Lab Practical 10 Submission

Roll No. and Name: 21BEI036 Meet Patel

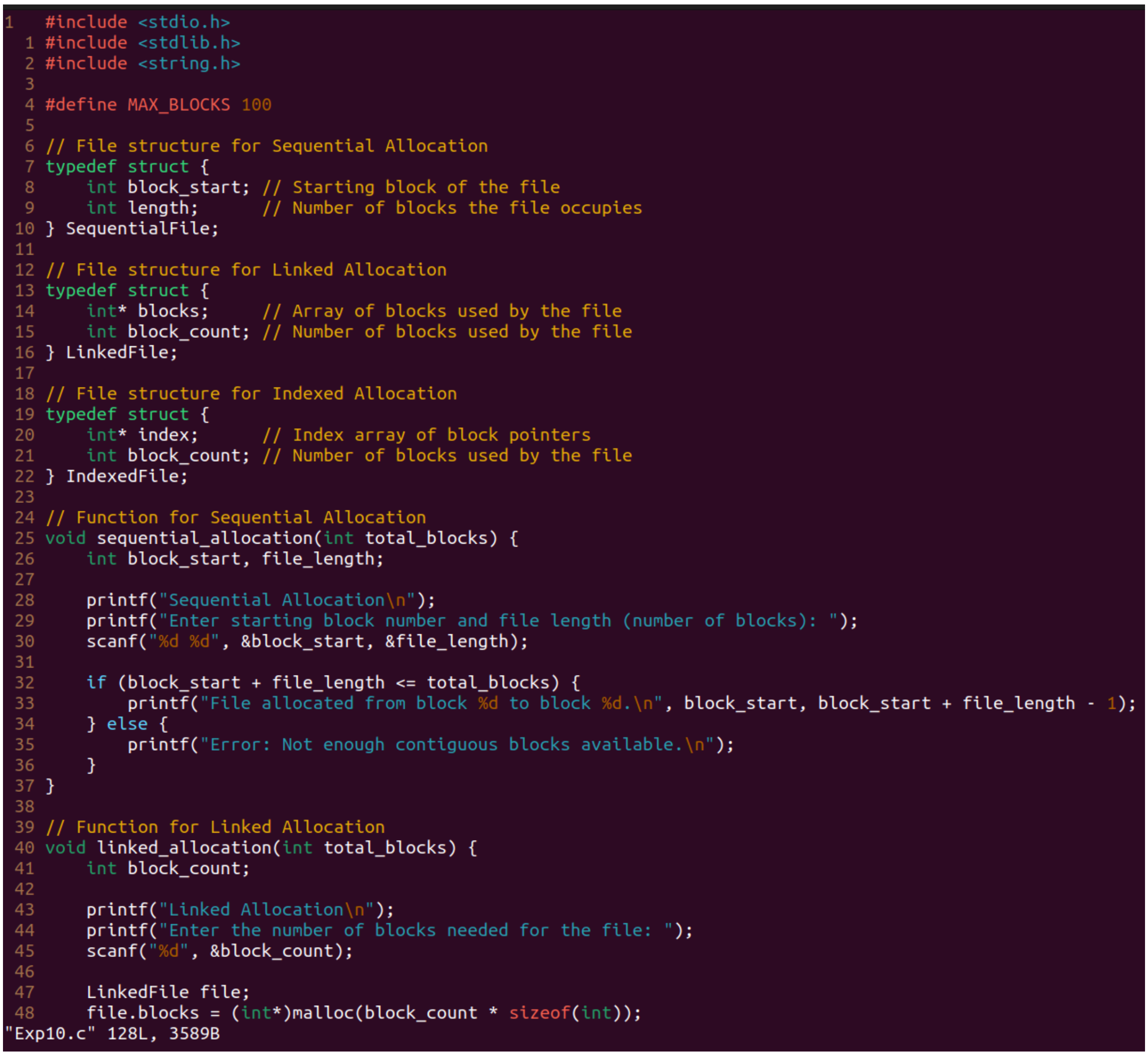
Course Code and Name: 2CSOE53 OS

Aim:

