NAME: MUHAMMAD SAIM NOMANI
ROLL NO: DT-22030
SUBJECT: OPERATING SYSTEM
CODE: CT-353
DATA SCIENCE
THIRD YEAR

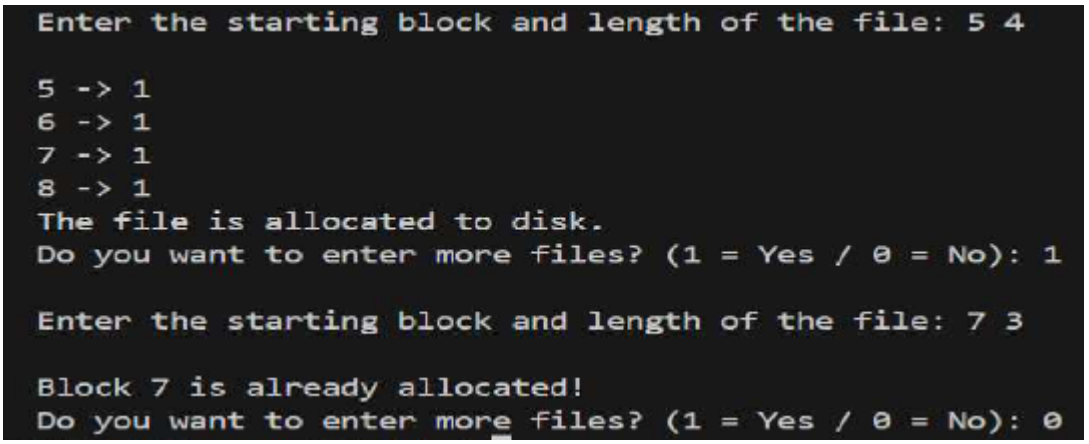
OS LAB: 14

a) SEQUENTIAL:

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

int main() {
    int f[50], i, st, j, len, c;

    // Initialize all blocks as free (0)
    for (i = 0; i < 50; i++)
        f[i] = 0;

    do {
        printf("\nEnter the starting block and length of file: ");
        scanf("%d%d", &st, &len);

        for (j = st; j < st + len; j++) {
            if (f[j] == 0) {
                f[j] = 1;
                printf("\n%d -> %d", j, f[j]);
            } else {
                printf("Block already allocated\n");
                break;
            }
        }

        if (j == st + len)
            printf("\nThe file is allocated to disk\n");

        printf("\nDo you want to enter more files? (1 - Yes / 0 - No): ");
        scanf("%d", &c);
```

```
    } while (c == 1);

    // getch(); // Uncomment if using Turbo C or similar environment

    return 0;
}
```

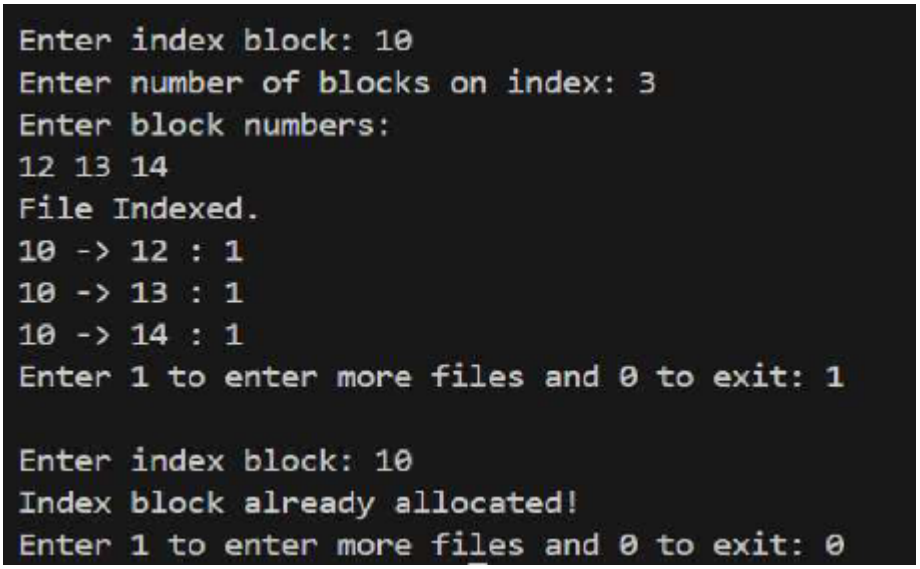
OUTPUT:



b) Indexed:

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

int f[50], inde[50];

int main() {
    int i, k, j, n, c, p;

    // Initialize all blocks as free
    for (i = 0; i < 50; i++)
        f[i] = 0;

    do {
```

```
        printf("Enter index block: ");
        scanf("%d", &p);

        if (f[p] == 0) {
            f[p] = 1;
            printf("Enter number of files on index: ");
            scanf("%d", &n);
        } else {
            printf("Block already allocated\n");
            continue;
        }

        // Input index blocks
        for (i = 0; i < n; i++) {
            scanf("%d", &inde[i]);
        }

        // Check if any index block is already allocated
        int already_allocated = 0;
        for (i = 0; i < n; i++) {
            if (f[inde[i]] == 1) {
                printf("Block %d already allocated\n", inde[i]);
                already_allocated = 1;
                break;
            }
        }
        if (already_allocated) continue;

        // Allocate index blocks
        for (j = 0; j < n; j++) {
            f[inde[j]] = 1;
        }

        printf("\nAllocated\nFile indexed:\n");
        for (k = 0; k < n; k++) {
            printf("%d -> %d : %d\n", p, inde[k], f[inde[k]]);
        }

        printf("Enter 1 to enter more files and 0 to exit: ");
        scanf("%d", &c);

        if (c != 1) break;

    } while (1);

    // getch(); // Uncomment if using Turbo C or similar environment
    return 0;
}
```



c) Linked:

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

int main() {
    int f[50], p, i, j, k, a, st, len, c;

    // Initialize all blocks as free
    for (i = 0; i < 50; i++)
        f[i] = 0;

    printf("Enter how many blocks that are already allocated: ");
    scanf("%d", &p);

    printf("\nEnter the block numbers that are already allocated:\n");
    for (i = 0; i < p; i++) {
        scanf("%d", &a);
        f[a] = 1;
    }

    while (1) {
        printf("Enter the starting index block & length: ");
        scanf("%d%d", &st, &len);

        k = len;
```

```
        for (j = st; j < (k + st); j++) {
            if (f[j] == 0) {
                f[j] = 1;
                printf("\n%d -> %d", j, f[j]);
            } else {
                printf("\n%d -> file is already allocated", j);
                k++;  // Increase k to extend length due to already allocated block
            }
        }

        printf("\nIf you want to enter one more file? (yes-1 / no-0): ");
        scanf("%d", &c);

        if (c != 1)
            break;
    }

    // getch(); // Uncomment if using Turbo C or similar environment
    return 0;
}
```

```
Enter how many blocks are already allocated: 3
Enter the block numbers that are already allocated:
3 5 9

Enter the starting index block and length: 2 4

2 -> 1
3 -> Block is already allocated
4 -> 1
5 -> Block is already allocated
6 -> 1
7 -> 1
Do you want to enter one more file? (1 = Yes / 0 = No): 1

Enter the starting index block and length: 10 3

10 -> 1
11 -> 1
12 -> 1
Do you want to enter one more file? (1 = Yes / 0 = No): 0
```