1. 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

typedef struct {

int next;

int used;

} Block;

typedef struct {

char name[20];

int start;

int end;

int length;

} File;

Block disk[MAX\_BLOCKS];

File files[MAX\_FILES];

int main() {

int totalBlocks, fileCount, i, j, k;

printf("Enter total number of memory blocks: ");

scanf("%d", &totalBlocks);

for (i = 0; i < totalBlocks; i++) {

disk[i].used = 0;

disk[i].next = -1;

}

printf("Enter number of files: ");

scanf("%d", &fileCount);

for (i = 0; i < fileCount; i++) {

printf("\nEnter file name: ");

scanf("%s", files[i].name);

printf("Enter number of blocks required for %s: ", files[i].name);

scanf("%d", &files[i].length);

int allocated = 0, prev = -1;

files[i].start = -1;

for (j = 0; j < totalBlocks && allocated < files[i].length; j++) {

if (disk[j].used == 0) {

disk[j].used = 1;

if (files[i].start == -1)

files[i].start = j;

if (prev != -1)

disk[prev].next = j;

prev = j;

allocated++;

}

}

if (allocated < files[i].length) {

printf("Not enough memory for file %s. Allocation failed.\n", files[i].name);

for (j = 0; j < totalBlocks; j++) {

if (disk[j].used && (j == files[i].start || disk[j].next != -1)) {

disk[j].used = 0;

disk[j].next = -1;

}

}

i--; continue;

}

files[i].end = prev;

}

printf("\nLinked File Allocation Table:\n");

printf("File\tStart\tEnd\tBlocks\n");

for (i = 0; i < fileCount; i++) {

printf("%s\t%d\t%d\t", files[i].name, files[i].start, files[i].end);

int current = files[i].start;

while (current != -1) {

printf("%d ", current);

current = disk[current].next;

}

printf("\n");

}

return 0;

}