Project on the basis of C programming

Here are some project ideas in C, including their objectives, techniques used, and brief descriptions along with their solutions.

1. Library Management System

- **Objective**: To develop a system that can manage library resources such as books, magazines, and journals. It will allow users to issue, return, and search for books, and provide details about borrowed items.
- **Techniques**: File Handling, Structures, Functions
- **Description**: The system will store the book information in a file, allowing users to add, issue, and return books. It will also keep track of the due dates and generate reports for library management.
- **Solution**: Below is a basic structure of the library management system with simplified functionality. For detailed functionality, the system could include user authentication and a database backend.

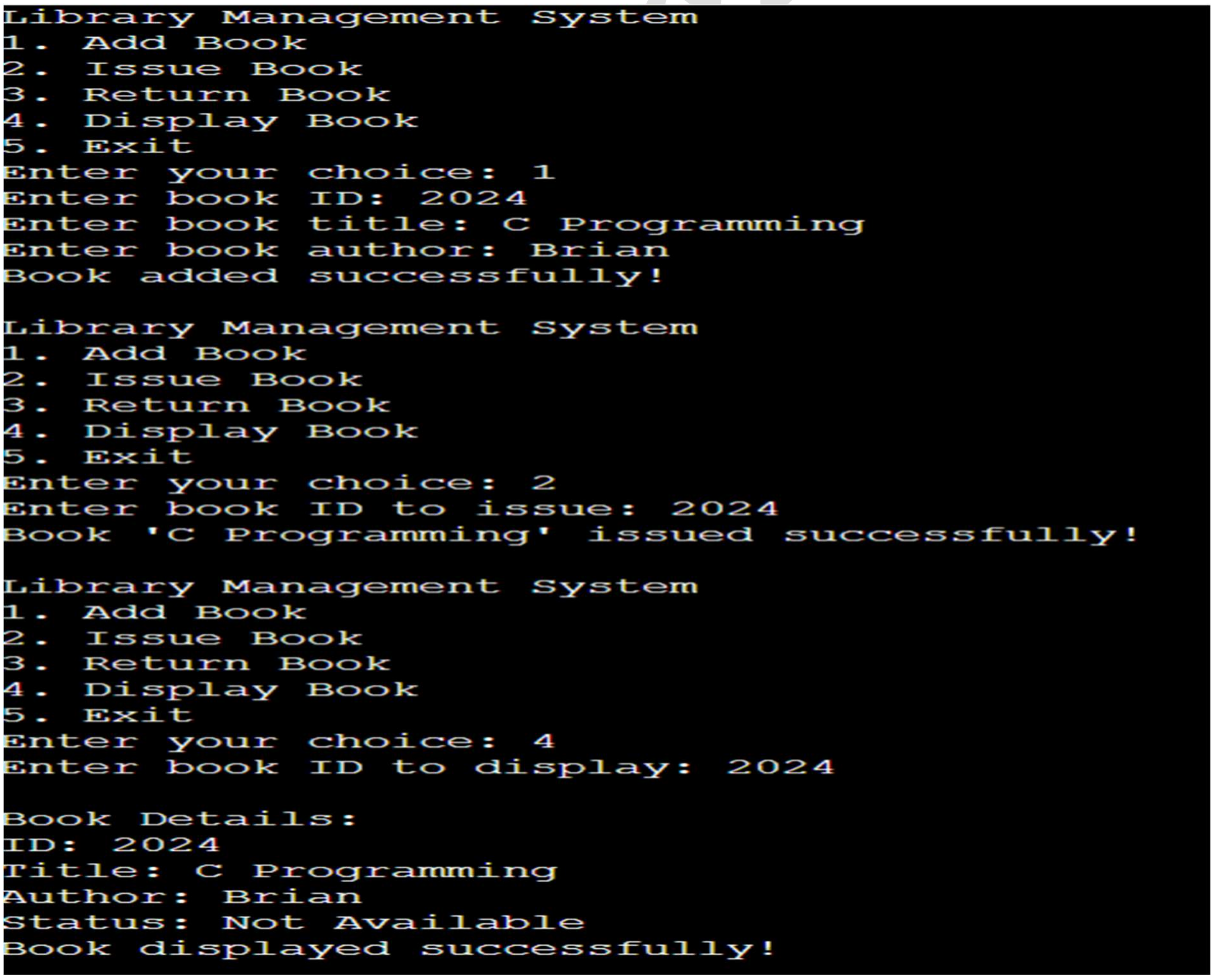
Source Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX_BOOKS 20  // Define a maximum number of books that can be added


