Operating Systems Lab
Assignment 6

Name: Aditya Kangune
Roll number: 33323
Batch: K11

Output:

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.LRU
4.Optimal
5.Exit
Enter your choice:2


******************** FCFS ********************


Enter length of page reference sequence:7

Enter the page reference sequence:1 3 0 3 5 6 3

Enter no of frames:3

FIFO

For 1 : Miss! 1
For 3 : Miss! 1 3
For 0 : Miss! 1 3 0
For 3 : Hit! 1 3 0
For 5 : Miss! 3 0 5
For 6 : Miss! 0 5 6
For 3 : Miss! 5 6 3
Total no of page faults:6

Page Replacement Algorithms
1.Enter data
2.FIFO
3.LRU
4.Optimal
5.Exit
Enter your choice:3


******************** LRU ********************


Enter length of page reference sequence:14

Enter the page reference sequence:7 0 1 2 0 3 0 4 2 3 0 3 2 3

Enter no of frames:4

LRU

For 7 : Miss! 7
For 0 : Miss! 7 0
For 1 : Miss! 7 0 1
For 2 : Miss! 7 0 1 2
For 0 : Hit! 7 0 1 2
For 3 : Miss! 3 0 1 2
For 0 : Hit! 3 0 1 2
For 4 : Miss! 3 0 4 2
For 2 : Hit! 3 0 4 2
For 3 : Hit! 3 0 4 2
For 0 : Hit! 3 0 4 2
For 3 : Hit! 3 0 4 2
For 2 : Hit! 3 0 4 2
For 3 : Hit! 3 0 4 2
```

```
Total no of page faults:6

Page Replacement Algorithms
1.Enter data
2.FIFO
3.LRU
4.Optimal
5.Exit
Enter your choice:4


******************** Optimal ********************


Enter length of page reference sequence:14

Enter the page reference sequence:7 0 1 2 0 3 0 4 2 3 0 3 2 3

Enter no of frames:4

Optimal

For 7 : Miss! 7
For 0 : Miss! 0
For 1 : Miss! 0 1
For 2 : Miss! 0 2
For 0 : Hit! 0 2
For 3 : Miss! 0 2 3
For 0 : Hit! 0 2 3
For 4 : Miss! 0 2 3 4
For 2 : Hit! 0 2 3 4
For 3 : Hit! 0 2 3 4
For 0 : Hit! 0 2 3 4
For 3 : Hit! 0 2 3 4
For 2 : Hit! 0 2 3 4
For 3 : Hit! 0 2 3 4
Total no of page faults:6
```

```
Page Replacement Algorithms
1.Enter data
2.FIFO
3.LRU
4.Optimal
5.Exit
Enter your choice:5

Thank You!!!
```

Code:

```
/*

==================================================================
==========
 Name        : Page_Replacement_Algorithm.c
 Author      : Aditya Kangune
 Version     :
 Copyright   : Your copyright notice
 Description : Hello World in C, Ansi-style

==================================================================
==========
 */

#include <stdio.h>
#include <stdlib.h>

int p[20];  // Page sequence
int frame[20]; // frame
int hit = 0, ishit;
int miss = 0;
int pagefaultcount = 0;
int n; //Number of pages in page sequence
int nf; // Number of frames

void getData()
{
    printf("\nEnter length of page reference sequence:");
    fflush(stdout);
    scanf("%d",&n);
    printf("\nEnter the page reference sequence:");
    fflush(stdout);
    for(int i=0; i<n; i++)
        scanf("%d",&p[i]);
    printf("\nEnter no of frames:");
    fflush(stdout);
    scanf("%d",&nf);
}

void init() {
	hit = 0;
	miss = 0;
	for (int i =0; i < nf; i++) {
		frame[i] = 9999;
	}
	pagefaultcount = 0;
}

int isHit(int page) {
	ishit = 0;
	for (int j = 0; j < nf; j++) {
		if (frame[j] == page) {
			ishit = 1;
			hit = hit + 1;
			break;
		}
	}
	return ishit;
```

```c
}

void dispPgFaultCnt()
{
    printf("\nTotal no of page faults:%d",pagefaultcount);
    fflush(stdout);
}


void displayPages() {
        for (int k=0; k<nf; k++) {
            if(frame[k]!=9999) {
                    printf(" %d",frame[k]);
                    fflush(stdout);
            }
        }
}

void fifo() {
        init();
        printf("\nFIFO\n");
        fflush(stdout);
        int k = 0;
        for (int i = 0; i < n; i++) {
                printf("\nFor %d : ", p[i]);
                fflush(stdout);

                if(isHit(p[i]) == 0) {
                        printf("Miss!");
                        fflush(stdout);

                        for (k = 0; k < nf -1 ; k++) {
                                frame[k] = frame[k + 1];
                        }
                        frame[k] = p[i];
                        pagefaultcount++;
                        displayPages();
```

```
		} else {
			printf("Hit!");
			fflush(stdout);
			displayPages();
		}
	}
	dispPgFaultCnt();
}

void lru() {
	init();

	printf("\nLRU\n");
	fflush(stdout);

	int least[20];

	int i, j, k;

	for (i = 0; i< n; i++) {
		printf("\nFor %d : ", p[i]);
		fflush(stdout);
		if (isHit(p[i]) == 0) {
			printf("Miss!");
			fflush(stdout);
			for (j = 0; j < nf; j++) {
				int pg = frame[j];
				int found = 0;
				for (k = i-1; k >= 0; k--) {
					if (pg == p[k]) {
						least[j] = k;
						found = 1;
						break;
					} else {
						found = 0;
					}
				}
```

```
                    if (!found) {
                        least[j] = -9999;
                    }
                }
                int min = 9999;
                int leastindex;
                for (j = 0; j < nf; j++) {
                    if (least[j] < min) {
                        min = least[j];
                        leastindex = j;
                    }
                }
                frame[leastindex] = p[i];
                pagefaultcount++;
                displayPages();
            } else {
                printf("Hit!");
                fflush(stdout);
                displayPages();
            }
        }
        dispPgFaultCnt();
}

void optimal() {
        init();

        printf("\nOptimal\n");
        fflush(stdout);

        int opt[20];

        int i, j, k;

        for (i = 0; i< n; i++) {
            printf("\nFor %d : ", p[i]);
            fflush(stdout);
```

```
if (isHit(p[i]) == 0) {
    printf("Miss!");
    fflush(stdout);
    for (j = 0; j < nf; j++) {
        int pg = frame[j];
        int found = 0;
        for (k = i; k < n; k++) {
            if (pg == p[k]) {
                opt[j] = k;
                found = 1;
                break;
            } else {
                found = 0;
            }
            }
        if (!found) {
            opt[j] = 9999;
        }
    }
    int max = -9999;
    int mostindex;
    for (j = 0; j < nf; j++) {
        if (opt[j] > max) {
            max = opt[j];
            mostindex = j;
        }
    }

    frame[mostindex] = p[i];
    pagefaultcount++;
    displayPages();
    } else {
        printf("Hit!");
        fflush(stdout);
        displayPages();
    }
}
```

```
                    dispPgFaultCnt();

}


int main(void) {
          int choice;
             while(1)
             {
                printf("\n\nPage Replacement Algorithms\n1.Enter
data\n2.FIFO\n3.LRU\n4.Optimal\n5.Exit\nEnter your choice:");
                fflush(stdout);
                scanf("%d",&choice);
                switch(choice)
                {
                case 1:

                   getData();
                   break;
                case 2:
                   printf("\n\n******************** FCFS ********************\n\n");
                   getData();
                   fifo();
                   break;
                case 3:
                   printf("\n\n******************** LRU ********************\n\n");
                   getData();
                   lru();
                   break;
                case 4:
                   printf("\n\n******************** Optimal ********************\n\n");
                   getData();
                   optimal();
                   break;
                case 5:
                   printf("\nThank You!!!");
                   fflush(stdout);
                return 0;
                break;
              default:
                 return 0;
                 break;
              }
           }
        return 0;
}
```