    else if (strstr(input, "multiply") || strstr(input, "product")) {
        int a, b;
        printf("Chatbot: Enter two numbers to multiply.\n");
        printf("First number: ");
        scanf("%d", &a);
        printf("Second number: ");
        scanf("%d", &b);
        getchar();
        printf("Chatbot: The product is %d.\n", a * b);
    }
    else if (strstr(input, "who am i")) {
        if (hasName && hasCollege) {
            printf("Chatbot: You're %s from %s.\n", name, college);
        } else if (hasName) {
            printf("Chatbot: You're %s, but I don't know your college yet.\n"
                , name);
        } else {
            printf("Chatbot: I don't know your name yet. Try saying 'My name
                is ...'\n");
        }
    }
    else {
        printf("Chatbot: Sorry, I didn't understand that. Can you rephrase? I
            don't have any answer.\n");
    }
}
return 0;
}
```

Output / Results



```
Chatbot: Hello! I'm ChatBot. Type 'bye' or 'exit' to quit.
You: hello
Chatbot: Hi there! What can I help you with?
You: how are you
Chatbot: I'm functioning well! How about you?
You: your name
Chatbot: I'm ChatBot, your virtual assistant.
You: what can you do
Chatbot: I can respond to greetings, answer questions, do simple math, and learn
    your name and college!
You: what is my name
Chatbot: I don't know your name yet. You can say 'My name is ...'
You: my name is charmi
Chatbot: Nice to meet you, charmi!
You: what is my college
Chatbot: I don't know your college yet. You can say 'My college is ...'
You: my college is Aurora University
Chatbot: aurora university sounds like a great college!
You: what can you do
Chatbot: I can respond to greetings, answer questions, do simple math, and learn
    your name and college!
```



```
Chatbot: I can respond to greetings, answer questions, do simple math, and learn
    your name and college!
You: add
Chatbot: Enter two numbers to add.
First number: 5
Second number: 6
Chatbot: The sum is 11.
You: multiply
Chatbot: Enter two numbers to multiply.
First number: 7
Second number: 8
Chatbot: The product is 56.
You: noe say who am i
Chatbot: You're charmi from aurora university.
You: great
Chatbot: Sorry, I didn't understand that. Can you rephrase? I don't have any answer.
You: great
Chatbot: Sorry, I didn't understand that. Can you rephrase? I don't have any answer.
You: bye
Chatbot: Goodbye! Have a great day!


=== Code Execution Successful ===
```

Optimization and Enhancements

1. Improve user experience with emojis, prompts, and spacing.
2. Add support for more commands (e.g., jokes, name interaction).
3. Modularize the code using functions to improve readability and reuse.
4. Plan future upgrades like enable the chatbot to engage in more complex, multi-step conversations.
5. To recognize and respond to user emotions, empathy, or sentiment.
6. To allow users to rate or provide feedback on chatbot responses

CONCLUSION

•This C-based chatbot project successfully implements a simple, rule-based conversational agent using fundamental programming concepts like strings, conditionals, loops, and functions.

•The project effectively demonstrates the logic behind text input handling, response matching, and conversation flow control. It simulates a human-like chat interface where the bot responds to specific user queries with predefined answers.

•This project serves as an excellent learning tool for beginners in C programming, showcasing how basic control structures can be used to model real-world applications. It can be further enhanced by integrating pattern matching, natural language processing (NLP) techniques for a more advanced user experience.

THANK
YOU