Implement the following page replacement algorithms using c

1.FIFO

2. Optimal

3.LRU

```
#include <stdio.h>

#define MAX_FRAMES 10
#define MAX_PAGES 100

int isInFrames(int frames[], int frameCount, int page) {
    for (int i = 0; i < frameCount; i++) {
        if (frames[i] == page) return 1;
    }
    return 0;
}

int findLRU(int time[], int frameCount) {
    int min = time[0], pos = 0;
    for (int i = 1; i < frameCount; i++) {
        if (time[i] < min) {
            min = time[i];
            pos = i;
        }
    }
    return pos;
}

int findOptimal(int pages[], int frames[], int pageCount, int frameCount, int currentIndex) {
    int farthest = -1, index = -1;
    for (int i = 0; i < frameCount; i++) {
        int j;
```

```
        for (j = currentIndex + 1; j < pageCount; j++) {
            if (frames[i] == pages[j]) {
                if (j > farthest) {
                    farthest = j;
                    index = i;
                }
                break;
            }
        }
        if (j == pageCount) return i; // Not used again
    }
    return (index == -1) ? 0 : index;
}

void printFrames(int frames[], int frameCount, int pageFault) {
    for (int i = 0; i < frameCount; i++) {
        if (frames[i] == -1) printf("- ");
        else printf("%d ", frames[i]);
    }
    if (pageFault) printf(" (Page Fault)");
    printf("\n");
}

void fifo(int pages[], int pageCount, int frameCount) {
    int frames[MAX_FRAMES], next = 0, faults = 0;

    for (int i = 0; i < frameCount; i++) frames[i] = -1;

    printf("\n--- FIFO Page Replacement ---\n");
    for (int i = 0; i < pageCount; i++) {
        if (!isInFrames(frames, frameCount, pages[i])) {
```

```
            frames[next] = pages[i];
            next = (next + 1) % frameCount;
            faults++;
            printFrames(frames, frameCount, 1);
        } else {
            printFrames(frames, frameCount, 0);
        }
    }
    printf("Total Page Faults (FIFO): %d\n", faults);
}

void optimal(int pages[], int pageCount, int frameCount) {
    int frames[MAX_FRAMES], faults = 0;

    for (int i = 0; i < frameCount; i++) frames[i] = -1;

    printf("\n--- Optimal Page Replacement ---\n");
    for (int i = 0; i < pageCount; i++) {
        if (!isInFrames(frames, frameCount, pages[i])) {
            int replaceIndex = -1;
            for (int j = 0; j < frameCount; j++) {
                if (frames[j] == -1) {
                    replaceIndex = j;
                    break;
                }
            }
            if (replaceIndex == -1)
                replaceIndex = findOptimal(pages, frames, pageCount, frameCount, i);

            frames[replaceIndex] = pages[i];
            faults++;
```

```
            printFrames(frames, frameCount, 1);
        } else {
            printFrames(frames, frameCount, 0);
        }
    }
    printf("Total Page Faults (Optimal): %d\n", faults);
}

void lru(int pages[], int pageCount, int frameCount) {
    int frames[MAX_FRAMES], time[MAX_FRAMES], faults = 0, counter = 0;

    for (int i = 0; i < frameCount; i++) {
        frames[i] = -1;
        time[i] = 0;
    }

    printf("\n--- LRU Page Replacement ---\n");
    for (int i = 0; i < pageCount; i++) {
        counter++;
        int found = 0;
        for (int j = 0; j < frameCount; j++) {
            if (frames[j] == pages[i]) {
                time[j] = counter;
                found = 1;
                break;
            }
        }

        if (!found) {
            int pos = -1;
            for (int j = 0; j < frameCount; j++) {
```

```
            if (frames[j] == -1) {
                pos = j;
                break;
            }
        }
        if (pos == -1)
            pos = findLRU(time, frameCount);

        frames[pos] = pages[i];
        time[pos] = counter;
        faults++;
        printFrames(frames, frameCount, 1);
    } else {
        printFrames(frames, frameCount, 0);
    }
  }
  printf("Total Page Faults (LRU): %d\n", faults);
}

int main() {
  int pages[MAX_PAGES], pageCount, frameCount, choice;

  printf("Enter number of pages: ");
  scanf("%d", &pageCount);

  printf("Enter page reference string: ");
  for (int i = 0; i < pageCount; i++)
    scanf("%d", &pages[i]);

  printf("Enter number of frames: ");
  scanf("%d", &frameCount);
```

```
    fifo(pages, pageCount, frameCount);
    optimal(pages, pageCount, frameCount);
    lru(pages, pageCount, frameCount);
    return 0;
}
```

```
Enter number of pages: 9
Enter page reference string: 1 2 3 1 0 2 3 1 6
Enter number of frames: 3

--- FIFO Page Replacement ---
1 - -  (Page Fault)
1 2 -  (Page Fault)
1 2 3  (Page Fault)
1 2 3
0 2 3  (Page Fault)
0 2 3
0 2 3
0 1 3  (Page Fault)
0 1 6  (Page Fault)
Total Page Faults (FIFO): 6

--- Optimal Page Replacement ---
1 - -  (Page Fault)
1 2 -  (Page Fault)
1 2 3  (Page Fault)
1 2 3
0 2 3  (Page Fault)
0 2 3
0 2 3
1 2 3  (Page Fault)
6 2 3  (Page Fault)
Total Page Faults (Optimal): 6

--- LRU Page Replacement ---
1 - -  (Page Fault)
1 2 -  (Page Fault)
1 2 3  (Page Fault)
1 2 3
1 0 3  (Page Fault)
1 0 2  (Page Fault)
3 0 2  (Page Fault)
3 1 2  (Page Fault)
3 1 6  (Page Fault)
Total Page Faults (LRU): 8
```