Name: Devanshu Surana

Roll No.: 12

Panel: C**, Batch:** C1

OS Lab 7 Code:

Title: Memory Management (Page Replacement Algorithms)

#include <stdio.h> int n, nf;

int in[100];

int p[50];

int hit = 0;

int i, j, k;

int pgfaultcnt = 0;

void accept() {

printf("\nEnter length of page reference:"); scanf("%d", &n);

printf("\nEnter pages:");

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

```
scanf("%d", &in[i]); printf("\nEnter no. of frames:"); scanf("%d", &nf);

}




void initialize() {

pgfaultcnt = 0;
for (i = 0; i < nf; i++)

p[i] = 9999; }

int isHit(int data) {

hit = 0;
for (j = 0; j < nf; j++) {

if (p[j] == data) {

hit = 1;
```

```
break; }

}

return hit; }

int getHitIndex(int data) {

int hitind;
for (k = 0; k < nf; k++)




{
if (p[k] == data) {

hitind = k;

break; }

}
```

```
return hitind; }

void dispPages() {

for (k = 0; k < nf; k++) {

if (p[k] != 9999) printf(" %d", p[k]);

} }

void dispPgFaultCnt() {

printf("\nTotal no of page faults:%d", pgfaultcnt); }

void fifo() {




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

```
printf("\nFrame %d :", in[i]);

if (isHit(in[i]) == 0) {

for (k = 0; k < nf - 1; k++) p[k] = p[k + 1];

p[k] = in[i]; printf("\tf"); pgfaultcnt++; dispPages();

} else

printf("\t"); }

dispPgFaultCnt(); }

void lru() {

initialize();




int least[50];
for (i = 0; i < n; i++) {
```

```
printf("\nFrame %d :", in[i]);

if (isHit(in[i]) == 0) {

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

int pg = p[j];
int found = 0;
for (k = i - 1; k >= 0; k--) {

if (pg == in[k]) {

least[j] = k; found = 1; break;

} else

found = 0; }

if (!found)
least[j] = -9999;
```

```
}
int min = 9999;
int repindex;
for (j = 0; j < nf; j++) {

if (least[j] < min) {

min = least[j];

repindex = j; }

}
p[repindex] = in[i]; pgfaultcnt++;

dispPages(); }

else printf("\t");

}

dispPgFaultCnt(); }
```

```
int main() {

int choice; while (1)

{

printf("\nPage Replacement Algorithms\n1.Enter data\n2.FIFO\n3.LRU\n4.Exit\nEnter your choice:");

scanf("%d", &choice); switch (choice)

{

case 1:

accept();

break; case 2:

fifo();

break; case 3:
```

```
lru();

break; default:

return 0;

break; }

} }
```

OUTPUT:

```
/tmp/yl8k6Zlfrv.o
Page Replacement Algorithms
1.Enter data
2.FIFO
3.LRU
4.Exit
Enter your choice:1
Enter length of page reference:7
Enter pages:1
3
0
3
5
6
3
Enter no. of frames:3
Page Replacement Algorithms
1.Enter data
2.FIFO
3.LRU
4.Exit
Enter your choice:2
Frame 1 :   f 1
Frame 3 :   f 1 3
Frame 0 :   f 1 3 0
Frame 3 :
Frame 5 :   f 3 0 5
Frame 6 :   f 0 5 6
Frame 3 :   f 5 6 3
Total no of page faults:6
Page Replacement Algorithms
1.Enter data
2.FIFO
3.LRU
4.Exit
Enter your choice:3
Frame 1 : 1
Frame 3 : 1 3
Frame 0 : 1 3 0
Frame 3 :
Frame 5 : 5 3 0
Frame 6 : 5 3 6
Frame 3 :
Total no of page faults:5
Page Replacement Algorithms
1.Enter data
2.FIFO
3.LRU
4.Exit
```