# Course Name: Operating systems

# LAB: 14

# Submitted By: Ebaad Khan

# Roll: DT-22045

## PROGRAM:

## Sequential:

```
#include <stdio.h>

#include <stdlib.h>


int main() {

   int f[50], i, st, len, j, c;


   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) {
            allocated = 0;
            break;
        }
    }

    if (allocated) {
        for (j = st; j < (st + len); j++) {
            f[j] = 1;
            printf("\n%d -> Allocated", j);
        }
        printf("\nFile successfully allocated.\n");
    } else {
        printf("\nBlock already allocated. File not allocated.\n");
    }

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

  return 0;
}
```

## Indexed:

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

```
#include <stdlib.h>

int main() {
  int f[50], i, p, n, c, inde[50];

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

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

    if (f[p] != 0) {
      printf("Index block already allocated. Try again.\n");
      continue;
    }

    f[p] = 1;

    printf("Enter number of blocks for the file: ");
    scanf("%d", &n);

    printf("Enter the blocks: ");
    int allocated = 1;
    for (i = 0; i < n; i++) {
      scanf("%d", &inde[i]);
```

```
        if (f[inde[i]] != 0)
            allocated = 0;
    }

    if (!allocated) {
        printf("Some blocks already allocated. File not indexed.\n");
        continue;
    }

    for (i = 0; i < n; i++)
        f[inde[i]] = 1;

    printf("File indexed successfully.\nIndex block %d points to:\n", p);
    for (i = 0; i < n; i++)
        printf("%d -> %d (Allocated)\n", p, inde[i]);

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

  } while (c == 1);

  return 0;
}
```

## Linked:

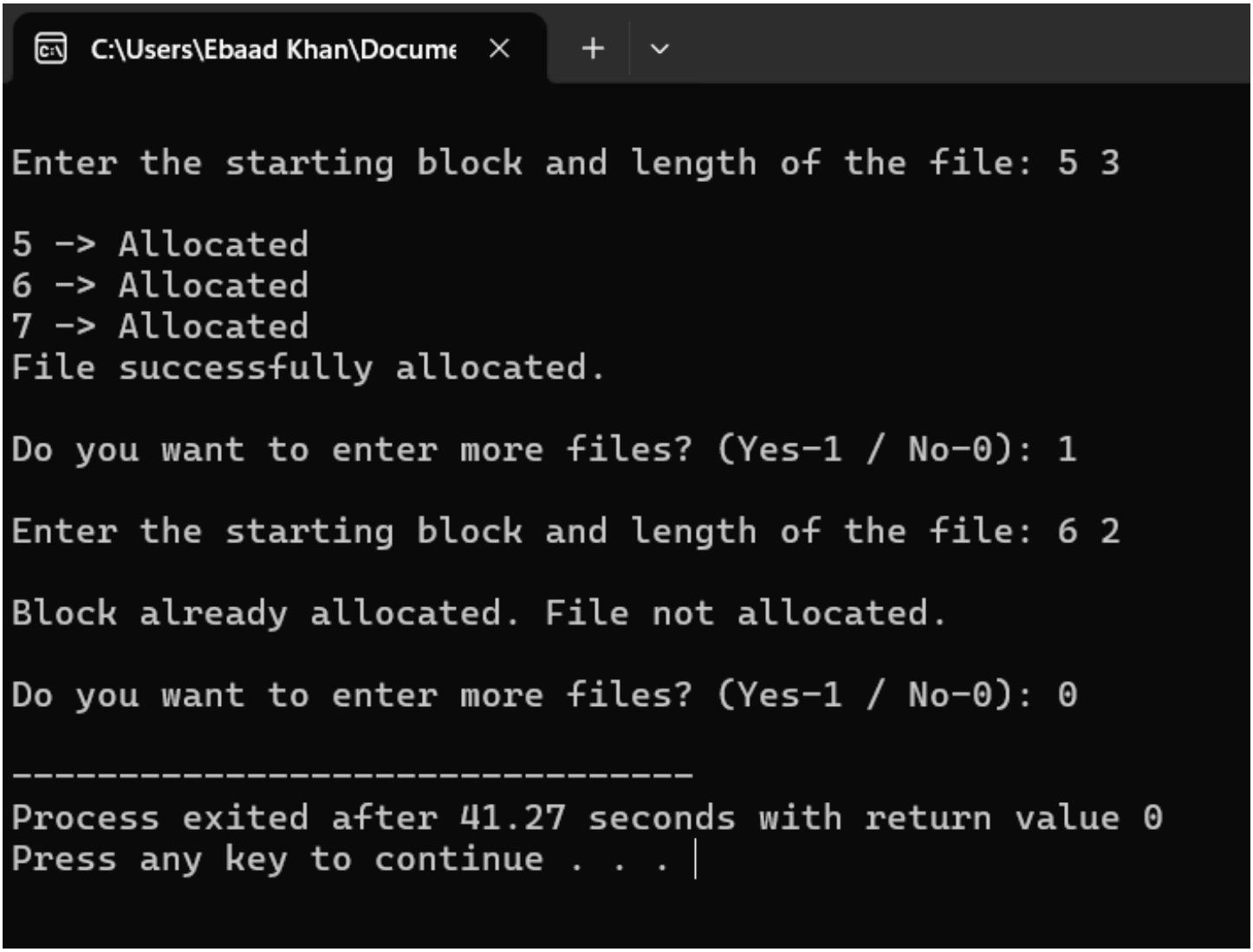
```
#include <stdio.h>
```

```
#include <stdlib.h>

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

  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: ");
  for (i = 0; i < p; i++) {
    scanf("%d", &a);
    f[a] = 1;
  }

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

    k = len;
    j = st;

    printf("Allocating blocks:\n");
    while (k > 0) {
```

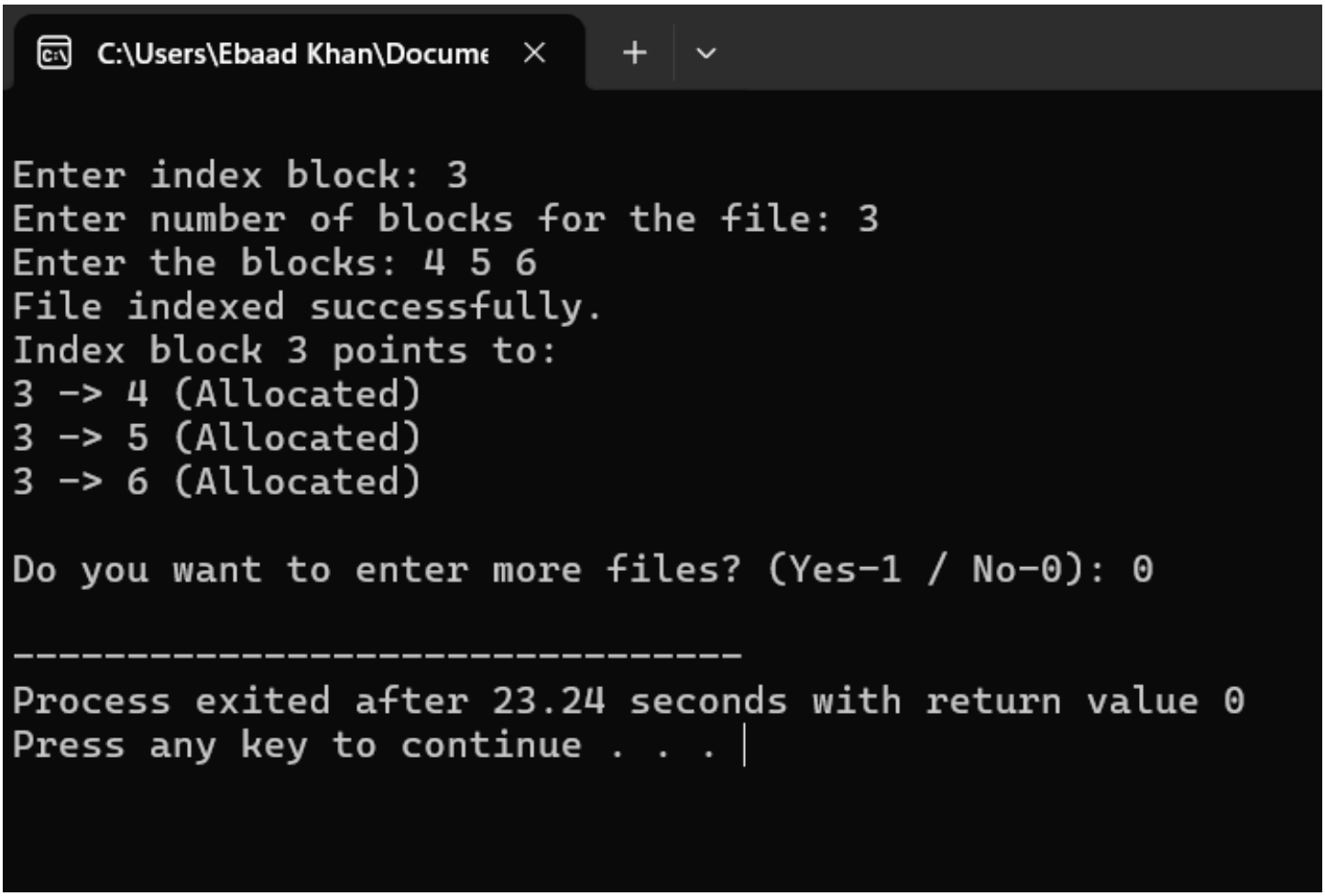
```
        if (j >= 50) {
            printf("Reached end of disk. File allocation incomplete.\n");
            break;
        }

        if (f[j] == 0) {
            f[j] = 1;
            printf("%d -> Allocated\n", j);
            k--;
        } else {
            printf("%d -> Already allocated\n", j);
        }

        j++;
    }

    if (k == 0)
        printf("File allocated successfully.\n");

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

  } while (c == 1);

  return 0;
}
```

**OUTPUT:**

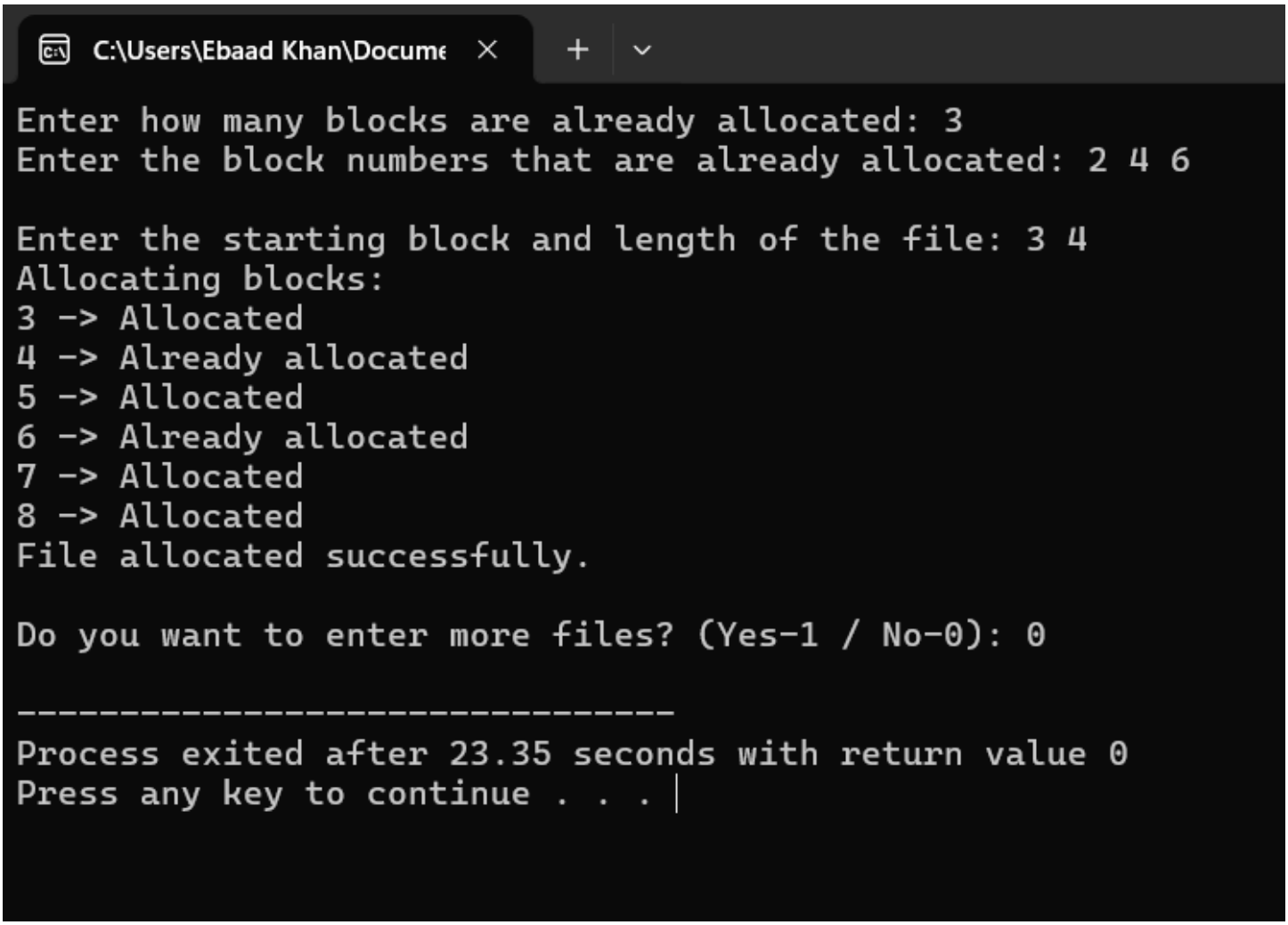
**Sequential:**




**Indexed:**


C:\Users\Ebaad Khan\Docume
Enter index block: 3
Enter number of blocks for the file: 3
Enter the blocks: 4 5 6
File indexed successfully.
Index block 3 points to:
3 -> 4 (Allocated)
3 -> 5 (Allocated)
3 -> 6 (Allocated)
Do you want to enter more files? (Yes-1 / No-0): 0
--------------------------------
Process exited after 23.24 seconds with return value 0
Press any key to continue . . .

**Linked:**


C:\Users\Ebaad Khan\Docume
Enter how many blocks are already allocated: 3
Enter the block numbers that are already allocated: 2 4 6
Enter the starting block and length of the file: 3 4
Allocating blocks:
3 -> Allocated
4 -> Already allocated
5 -> Allocated
6 -> Already allocated
7 -> Allocated
8 -> Allocated
File allocated successfully.
Do you want to enter more files? (Yes-1 / No-0): 0
--------------------------------
Process exited after 23.35 seconds with return value 0
Press any key to continue . . .