

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

a) Sequential

```
#include <stdio.h>

#define MAX 100

int main() {
    int memory[MAX], startBlock, length, i, j, choice;

    for(i = 0; i < MAX; i++)
        memory[i] = 0;

    printf("Sequential File Allocation Simulation\n");

    while(1) {
        printf("\nEnter starting block and length of the file: ");
        scanf("%d %d", &startBlock, &length);

        if(startBlock < 0 || startBlock + length > MAX) {
            printf("Invalid block range. Try again.\n");
            continue;
        }

        int allocated = 1;
        for(j = startBlock; j < startBlock + length; j++) {
            if(memory[j] == 1) {
                allocated = 0;
                break;
            }
        }

        if(allocated) {
            for(j = startBlock; j < startBlock + length; j++)
                memory[j] = 1;

            printf("File allocated from block %d to %d\n", startBlock,
startBlock + length - 1);
        } else {
            printf("Blocks not available. File not allocated.\n");
        }

        printf("Do you want to enter another file? (1 for Yes / 0 for No): ");
        scanf("%d", &choice);
    }
}
```

```
        if(choice == 0)
            break;
    }

    printf("\nMemory Block Status:\n");
    for(i = 0; i < MAX; i++) {
        printf("%d", memory[i]);
        if((i + 1) % 10 == 0)
            printf("\n");
    }

    return 0;
```

Output

Sequential File Allocation Simulation

Enter starting block and length of the file: 9

78

File allocated from block 9 to 86

Do you want to enter another file? (1 for Yes / 0 for No):