struct Book {

   int id;

   char title[100];

   char author[100];

   int available;

};


void addBook(struct Book *book) {

   printf("Enter book ID: ");

   scanf("%d", &book->id);

   getchar(); // Consume newline left by scanf
```

```
    printf("Enter book title: ");
    fgets(book->title, sizeof(book->title), stdin);
    book->title[strcspn(book->title, "\n")] = '\0'; // Remove newline character

    printf("Enter book author: ");
    fgets(book->author, sizeof(book->author), stdin);
    book->author[strcspn(book->author, "\n")] = '\0'; // Remove newline character

    book->available = 1;
    printf("Book added successfully!\n");
}

void issueBook(struct Book *book) {
    if (book->available == 1) {
        book->available = 0;
        printf("Book '%s' issued successfully!\n", book->title);
    } else {
        printf("Sorry, the book '%s' is currently unavailable.\n", book->title);
    }
}

void returnBook(struct Book *book) {
    book->available = 1;
    printf("Book '%s' returned successfully!\n", book->title);
}

void displayBook(struct Book book) {
    printf("\nBook Details:\n");
    printf("ID: %d\n", book.id);
    printf("Title: %s\n", book.title);
    printf("Author: %s\n", book.author);
    printf("Status: %s\n", book.available ? "Available" : "Not Available");
    printf("Book displayed successfully!\n");
}

void clearInputBuffer() {
    while (getchar() != '\n');  // Clear the input buffer
}

int main() {
    struct Book books[MAX_BOOKS];  // Array of books
    int bookCount = 0;            // Keep track of how many books are added
    int choice;

    while (1) {
        printf("\nLibrary Management System\n");
        printf("1. Add Book\n");
        printf("2. Issue Book\n");
        printf("3. Return Book\n");
        printf("4. Display Book\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");

        if (scanf("%d", &choice) != 1) {
            clearInputBuffer();  // Clear the buffer if invalid input
            printf("Invalid choice! Please enter a number between 1 and 5.\n");
            continue;
        }

        switch (choice) {
            case 1:
                if (bookCount < MAX_BOOKS) {
```

```
            addBook(&books[bookCount]);
            bookCount++;
        } else {
            printf("Cannot add more books, the library is full.\n");
        }
        break;
    case 2:
        if (bookCount > 0) {
            printf("Enter book ID to issue: ");
            int id;
            scanf("%d", &id);
            int found = 0;
            for (int i = 0; i < bookCount; i++) {
                if (books[i].id == id) {
                    issueBook(&books[i]);
                    found = 1;
                    break;
                }
            }
            if (!found) {
                printf("Book with ID %d not found.\n", id);
            }
        } else {
            printf("No books available in the library.\n");
        }
        break;
    case 3:
        if (bookCount > 0) {
            printf("Enter book ID to return: ");
            int id;
            scanf("%d", &id);
            int found = 0;
            for (int i = 0; i < bookCount; i++) {
                if (books[i].id == id) {
                    returnBook(&books[i]);
                    found = 1;
                    break;
                }
            }
            if (!found) {
                printf("Book with ID %d not found.\n", id);
            }
        } else {
            printf("No books available in the library.\n");
        }
        break;
    case 4:
        if (bookCount > 0) {
            printf("Enter book ID to display: ");
            int id;
            scanf("%d", &id);
            int found = 0;
            for (int i = 0; i < bookCount; i++) {
                if (books[i].id == id) {
                    displayBook(books[i]);
                    found = 1;
                    break;
                }
            }
            if (!found) {
                printf("Book with ID %d not found.\n", id);
            }
            } else {
                printf("No books available in the library.\n");
            }
            break;
        case 5:
            printf("Exiting... Library management work completed successfully!\n");
            exit(0);
        default:
            printf("Invalid choice! Try again.\n");
        }
    }
    return 0;
}
```

Output:



```
Library Management System
1. Add Book
2. Issue Book
3. Return Book
4. Display Book
5. Exit
Enter your choice: 1
Enter book ID: 2024
Enter book title: C Programming
Enter book author: Brian
Book added successfully!

Library Management System
1. Add Book
2. Issue Book
3. Return Book
4. Display Book
5. Exit
Enter your choice: 2
Enter book ID to issue: 2024
Book 'C Programming' issued successfully!

Library Management System
1. Add Book
2. Issue Book
3. Return Book
4. Display Book
5. Exit
Enter your choice: 4
Enter book ID to display: 2024

Book Details:
ID: 2024
Title: C Programming
Author: Brian
Status: Not Available
Book displayed successfully!
```

2. Student Management System

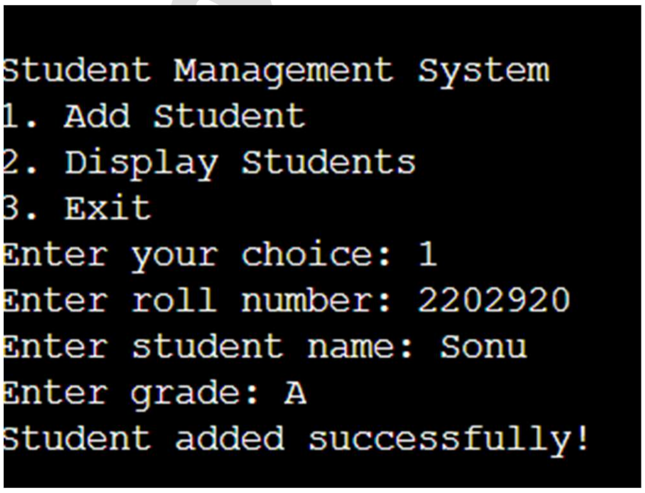
- **Objective**: To manage the information of students such as personal details, grades, and courses they are enrolled in.
- **Techniques**: File Handling, Structures, Arrays
- **Description**: The project involves storing student details like name, roll number, grades, and courses in a file. The system will allow adding, updating, deleting, and viewing student records.
- **Solution**: Here is a simple implementation of adding and viewing student records.

Source Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


struct Student {

   int roll_no;

   char name[100];

   float grade;

};


void addStudent() {

   struct Student student;

   FILE *file = fopen("students.dat", "a");


   if (!file) {

      printf("Error opening file!\n");

      return;

   }


   printf("Enter roll number: ");

   scanf("%d", &student.roll_no);

   getchar(); // consume newline

   printf("Enter student name: ");
```

```
    fgets(student.name, sizeof(student.name), stdin);
    printf("Enter grade: ");
    scanf("%f", &student.grade);

    fwrite(&student, sizeof(struct Student), 1, file);
    fclose(file);
    printf("Student added successfully!\n");
}

void displayStudents() {
    struct Student student;
    FILE *file = fopen("students.dat", "r");

    if (!file) {
        printf("Error opening file!\n");
        return;
    }

    while (fread(&student, sizeof(struct Student), 1, file)) {
        printf("Roll No: %d\n", student.roll_no);
        printf("Name: %s", student.name);
        printf("Grade: %.2f\n\n", student.grade);
    }

    fclose(file);
}

int main() {
    int choice;
    while (1) {
        printf("\nStudent Management System\n");
        printf("1. Add Student\n");
        printf("2. Display Students\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
          case 1:
            addStudent();
            break;
          case 2:
            displayStudents();
            break;
          case 3:
            exit(0);
          default:
            printf("Invalid choice! Try again.\n");
        }
    }
    return 0;
}
```

