

LAB-14

HUZAIFA SALMAN DT-34

a) Sequential

CODE:

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

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

    // clrscr();
    for (i = 0; i < 50; i++)
        f[i] = 0;

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

        int allocated = 1;

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

        if (allocated)
            printf("\nThe file is allocated to disk.");

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

    } while (c == 1);

    getch();
    return 0;
}
```

:

Enter the starting block and length of the file: 5 4

5 -> 1

6 -> 1

7 -> 1

8 -> 1

The file is allocated to disk.

Do you want to enter more files? (1 = Yes / 0 = No): 1

Enter the starting block and length of the file: 7 3

Block 7 is already allocated!

Do you want to enter more files? (1 = Yes / 0 = No): 0

PS C:\6th-sems\OS labs>

b) Indexed

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

int main() {
    int f[50], i, j, k, indexBlock, n, c, inde[50];

    // clrscr();
    for (i = 0; i < 50; i++)
        f[i] = 0;

    do {
        printf("\nEnter index block: ");
        scanf("%d", &indexBlock);

        if (f[indexBlock] == 0) {
            f[indexBlock] = 1;

            printf("Enter number of blocks on index: ");
            scanf("%d", &n);

            printf("Enter block numbers:\n");
            for (i = 0; i < n; i++)
                scanf("%d", &inde[i]);

            int allocated = 1;

            for (i = 0; i < n; i++) {
                if (f[inde[i]] == 1) {
                    printf("Block %d is already allocated!\n", inde[i]);
```

```

        allocated = 0;
        break;
    }
}

if (allocated) {
    for (j = 0; j < n; j++)
        f[inde[j]] = 1;

    printf("File Indexed.\n");
    for (k = 0; k < n; k++)
        printf("%d -> %d : %d\n", indexBlock, inde[k], f[inde[k]]);
}

} else {
    printf("Index block already allocated!\n");
}

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

} while (c == 1);

getch();
return 0;
}

```

OUTPUT:

```

Enter index block: 10
Enter number of blocks on index: 3
Enter block numbers:
12 13 14
File Indexed.
10 -> 12 : 1
10 -> 13 : 1
10 -> 14 : 1
Enter 1 to enter more files and 0 to exit: 1

Enter index block: 10
Index block already allocated!
Enter 1 to enter more files and 0 to exit: 0
PS C:\6th-sems\OS labs>

```

c) Linked

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

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

    clrscr();
    for (i = 0; i < 50; i++)
        f[i] = 0;

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

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

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

        k = len;

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

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

    } while (c == 1);

    getch();
    return 0;
}
```

```
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):
```

```
Enter the starting index block and length: 10 3
```

```
10 -> 1
```

