Write a C program to simulate page replacement algorithms:
a)FIFO
b)LRU
c)Optimal

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

void printFrames(int frames[], int f) {
   for (int i = 0; i < f; i++) {

      if (frames[i] == -1)
         printf("- ");
      else
         printf("%d ",frames[i]);
   }
   printf("\n");
}

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

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

```
int findOptimal(int pages[], int frames[], int n, int f, int index) {
    int farthest = index, pos = -1;
    for (int i = 0; i < f; i++) {
        int j;
        for (j = index; j < n; j++) {
            if (frames[i] == pages[j]) {
                if (j > farthest) {
                    farthest = j;
                    pos = i;
                }
                break;
            }
        }
        if (j == n) return i;
    }
    return (pos == -1) ? 0 : pos;
}

void fifo(int pages[], int n, int f) {
    int frames[f], front = 0, faults = 0;
    for (int i = 0; i < f; i++) frames[i] = -1;

    printf("\nFIFO Page Replacement:\n");

    for (int i = 0; i < n; i++) {
        if (!isInFrame(frames, f, pages[i])) {
            frames[front] = pages[i];
            front = (front + 1) % f;
            faults++;
        }
        printf("PR No . %d : ",i+1);
        printFrames(frames, f);
    }

    printf("FIFO Page Faults: %d\n", faults);
}

void lru(int pages[], int n, int f) {
    int frames[f], time[f], faults = 0, counter = 0;
```

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

    printf("\nLRU Page Replacement:\n");

    for (int i = 0; i < n; i++) {
        counter++;
        if (!isInFrame(frames, f, pages[i])) {
            int index = -1;
            for (int j = 0; j < f; j++) {
                if (frames[j] == -1) {
                    index = j;
                    break;
                }
            }
            if (index == -1)
                index = findLRU(time, f);
            frames[index] = pages[i];
            time[index] = counter;
            faults++;
        } else {
            for (int j = 0; j < f; j++) {
                if (frames[j] == pages[i])
                    time[j] = counter;
            }
        }
        printf("PR No . %d : ",i+1);
        printFrames(frames, f);
    }

    printf("LRU Page Faults: %d\n", faults);
}

void optimal(int pages[], int n, int f) {
    int frames[f], faults = 0;
    for (int i = 0; i < f; i++) frames[i] = -1;
    printf("\nOptimal Page Replacement:\n");

    for (int i = 0; i < n; i++) {
        if (!isInFrame(frames, f, pages[i])) {
```

```
            int index = -1;
            for (int j = 0; j < f; j++) {
                if (frames[j] == -1) {
                    index = j;
                    break;
                }
            }
            if (index == -1)
                index = findOptimal(pages, frames, n, f, i + 1);
            frames[index] = pages[i];
            faults++;
        }
        printf("PR No . %d : ",i+1);
        printFrames(frames, f);
    }
    printf("Optimal Page Faults: %d\n", faults);
}

int main() {
    int n, f;
    printf("Enter number of frames: ");
    scanf("%d", &f);
    printf("Enter length of reference string : ");
    scanf("%d", &n);

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

    fifo(pages, n, f);
    lru(pages, n, f);
    optimal(pages, n, f);



    return 0;
}
```

OUTPUT

```
Enter number of frames: 3
Enter length of reference string : 6
Enter page reference string:
1 3 0 3 5 6

FIFO Page Replacement:
PR No . 1 : 1 - -
PR No . 2 : 1 3 -
PR No . 3 : 1 3 0
PR No . 4 : 1 3 0
PR No . 5 : 5 3 0
PR No . 6 : 5 6 0
FIFO Page Faults: 5

LRU Page Replacement:
PR No . 1 : 1 - -
PR No . 2 : 1 3 -
PR No . 3 : 1 3 0
PR No . 4 : 1 3 0
PR No . 5 : 5 3 0
PR No . 6 : 5 3 6
LRU Page Faults: 5

Optimal Page Replacement:
PR No . 1 : 1 - -
PR No . 2 : 1 3 -
PR No . 3 : 1 3 0
PR No . 4 : 1 3 0
PR No . 5 : 5 3 0
PR No . 6 : 6 3 0
Optimal Page Faults: 5
```