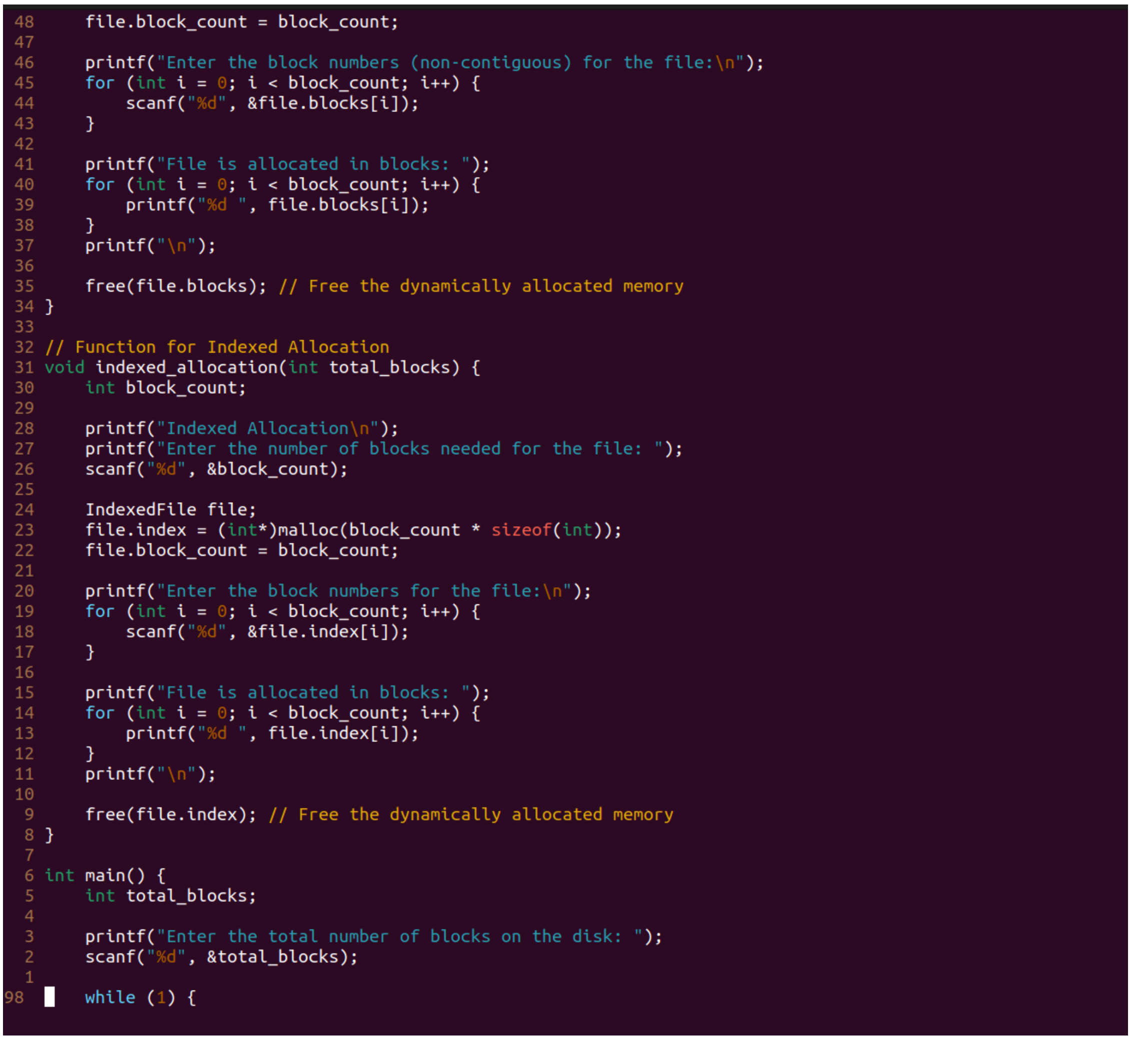
Write a C program to simulate the following file allocation strategies.

a) Sequential b) Linked c)) Indexed

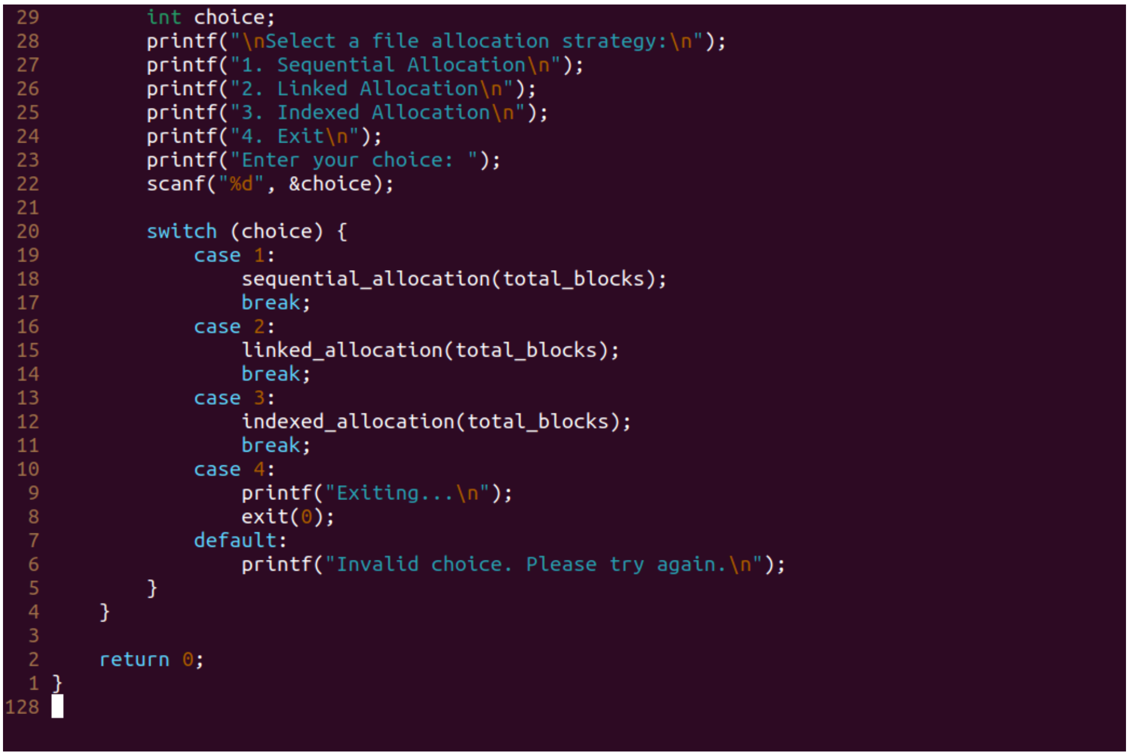
code:



