OPERATING SYSTEMS (CT-353) LAB 14

NAME: Ajiya Anwar ROLL NO: DT-22006

CODE:

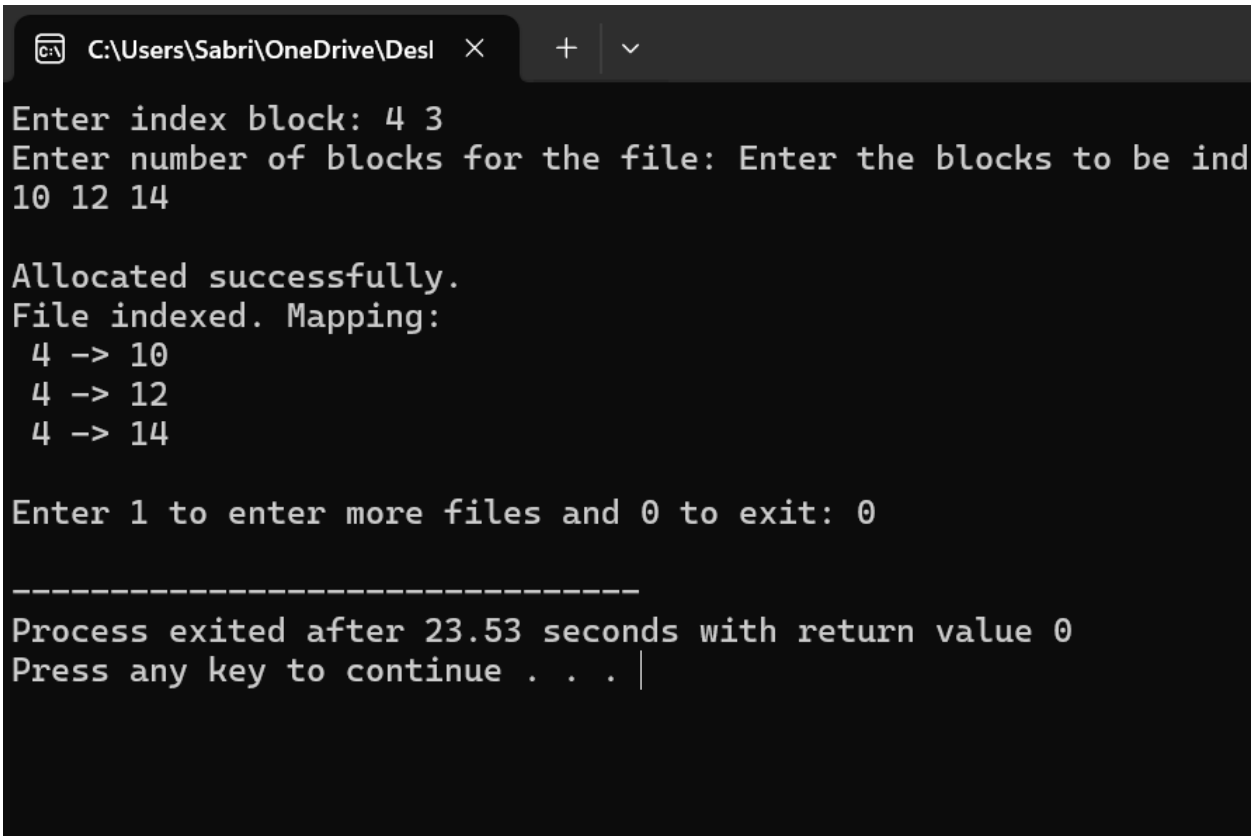
Indexed:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int f[50], i, k, j, indexBlock[50], 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 blocks for the file: ");
         scanf("%d", &n);
      } else {
         printf("Block already allocated!\n");
         continue;
      }

      printf("Enter the blocks to be indexed:\n");
      for(i = 0; i < n; i++)
         scanf("%d", &indexBlock[i]);
```

```
    // Check if any of the blocks are already allocated
    for(i = 0; i < n; i++) {
      if(f[indexBlock[i]] == 1) {
        printf("Block %d is already allocated!\n", indexBlock[i]);
        f[p] = 0; // deallocate index block
        for(j = 0; j < i; j++)
          f[indexBlock[j]] = 0;
        goto skip;
      }
    }

