/\*

# PROGRAM TO IMPLEMENT FILE ALLOCATION STRATEGIES - INDEXED

Ashis Solomon

CS4B 17

MDL20CS035

\*/

**CODE:**

#include <stdio.h>

#define MAX 100

int blocks[MAX];

int indices[10];

typedef struct {

  int start;

  int len;

  int alloc[25];

  int flag;

}

files;

files file[10];

void allocate(int fno) {

  int i = file[fno].start;

  int count = 0;

  do {

    if ((i == file[fno].start) && (blocks[i] == 1)) {

      file[fno].flag = 0;

      break;

    }

    if (blocks[i] == 0) {

      blocks[i] = 1;

      file[fno].alloc[count] = i;

      count++;

    }

    i = (i + 1) % MAX;

  } while (i != file[fno].start && count < file[fno].len);

  if (count == file[fno].len)

    file[fno].flag = 1;

  else

    file[fno].flag = 0;

}

void display(int n) {

  int i, j;

  printf("\n-----------------------------\n");

  printf("The indices locations are: \n");

  printf("File No.\tIndex block\tStarting block\tLength\tStatus\n");

  for (int i = 0; i < n; i++) {

    if (indices[i] != -1 && file[i].flag == 1)

      printf("%d\t\t%d\t\t%d\t\t%d\tAllocated\n", (i + 1), indices[i], file[i].start,

        file[i].len);

    else

      printf("%d\t\t-\t\t-\t\t-\tUnallocated\n", (i + 1));

  }

  for (i = 0; i < n; i++) {

    if (file[i].flag == 1) {

      printf("\nIndex Block of File %d\n", (i + 1));

      printf("\nBlock No.\tBlock Location\n");

      for (j = 0; j < file[i].len; j++)

        printf("%d\t\t%d\n", (j + 1), file[i].alloc[j]);

    } } }

int main() {

    int n, filled, x;

    for (int i = 0; i < MAX; i++)

      blocks[i] = 0;

    printf("Enter the number of blocks already occupied: ");

    scanf("%d", & filled);

    for (int i = 0; i < filled; i++) {

      printf("Enter the location of the occupied block: ");

      scanf("%d", & x);

      blocks[x] = 1; }

    printf("Enter the number of files to be allocated: ");

    scanf("%d", & n);

    for (int i = 0; i < n; i++) {

      printf("\nEnter the location of the index block for file %d: ", (i + 1));

      scanf("%d", & x);

      if (blocks[x] == 0) {

        blocks[x] = 1;

        indices[i] = x;

      } else {  indices[i] = -1;

        printf("Index block is already occupied! Unable to store file %d\n", (i + 1));

        continue;

      }

      printf("Enter the starting location of file %d: ", (i + 1));

      scanf("%d", & file[i].start);

      printf("Enter the length of file %d: ", (i + 1));

      scanf("%d", & file[i].len);

      allocate(i);

      if (file[i].flag == 1)

        printf("File %d was successfully allocated!\n", (i + 1));

      else

        printf("Starting location already occupied! Unable to allocate disk space to file % d\n ", (i+1));

        }

      display(n);

      return 0;

    }

**OUTPUT:**