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

#define MAX_BLOCKS 100

// File structure for Sequential Allocation
typedef struct {
    int block_start; // Starting block of the file
    int length;      // Number of blocks the file occupies
} SequentialFile;

// File structure for Linked Allocation
typedef struct {
    int* blocks;     // Array of blocks used by the file
    int block_count; // Number of blocks used by the file
} LinkedFile;

// File structure for Indexed Allocation
typedef struct {
    int* index;      // Index array of block pointers
    int block_count; // Number of blocks used by the file
} IndexedFile;

// Function for Sequential Allocation
void sequential_allocation(int total_blocks) {
    int block_start, file_length;

    printf("Sequential Allocation\n");
    printf("Enter starting block number and file length (number of blocks): ");
    scanf("%d %d", &block_start, &file_length);

    if (block_start + file_length <= total_blocks) {
        printf("File allocated from block %d to block %d.\n", block_start, block_start + file_length - 1);
    } else {
        printf("Error: Not enough contiguous blocks available.\n");
    }
}

// Function for Linked Allocation
void linked_allocation(int total_blocks) {
    int block_count;

    printf("Linked Allocation\n");
    printf("Enter the number of blocks needed for the file: ");
    scanf("%d", &block_count);

    LinkedFile file;
    file.blocks = (int*)malloc(block_count * sizeof(int));
"Exp10.c" 128L, 3589B
```

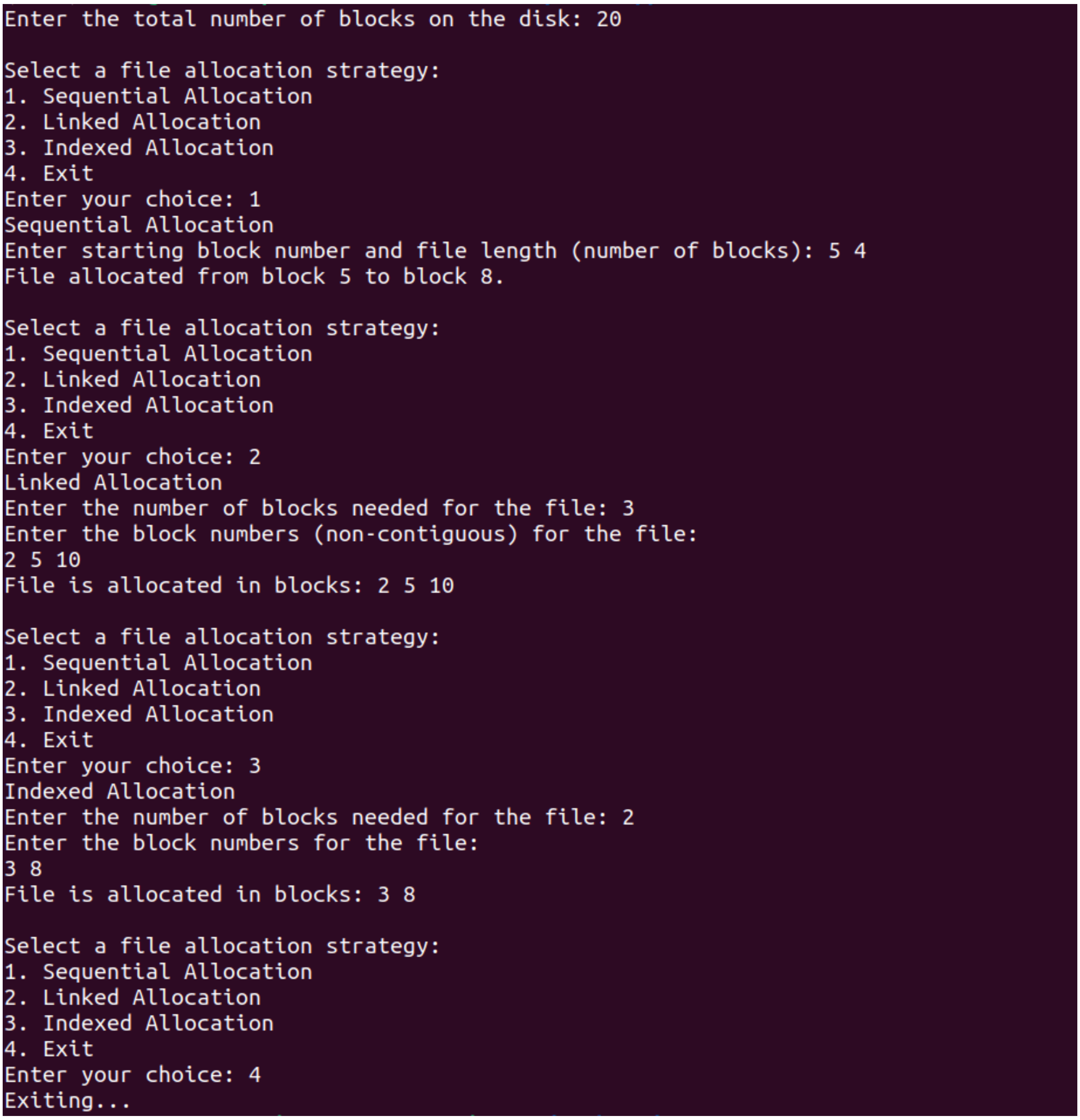


```c
    file.block_count = block_count;

    printf("Enter the block numbers (non-contiguous) for the file:\n");
    for (int i = 0; i < block_count; i++) {
        scanf("%d", &file.blocks[i]);
    }

    printf("File is allocated in blocks: ");
    for (int i = 0; i < block_count; i++) {
        printf("%d ", file.blocks[i]);
    }
    printf("\n");

    free(file.blocks); // Free the dynamically allocated memory
}

// Function for Indexed Allocation
void indexed_allocation(int total_blocks) {
    int block_count;

    printf("Indexed Allocation\n");
    printf("Enter the number of blocks needed for the file: ");
    scanf("%d", &block_count);

    IndexedFile file;
    file.index = (int*)malloc(block_count * sizeof(int));
    file.block_count = block_count;

    printf("Enter the block numbers for the file:\n");
    for (int i = 0; i < block_count; i++) {
        scanf("%d", &file.index[i]);
    }

    printf("File is allocated in blocks: ");
    for (int i = 0; i < block_count; i++) {
        printf("%d ", file.index[i]);
    }
    printf("\n");

    free(file.index); // Free the dynamically allocated memory
}

int main() {
    int total_blocks;

    printf("Enter the total number of blocks on the disk: ");
    scanf("%d", &total_blocks);

    while (1) {
```



```c
        int choice;
        printf("\nSelect a file allocation strategy:\n");
        printf("1. Sequential Allocation\n");
        printf("2. Linked Allocation\n");
        printf("3. Indexed Allocation\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                sequential_allocation(total_blocks);
                break;
            case 2:
                linked_allocation(total_blocks);
                break;
            case 3:
                indexed_allocation(total_blocks);
                break;
            case 4:
                printf("Exiting...\n");
                exit(0);
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }

    return 0;
}
```

output:



```
Enter the total number of blocks on the disk: 20

Select a file allocation strategy:
1. Sequential Allocation
2. Linked Allocation
3. Indexed Allocation
4. Exit
Enter your choice: 1
Sequential Allocation
Enter starting block number and file length (number of blocks): 5 4
File allocated from block 5 to block 8.

Select a file allocation strategy:
1. Sequential Allocation
2. Linked Allocation
3. Indexed Allocation
4. Exit
Enter your choice: 2
Linked Allocation
Enter the number of blocks needed for the file: 3
Enter the block numbers (non-contiguous) for the file:
2 5 10
File is allocated in blocks: 2 5 10

Select a file allocation strategy:
1. Sequential Allocation
2. Linked Allocation
3. Indexed Allocation
4. Exit
Enter your choice: 3
Indexed Allocation
Enter the number of blocks needed for the file: 2
Enter the block numbers for the file:
3 8
File is allocated in blocks: 3 8

Select a file allocation strategy:
1. Sequential Allocation
2. Linked Allocation
3. Indexed Allocation
4. Exit
Enter your choice: 4
Exiting...
```