EXP 36: With linked allocation, each file is a linked list of disk blocks; the disk blocks may be scattered anywhere on the disk. The directory contains a pointer to the first and last blocks of the file. Each block contains a pointer to the next block. Design a C program to simulate the file allocation strategy.

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

#define MAX_BLOCKS 100
#define MAX_FILES 10
#define MAX_FILE_BLOCKS 10

typedef struct {
    int next;  // -1 if it's the last block
    int used;  // 0 = free, 1 = used
} DiskBlock;

typedef struct {
    char name[20];
    int start;
    int end;
    int blockCount;
} File;

DiskBlock disk[MAX_BLOCKS];
File files[MAX_FILES];
int fileCount = 0;
```

```
// Allocate a free block and return its index
int allocateBlock() {
    for (int i = 0; i < MAX_BLOCKS; i++) {
        if (!disk[i].used) {
            disk[i].used = 1;
            disk[i].next = -1; // default to end of file
            return i;
        }
    }
    return -1; // no free blocks
}

void createFile() {
    if (fileCount >= MAX_FILES) {
        printf("File limit reached.\n");
        return;
    }

    char name[20];
    int blocks;

    printf("Enter file name: ");
    scanf("%s", name);
    printf("Enter number of blocks: ");
    scanf("%d", &blocks);

    if (blocks > MAX_FILE_BLOCKS) {
        printf("Exceeded max file blocks.\n");
        return;
```

```
}

int blockIndexes[MAX_FILE_BLOCKS];
for (int i = 0; i < blocks; i++) {
   int b = allocateBlock();
   if (b == -1) {
      printf("Not enough space. Rolling back allocation.\n");
      for (int j = 0; j < i; j++)
         disk[blockIndexes[j]].used = 0;
      return;
   }
   blockIndexes[i] = b;
}

// Link blocks
for (int i = 0; i < blocks - 1; i++) {
   disk[blockIndexes[i]].next = blockIndexes[i + 1];
}

// Save file info
strcpy(files[fileCount].name, name);
files[fileCount].start = blockIndexes[0];
files[fileCount].end = blockIndexes[blocks - 1];
files[fileCount].blockCount = blocks;
fileCount++;

printf("File '%s' created.\n", name);
printf("Blocks allocated: ");
for (int i = 0; i < blocks; i++)
```

```
        printf("%d ", blockIndexes[i]);
    printf("\n");
}

void readFile() {
    char name[20];
    printf("Enter file name to read: ");
    scanf("%s", name);

    for (int f = 0; f < fileCount; f++) {
        if (strcmp(files[f].name, name) == 0) {
            printf("Reading file '%s':\n", name);
            int current = files[f].start;
            int count = 1;
            while (current != -1) {
                printf("Block %d (Record %d)\n", current, count++);
                current = disk[current].next;
            }
            return;
        }
    }
    printf("File not found.\n");
}

void displayDisk() {
    printf("\nDisk Blocks:\n");
    printf("Block | Used | Next\n");
    for (int i = 0; i < MAX_BLOCKS; i++) {
        printf("%5d |  %3d  | %4d\n", i, disk[i].used, disk[i].next);
```

```
    }
}

int main() {
    int choice;
    while (1) {
        printf("\n--- Linked File Allocation ---\n");
        printf("1. Create File\n");
        printf("2. Read File\n");
        printf("3. Display Disk\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: createFile(); break;
            case 2: readFile(); break;
            case 3: displayDisk(); break;
            case 4: exit(0);
            default: printf("Invalid choice.\n");
        }
    }
    return 0;
}
```

Sample Output

```
--- Linked File Allocation ---
1. Create File
2. Read File
3. Display Disk
4. Exit
Enter your choice: 1
Enter file name: Games
Enter number of blocks: 7
File 'Games' created.
Blocks allocated: 0 1 2 3 4 5 6

--- Linked File Allocation ---
1. Create File
2. Read File
3. Display Disk
4. Exit
Enter your choice: 2
Enter file name to read: Games
Reading file 'Games':
Block 0 (Record 1)
Block 1 (Record 2)
Block 2 (Record 3)
Block 3 (Record 4)
Block 4 (Record 5)
Block 5 (Record 6)
Block 6 (Record 7)

--- Linked File Allocation ---
1. Create File
2. Read File
3. Display Disk
4. Exit
Enter your choice: 3

Disk Blocks:
Block | Used | Next
    0 |    1 |    1
    1 |    1 |    2
    2 |    1 |    3
    3 |    1 |    4
    4 |    1 |    5
    5 |    1 |    6
    6 |    1 |   -1
    7 |    0 |    0
    8 |    0 |    0
    9 |    0 |    0
   10 |    0 |    0
   11 |    0 |    0
   12 |    0 |    0
   13 |    0 |    0
   14 |    0 |    0
   15 |    0 |    0
   16 |    0 |    0
   17 |    0 |    0
   18 |    0 |    0
   19 |    0 |    0
   20 |    0 |    0
   21 |    0 |    0
   22 |    0 |    0
   23 |    0 |    0
   24 |    0 |    0
   25 |    0 |    0
   26 |    0 |    0
   27 |    0 |    0
   28 |    0 |    0
   29 |    0 |    0
   30 |    0 |    0
   31 |    0 |    0
   32 |    0 |    0
   33 |    0 |    0
   34 |    0 |    0
   35 |    0 |    0
   36 |    0 |    0
   37 |    0 |    0
   38 |    0 |    0
   39 |    0 |    0
   40 |    0 |    0
   41 |    0 |    0
   42 |    0 |    0
   43 |    0 |    0
   44 |    0 |    0
   45 |    0 |    0
   46 |    0 |    0
   47 |    0 |    0
   48 |    0 |    0
   49 |    0 |    0
   50 |    0 |    0
   51 |    0 |    0
   52 |    0 |    0
   53 |    0 |    0
   54 |    0 |    0
   55 |    0 |    0
   56 |    0 |    0
   57 |    0 |    0
   58 |    0 |    0
   59 |    0 |    0
   60 |    0 |    0
   61 |    0 |    0
   62 |    0 |    0
   63 |    0 |    0
   64 |    0 |    0
   65 |    0 |    0
   66 |    0 |    0
   67 |    0 |    0
   68 |    0 |    0
   69 |    0 |    0
   70 |    0 |    0
   71 |    0 |    0
   72 |    0 |    0
   73 |    0 |    0
   74 |    0 |    0
   75 |    0 |    0
   76 |    0 |    0
   77 |    0 |    0
   78 |    0 |    0
   79 |    0 |    0
   80 |    0 |    0
   81 |    0 |    0
   82 |    0 |    0
   83 |    0 |    0
   84 |    0 |    0
   85 |    0 |    0
   86 |    0 |    0
   87 |    0 |    0
   88 |    0 |    0
   89 |    0 |    0
   90 |    0 |    0
   91 |    0 |    0
   92 |    0 |    0
   93 |    0 |    0
   94 |    0 |    0
   95 |    0 |    0
   96 |    0 |    0
   97 |    0 |    0
   98 |    0 |    0
   99 |    0 |    0

--- Linked File Allocation ---
1. Create File
2. Read File
3. Display Disk
4. Exit
Enter your choice: 4

--------------------------------
Process exited after 47.69 seconds with return value 0
Press any key to continue . . .
```