Output:



```
Student Management System
1. Add Student
2. Display Students
3. Exit
Enter your choice: 1
Enter roll number: 2202920
Enter student name: Sonu
Enter grade: A
Student added successfully!
```

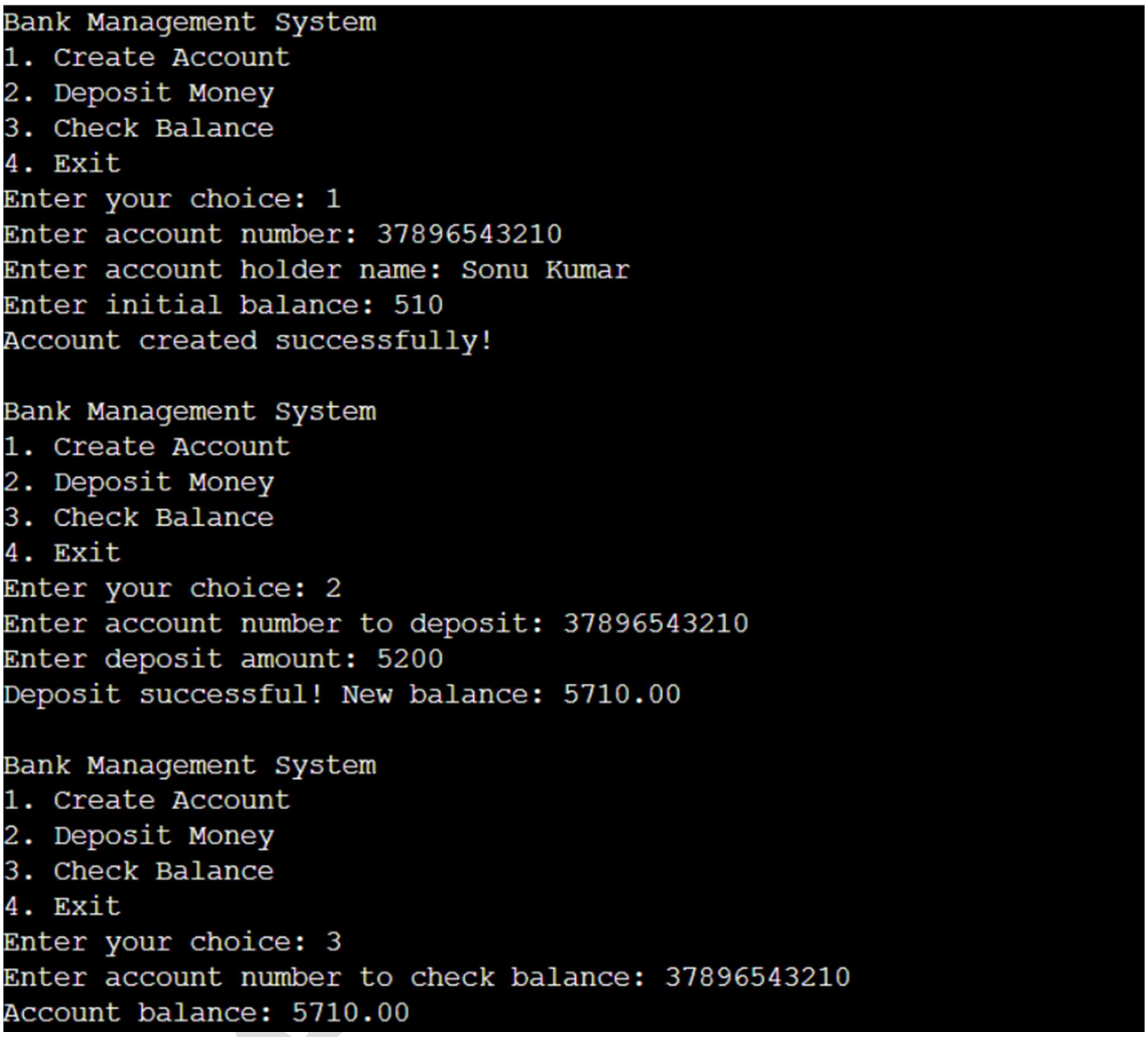
3. Simple Bank Management System

- **Objective**: To manage the accounts of customers, including operations like deposit, withdrawal, and checking account balance.
- **Techniques**: File Handling, Functions, Structures
- **Description**: The project will simulate basic bank operations. It will manage customer accounts by storing account details in a file and provide options to deposit, withdraw, and view the balance.
- **Solution**: Below is a basic implementation of deposit, withdraw, and check balance functionalities.

Source Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Account {
    int account_no;
    char name[100];
    float balance;
};

void createAccount() {
    struct Account acc;
    FILE *file = fopen("accounts.dat", "a");

    if (!file) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter account number: ");
    scanf("%d", &acc.account_no);
    getchar(); // consume newline
    printf("Enter account holder name: ");
    fgets(acc.name, sizeof(acc.name), stdin);
    printf("Enter initial balance: ");
    scanf("%f", &acc.balance);

    fwrite(&acc, sizeof(struct Account), 1, file);
    fclose(file);
    printf("Account created successfully!\n");
}

void deposit() {
    int acc_no;
    float amount;
    struct Account acc;
    FILE *file = fopen("accounts.dat", "r+");

    if (!file) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter account number to deposit: ");
    scanf("%d", &acc_no);
    printf("Enter deposit amount: ");
    scanf("%f", &amount);

    while (fread(&acc, sizeof(struct Account), 1, file)) {
        if (acc.account_no == acc_no) {
            acc.balance += amount;