    // Allocate the blocks
    for(i = 0; i < n; i++)
      f[indexBlock[i]] = 1;

    printf("\nAllocated successfully.\n");
    printf("File indexed. Mapping:\n");
    for(i = 0; i < n; i++)
      printf(" %d -> %d\n", p, indexBlock[i]);

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

  } while(c == 1);

  return 0;
}
```

OPERATING SYSTEMS (CT-353) LAB 14

OUTPUT:



CODE:

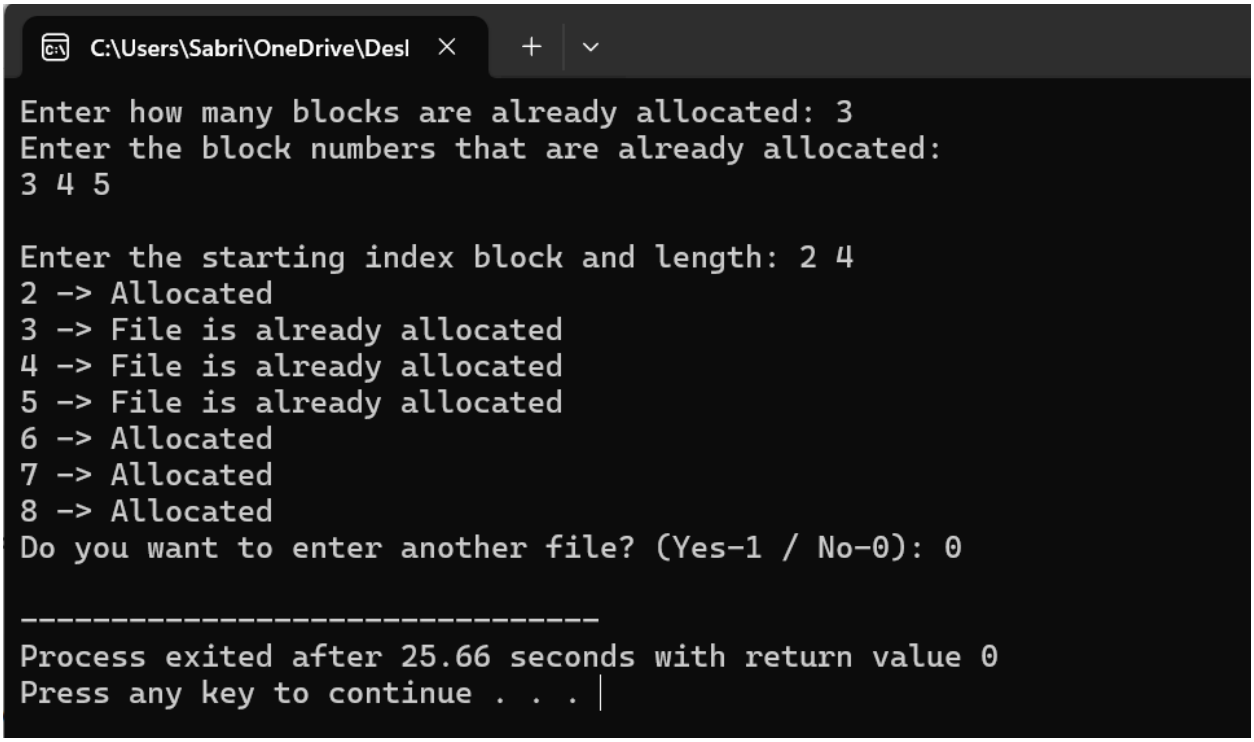
linked

```
#include <stdio.h>

#include <stdlib.h>


int main() {

   int f[50], p, i, j, a, st, len, n, c, k;


   // Initialize all disk blocks to free

   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("%d -> Allocated\n", j);
      } else {
         printf("%d -> File is already allocated\n", j);
         k++; // Extend range to allocate full length
      }
   }

   printf("Do you want to enter another file? (Yes-1 / No-0): ");
   scanf("%d", &c);
} while(c == 1);

return 0;
```

}

OUTPUT:



CODE:

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 & length of file: ");
```

```
        scanf("%d%d", &st, &len);

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

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

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

    } while(c == 1);

    return 0;
}
```

OPERATING SYSTEMS (CT-353) LAB 14

OUTPUT: