DSA ASSIGNMENT II

Dynamic Memory Allocation and List

By Naveen Jayaraj

RA2311026050064

Answer

1) Implementation of Matrix Multiplication using Dynamic Memory Allocation. Ensure to allocate the memory using appropriate functions and access the array using pointers.

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

int* allocateMatrix(int rows, int cols) {
    return (int*)malloc(rows * cols * sizeof(int));
}

void inputMatrix(int* matrix, int rows, int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Enter element [%d][%d]: ", i, j);
            scanf("%d", (matrix + i * cols + j));
        }
    }
}

void printMatrix(int* matrix, int rows, int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d ", *(matrix + i * cols + j));
        }
        printf("\n");
    }
}

int* multiplyMatrices(int* matrix1, int* matrix2, int rows1, int cols1, int cols2) {
    int* result = allocateMatrix(rows1, cols2);
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols2; j++) {
            *(result + i * cols2 + j) = 0;
            for (int k = 0; k < cols1; k++) {
                *(result + i * cols2 + j) += (*(matrix1 + i * cols1 + k)) *
                    (*(matrix2 + k * cols2 + j));
            }
        }
    }
    return result;
}

int main() {
    int rows1, cols1, rows2, cols2;

    printf("Enter number of rows and columns for the first matrix: ");
    scanf("%d %d", &rows1, &cols1);

    printf("Enter number of rows and columns for the second matrix: ");
    scanf("%d %d", &rows2, &cols2);

    if (cols1 != rows2) {
        printf("Error: Matrix multiplication is not possible with the given
            dimensions.\n");
        return -1;
    }
```

Output:

```
Enter number of rows and columns for the first matrix: 3
3
Enter number of rows and columns for the second matrix: 3
3
Enter elements for the first matrix:
Enter element [0][0]: 1
Enter element [0][1]: 2
Enter element [0][2]: 3
Enter element [1][0]: 4
Enter element [1][1]: 5
Enter element [1][2]: 6
Enter element [2][0]: 7
Enter element [2][1]: 8
Enter element [2][2]: 9
Enter elements for the second matrix:
Enter element [0][0]: 1
Enter element [0][1]: 2
Enter element [0][2]: 3
Enter element [1][0]: 4
Enter element [1][1]: 5
Enter element [1][2]: 6
Enter element [2][0]: 7
Enter element [2][1]: 8
Enter element [2][2]: 9
Result of matrix multiplication:
30 36 42
66 81 96
102 126 150
```

2)

```c
#include <stdio.h>
#include <string.h>
#define MAX 100
#define NAME_LEN 50

void displayList(char students[MAX][NAME_LEN], int count) {
    if (count == 0) {
        printf("The student list is empty.\n");
    } else {
        printf("Student list: [");
        for (int i = 0; i < count; i++) {
            printf("%s", students[i]);
            if (i < count - 1) {
                printf(", ");
            }
        }
        printf("]\n");
    }
}
void createList(char students[MAX][NAME_LEN], int *count) {
    printf("Enter the number of students: ");
    scanf("%d", count);
    getchar();

    for (int i = 0; i < *count; i++) {
        printf("Enter student name %d: ", i + 1);
        fgets(students[i], NAME_LEN, stdin);
        students[i][strcspn(students[i], "\n")] = 0;
    }
    displayList(students, *count);
}
```

```c
void insertStudent(char students[MAX][NAME_LEN], int *count) {
    if (*count == MAX) {
        printf("Student list is full, cannot insert more students.\n");
        return;
    }

    char name[NAME_LEN];
    int pos;

    printf("Enter the student's name to insert: ");
    getchar();
    fgets(name, NAME_LEN, stdin);
    name[strcspn(name, "\n")] = 0;

    printf("Enter the position (0-based index) to insert the student: ");
    scanf("%d", &pos);

    if (pos < 0 || pos > *count) {
        printf("Invalid position.\n");
        return;
    }

    for (int i = *count; i > pos; i--) {
        strcpy(students[i], students[i - 1]);
    }
    strcpy(students[pos], name);
    (*count)++;

    displayList(students, *count);
}

void deleteStudent(char students[MAX][NAME_LEN], int *count) {
    if (*count == 0) {
        printf("The student list is empty.\n");
        return;
    }

    char choice;
    printf("Delete by name or position? (n/p): ");
    getchar();
    scanf("%c", &choice);

    if (choice == 'n') {
        char name[NAME_LEN];
        int found = 0;
        printf("Enter the student's name to delete: ");
        getchar();
        fgets(name, NAME_LEN, stdin);
        name[strcspn(name, "\n")] = 0;

        for (int i = 0; i < *count; i++) {
            if (strcmp(students[i], name) == 0) {
                for (int j = i; j < *count - 1; j++) {
                    strcpy(students[j], students[j + 1]);
                }
                (*count)--;
                found = 1;
                break;
            }
        }

        if (!found) {

```

```
95              printf("Student not found.\n");
96          }
97      } else if (choice == 'p') {
98          int pos;
99          printf("Enter the position to delete the student: ");
100         scanf("%d", &pos);
101
102         if (pos < 0 || pos >= *count) {
103             printf("Invalid position.\n");
104             return;
105         }
106
107         for (int i = pos; i < *count - 1; i++) {
108             strcpy(students[i], students[i + 1]);
109         }
110         (*count)--;
111     } else {
112         printf("Invalid choice.\n");
113     }
114
115     displayList(students, *count);
116 }
117
118 void searchStudent(char students[MAX][NAME_LEN], int count) {
119     char name[NAME_LEN];
120     int found = 0;
121
122     if (count == 0) {
123         printf("The student list is empty.\n");
124         return;
125     }
126
127     printf("Enter the student's name to search: ");
128     getchar();
129     fgets(name, NAME_LEN, stdin);
130     name[strcspn(name, "\n")] = 0;
131
132     for (int i = 0; i < count; i++) {
133         if (strcmp(students[i], name) == 0) {
134             printf("%s found at position %d\n", name, i);
135             found = 1;
136             break;
137         }
138     }
139
140     if (!found) {
141         printf("%s not found in the list.\n", name);
142     }
143 }
```

```c
int main() {
    char students[MAX][NAME_LEN];
    int count = 0;
    int choice;

    do {
        printf("\n1. Create the list of students\n");
        printf("2. Insert a new student\n");
        printf("3. Delete a student\n");
        printf("4. Display student list\n");
        printf("5. Search for a student\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                createList(students, &count);
                break;
            case 2:
                insertStudent(students, &count);
                break;
            case 3:
                deleteStudent(students, &count);
                break;
            case 4:
                displayList(students, count);
                break;
            case 5:
                searchStudent(students, count);
                break;
            case 6:
                printf("Exiting the program...\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 6);

    return 0;
}
```

Output:

```
1. Create the list of students
2. Insert a new student
3. Delete a student
4. Display student list
5. Search for a student
6. Exit
Enter your choice:
```

```
Enter your choice: 1
Enter the number of students: 4
Enter student name 1: Naveen
Enter student name 2: Manu
Enter student name 3: Aswin
Enter student name 4: Madhav
Student list: [Naveen, Manu, Aswin, Madhav]
```

```
Enter your choice: 2
Enter the student's name to insert: Anamika
Enter the position (0-based index) to insert the student: 4
Student list: [Naveen, Manu, Aswin, Madhav, Anamika]
```

```
Enter your choice: 3
Delete by name or position? (n/p): p
Enter the position to delete the student: 1
Student list: [Naveen, Aswin, Madhav, Anamika]
```

```
Enter your choice: 4
Student list: [Naveen, Aswin, Madhav, Anamika]
```

```
Enter your choice: 5
Enter the student's name to search: Madhav
Madhav found at position 2
```