

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

```
int findLRU(int time[], int n) {
```

```
    int i, min = time[0], pos = 0;
```

```
    for (i = 1; i < n; ++i) {
```

```
        if (time[i] < min) {
```

```
            min = time[i];
```

```
            pos = i;
```

```
        }
```

```
    }
```

```
    return pos;
```

```
}
```

```
int main() {
```

```
    int pages[50], frame[10], time[10];
```

```
    int n, f, i, j, pos, counter = 0, page_faults = 0, flag1, flag2;
```

```
    printf("Enter the number of pages: ");
```

```
    scanf("%d", &n);
```

```
    printf("Enter the page reference string: ");
```

```
    for (i = 0; i < n; ++i)
```

```
        scanf("%d", &pages[i]);
```

```
    printf("Enter the number of frames (minimum 3): ");
```

```
    scanf("%d", &f);
```

```
    if (f < 3) {
```

```
        printf("Error: Minimum frame size must be 3!\n");
```

```
        return 0;
```

```
    }
```

```

for (i = 0; i < f; ++i)

    frame[i] = -1;


printf("\nPage Replacement Process (LRU):\n");


for (i = 0; i < n; ++i) {

    flag1 = flag2 = 0;


    // Check if page is already present in frame
    for (j = 0; j < f; ++j) {

        if (frame[j] == pages[i]) {

            counter++;

            time[j] = counter; // Update recent use time

            flag1 = flag2 = 1;

            break;

        }

    }

}


// If page not found in frame
if (flag1 == 0) {

    for (j = 0; j < f; ++j) {

        if (frame[j] == -1) {

            counter++;

            page_faults++;

            frame[j] = pages[i];

            time[j] = counter;

            flag2 = 1;

            break;

        }

    }

}

```

```

    }

    // If all frames are full
    if (flag2 == 0) {
        pos = findLRU(time, f);
        counter++;
        page_faults++;
        frame[pos] = pages[i];
        time[pos] = counter;
    }

    printf("Page %d -> Frames: ", pages[i]);
    for (j = 0; j < f; ++j) {
        if (frame[j] != -1)
            printf("%d ", frame[j]);
        else
            printf("- ");
    }
    printf("\n");
}

printf("\nTotal Page Faults = %d\n", page_faults);
printf("Page Fault Rate = %.2f%%\n", ((float)page_faults / n) * 100);

return 0;
}

```

OUTPUT

Enter the number of pages: 12

Enter the page reference string: 1 2 3 4 1 2 5 1 2 3 4 5

Enter the number of frames (minimum 3): 3

Page Replacement Process (LRU):

Page 1 -> Frames: 1 - -

Page 2 -> Frames: 1 2 -

Page 3 -> Frames: 1 2 3

Page 4 -> Frames: 4 2 3

Page 1 -> Frames: 4 1 3

Page 2 -> Frames: 4 1 2

Page 5 -> Frames: 5 1 2

Page 1 -> Frames: 5 1 2

Page 2 -> Frames: 5 1 2

Page 3 -> Frames: 3 1 2

Page 4 -> Frames: 3 4 2

Page 5 -> Frames: 3 4 5

Total Page Faults = 9

Page Fault Rate = 75.00%