            fseek(file, -sizeof(struct Account), SEEK_CUR);
            fwrite(&acc, sizeof(struct Account), 1, file);
            printf("Deposit successful! New balance: %.2f\n", acc.balance);
            fclose(file);
            return;
        }
    }

    printf("Account not found!\n");
    fclose(file);
}

void checkBalance() {
    int acc_no;
    struct Account acc;
    FILE *file = fopen("accounts.dat", "r");

    if (!file) {
        printf("Error opening file!\n");
        return;
    }

    printf("Enter account number to check balance: ");
    scanf("%d", &acc_no);

    while (fread(&acc, sizeof(struct Account), 1, file)) {
        if (acc.account_no == acc_no) {
            printf("Account balance: %.2f\n", acc.balance);
            fclose(file);
            return;
        }
    }
    printf("Account not found!\n");
    fclose(file);
}

int main() {
    int choice;
    while (1) {
        printf("\nBank Management System\n");
        printf("1. Create Account\n");
        printf("2. Deposit Money\n");
        printf("3. Check Balance\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                createAccount();
                break;
            case 2:
                deposit();
                break;
            case 3:
                checkBalance();
                break;
            case 4:
                exit(0);
            default:
                printf("Invalid choice! Try again.\n");
        }
    }
    return 0;

}
```

Output:



```
Bank Management System
1. Create Account
2. Deposit Money
3. Check Balance
4. Exit
Enter your choice: 1
Enter account number: 37896543210
Enter account holder name: Sonu Kumar
Enter initial balance: 510
Account created successfully!

Bank Management System
1. Create Account
2. Deposit Money
3. Check Balance
4. Exit
Enter your choice: 2
Enter account number to deposit: 37896543210
Enter deposit amount: 5200
Deposit successful! New balance: 5710.00

Bank Management System
1. Create Account
2. Deposit Money
3. Check Balance
4. Exit
Enter your choice: 3
Enter account number to check balance: 37896543210
Account balance: 5710.00
```

4. Online Quiz System

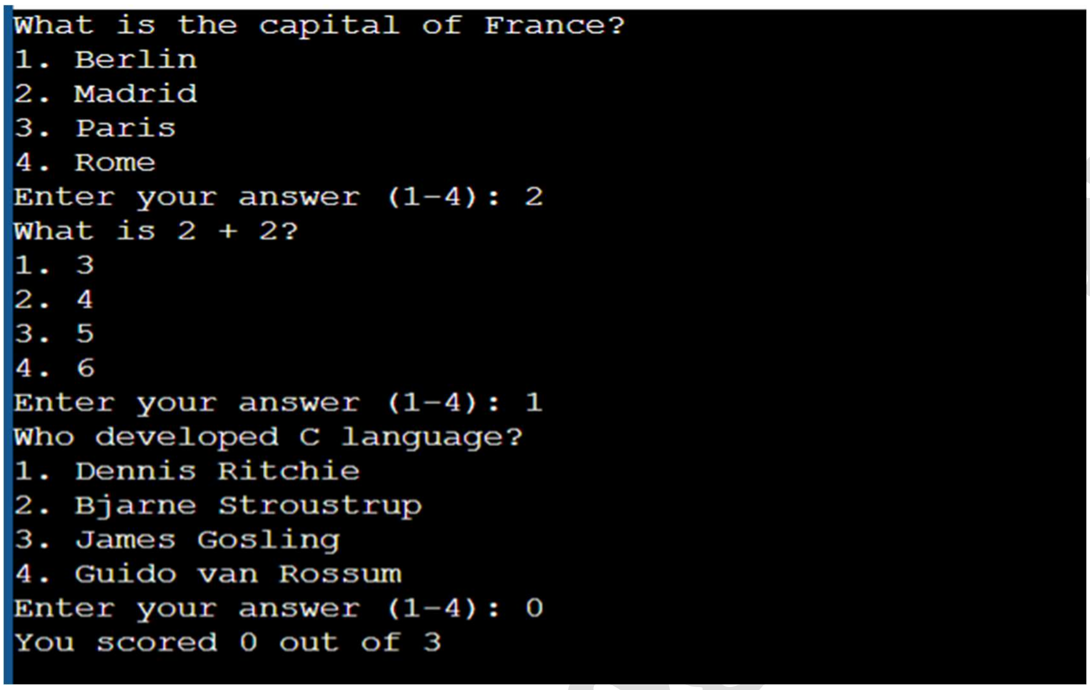
- **Objective**: To create an online quiz system where users can take a quiz, get a score, and review their performance.
- **Techniques**: Arrays, Functions, User Input Handling
- **Description**: The project will allow users to answer multiple-choice questions and display their scores after completion. The quiz will store questions and options in arrays and calculate the score based on user answers.
- **Solution**: Below is a simplified implementation of an online quiz system.

Source Code:

```
#include <stdio.h>
```

```
struct Question {
    char question[200];
    char options[4][100];
    int correctOption;
};

void displayQuestion(struct Question q) {
    printf("%s\n", q.question);
    for (int i = 0; i < 4; i++) {
        printf("%d. %s\n", i + 1, q.options[i]);
    }
}

int main() {
    struct Question questions[3] = {
        {"What is the capital of France?", {"Berlin", "Madrid", "Paris", "Rome"}, 2},
        {"What is 2 + 2?", {"3", "4", "5", "6"}, 1},
        {"Who developed C language?", {"Dennis Ritchie", "Bjarne Stroustrup", "James Gosling", "Guido
van Rossum"}, 0}
    };

    int score = 0, answer;
    for (int i = 0; i < 3; i++) {
        displayQuestion(questions[i]);
        printf("Enter your answer (1-4): ");
        scanf("%d", &answer);
        if (answer - 1 == questions[i].correctOption) {
            score++;
        }
    }
    printf("You scored %d out of 3\n", score);
    return 0;
}
```

Output:



```
What is the capital of France?
1. Berlin
2. Madrid
3. Paris
4. Rome
Enter your answer (1-4): 2
What is 2 + 2?
1. 3
2. 4
3. 5
4. 6
Enter your answer (1-4): 1
Who developed C language?
1. Dennis Ritchie
2. Bjarne Stroustrup
3. James Gosling
4. Guido van Rossum
Enter your answer (1-4): 0
You scored 0 out of 3
```

5. To-Do List Application

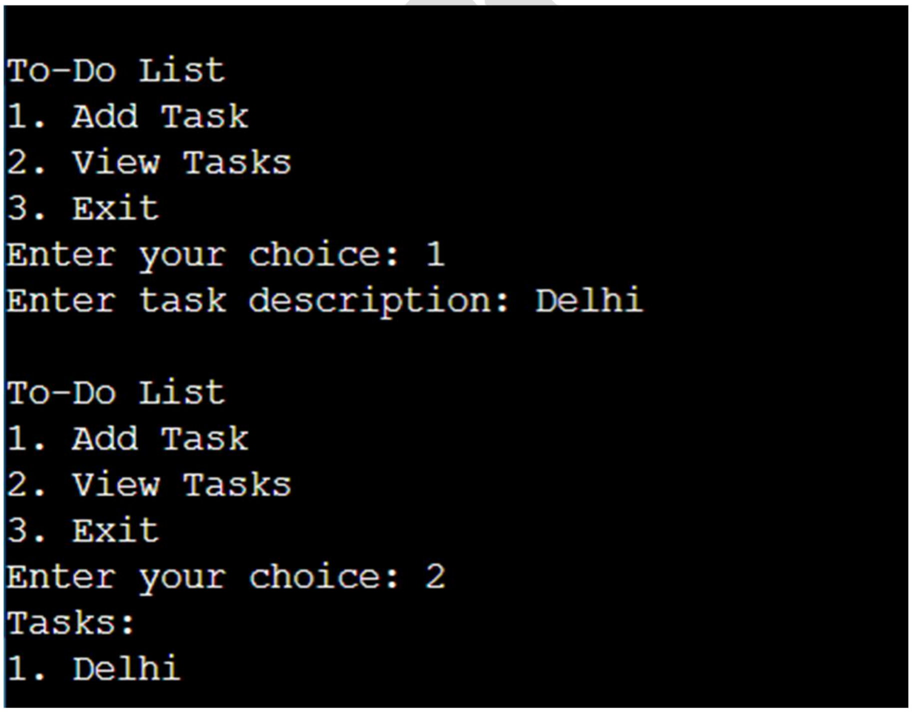
- **Objective**: To create a simple to-do list application where users can add, view, and delete tasks.
- **Techniques**: Arrays, Functions
- **Description**: This project will manage a list of tasks. Users will be able to add tasks, view all tasks, and delete completed tasks.

Source Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Task {
    int id;
    char description[100];
};
```

```
void addTask(struct Task tasks[], int *taskCount) {
    printf("Enter task description: ");
    getchar(); // consume newline
    fgets(tasks[*taskCount].description, sizeof(tasks[*taskCount].description), stdin);
    tasks[*taskCount].id = *taskCount + 1;
    (*taskCount)++;
}

void viewTasks(struct Task tasks[], int taskCount) {
    if (taskCount == 0) {
        printf("No tasks available.\n");
    } else {
        printf("Tasks:\n");
        for (int i = 0; i < taskCount; i++) {
            printf("%d. %s", tasks[i].id, tasks[i].description);
        }
    }
}

int main() {
    struct Task tasks[10];
    int taskCount = 0;
    int choice;

    while (1) {
        printf("\nTo-Do List\n");
        printf("1. Add Task\n");
        printf("2. View Tasks\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                addTask(tasks, &taskCount);
                break;
            case 2:
                viewTasks(tasks, taskCount);
                break;
            case 3:
                exit(0);
            default:
                printf("Invalid choice! Try again.\n");
        }
    }
    return 0;
}
```

Output:



```
To-Do List
1. Add Task
2. View Tasks
3. Exit
Enter your choice: 1
Enter task description: Delhi

To-Do List
1. Add Task
2. View Tasks
3. Exit
Enter your choice: 2
Tasks:
1. Delhi
```

Thanks