```c
/*
 * Name: Preston Tighe
 * Program 6
 *
 * Command: gcc program6.c -o program6 && ./program6 && cat results.txt
 */

#include <stdio.h>
#include <stdlib.h>
#define CLOCK_SIZE 4

typedef struct tag {
    int s[CLOCK_SIZE][4];
}

QUEUE;
QUEUE queue;
int nf = 0;

int inblock(int a[CLOCK_SIZE][4], int page, char operation) {
    for (int i = 0; i < CLOCK_SIZE; i++) {
        if (a[i][1] == page) {
            a[i][2] = 1;
            if (operation == 'w') {
                a[i][3] = 1;
            }
            return (1);
        }
    }
    return (0);
}

int search(int a[CLOCK_SIZE][4]) {
    for (int i = 0; i < CLOCK_SIZE; i++) {
        if (a[(nf + i) % 4][2] == 0 && a[(nf + i) % 4][3] == 0) {
            nf = (nf + i) % 4;
            return (nf);
        }
    }
    for (int i = 0; i < CLOCK_SIZE; i++) {
        if (a[(nf + i) % 4][2] == 0 && a[(nf + i) % 4][3] == 1) {
            nf = (nf + i) % 4;
            return (nf);
        }
        a[(nf + i) % 4][2] = 0;
    }
    return (search(a));
}

void writeClockToFile(FILE * filePtr, int page, char operation) {
    int pa;
    char op;
    pa = page;
    op = operation;
    fprintf(filePtr, "FRAME         PAGE       USE MODIFY\n");
    if (inblock(queue.s, pa, op)) {
        return;
    } else if (queue.s[CLOCK_SIZE - 1][1] != -1) {
        int j = search(queue.s);
        queue.s[j][1] = page;
        queue.s[j][2] = 1;
        if (operation == 'w') {
            queue.s[j][3] = 1;
        } else {
            queue.s[j][3] = 0;
        }
        nf = (j + 1) % 4;
        return;
    } else {
        for (int i = 0; i < CLOCK_SIZE; i++) {
            if (queue.s[i + nf][1] == -1) {
                queue.s[i + nf][1] = page;
                queue.s[i + nf][2] = 1;
                if (operation == 'w') {
                    queue.s[i + nf][3] = 1;
                }
                nf = (i + nf + 1) % 4;
                return;
            }
        }
    }
}

int main() {
    int page;
    char operation;
    for (int i = 0; i < CLOCK_SIZE; i++) {
        queue.s[i][0] = i;
        queue.s[i][1] = -1;
    }
    for (int i = 0; i < CLOCK_SIZE; i++) {}
    char inFileName[] = "testdata.txt";
    FILE * inFilePtr = fopen(inFileName, "r");
    if (inFilePtr == NULL) {
        printf("File %s could not be opened.\n", inFileName);
        exit(1);
    }
    char outFileName[] = "results.txt";
    FILE * outFilePtr = fopen(outFileName, "w");
    if (outFilePtr == NULL) {
        printf("File %s could not be opened.\n", outFileName);
        exit(1);
    }
    fscanf(inFilePtr, "%d%c", & page, & operation);
    while (!feof(inFilePtr)) {
        fprintf(outFilePtr, "Page referenced: %d %c\n", page, operation);
        writeClockToFile(outFilePtr, page, operation);
        for (int i = 0; i < CLOCK_SIZE; i++) {
            char spa = queue.s[i][1] < 0 ? '\0' : ' ';
            if (i == nf) {
                fprintf(outFilePtr, " %d           %c%d         %d         %d <- next frame\n", queue.s[i][0], spa, queue.s[i][1], queue.s[i][2], queue.s[i][3]);
            } else {
                fprintf(outFilePtr, " %d           %c%d         %d         %d    \n", queue.s[i][0], spa, queue.s[i][1], queue.s[i][2], queue.s[i][3]);
            }
        }
        fprintf(outFilePtr, "\n");
        fscanf(inFilePtr, "%d%c", & page, & operation);
    }
    /* end while */
    fclose(inFilePtr);
    fclose(outFilePtr);
    return (0);
```

```c
};
```