Name : Omkar Arun Shinde

PRN No.: 122B1B258

Assignment No. 08

Problem Statement : Write a to implement paging replacement algorithms :

a) FCFS

b) Least Recently Used (LRU)

c) Optimal algorithm

Code :

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

#define MAX_FRAMES 10
#define MAX_PAGES 50
int n, Size;

int isHit(int Frame[], int page) {
   for (int i = 0; i < Size; i++) {
      if (Frame[i] == page)
         return 1;
   }
   return 0;
}

void FCFS(int PageSeq[]) {
   printf("\n--- FCFS Page Replacement ---\n");
   int Frame[MAX_FRAMES];
```

```
    int front = 0, faults = 0;

    for (int i = 0; i < Size; i++) {
        Frame[i] = -1;
    }

    for (int i = 0; i < n; i++) {
        if (!isHit(Frame, PageSeq[i])) {
            faults++;
            Frame[front] = PageSeq[i];
            front = (front + 1) % Size;
        }
        printf("Page %d: ", PageSeq[i]);
        for (int j = 0; j < Size; j++) {
            Frame[j] == -1 ? printf("- ") : printf("%d ", Frame[j]);
        }
        printf("\n");
    }
    printf("Total Page Faults (FCFS): %d\n", faults);
    printf("Total Page Hits (FCFS): %d\n", n - faults);
    printf("Hit Ratio: %.2f%%\n", ((float)(n - faults) / n) * 100);
}

void LRU(int PageSeq[]) {
    printf("\n--- LRU Page Replacement ---\n");
    int Frame[MAX_FRAMES];
    int count[MAX_FRAMES] = {0};
    int Time = 0, faults = 0;
```

```
for (int i = 0; i < Size; i++) {
    Frame[i] = -1;
}

for (int i = 0; i < n; i++) {
    Time++;
    int hit = 0;

    for (int j = 0; j < Size; j++) {
        if (Frame[j] == PageSeq[i]) {
            hit = 1;
            count[j] = Time;
            break;
        }
    }

    if (!hit) {
        faults++;
        int min = INT_MAX, replace_index = -1;
        for (int j = 0; j < Size; j++) {
            if (Frame[j] == -1) {
                replace_index = j;
                break;
            } else if (count[j] < min) {
                min = count[j];
                replace_index = j;
            }
```

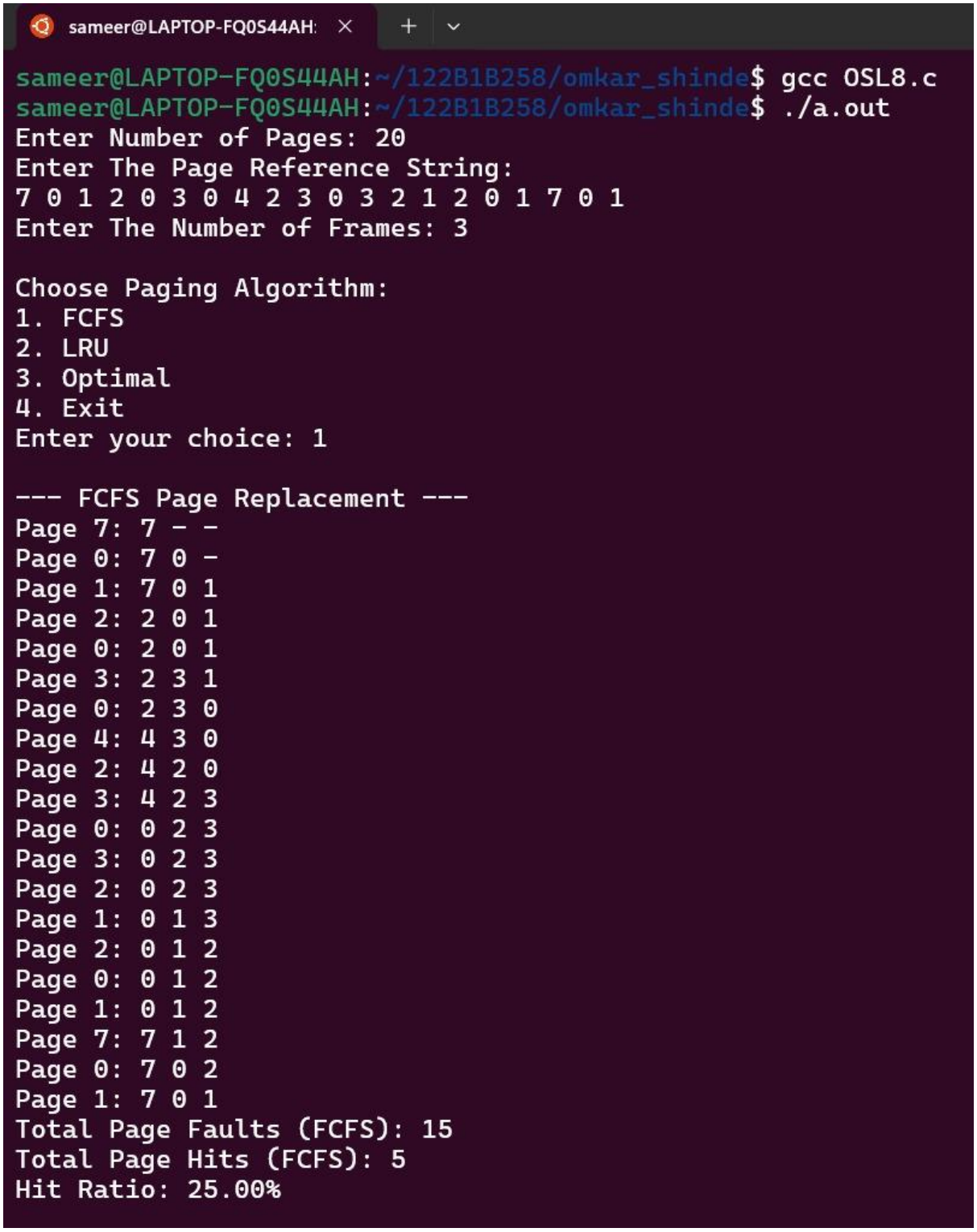
```
            }
            Frame[replace_index] = PageSeq[i];
            count[replace_index] = Time;
        }

        printf("Page %d: ", PageSeq[i]);
        for (int j = 0; j < Size; j++) {
            Frame[j] == -1 ? printf("- ") : printf("%d ", Frame[j]);
        }
        printf("\n");
    }
    printf("Total Page Faults (LRU): %d\n", faults);
    printf("Total Page Hits (LRU): %d\n", n - faults);
    printf("Hit Ratio: %.2f%%\n", ((float)(n - faults) / n) * 100);
}

int predict(int PageSeq[], int Frame[], int index) {
    int Far = -1, Found = -1;
    for (int i = 0; i < Size; i++) {
        int j;
        for (j = index; j < n; j++) {
            if (Frame[i] == PageSeq[j]) {
                if (j > Far) {
                    Far = j;
                    Found = i;
                }
                break;
            }
```

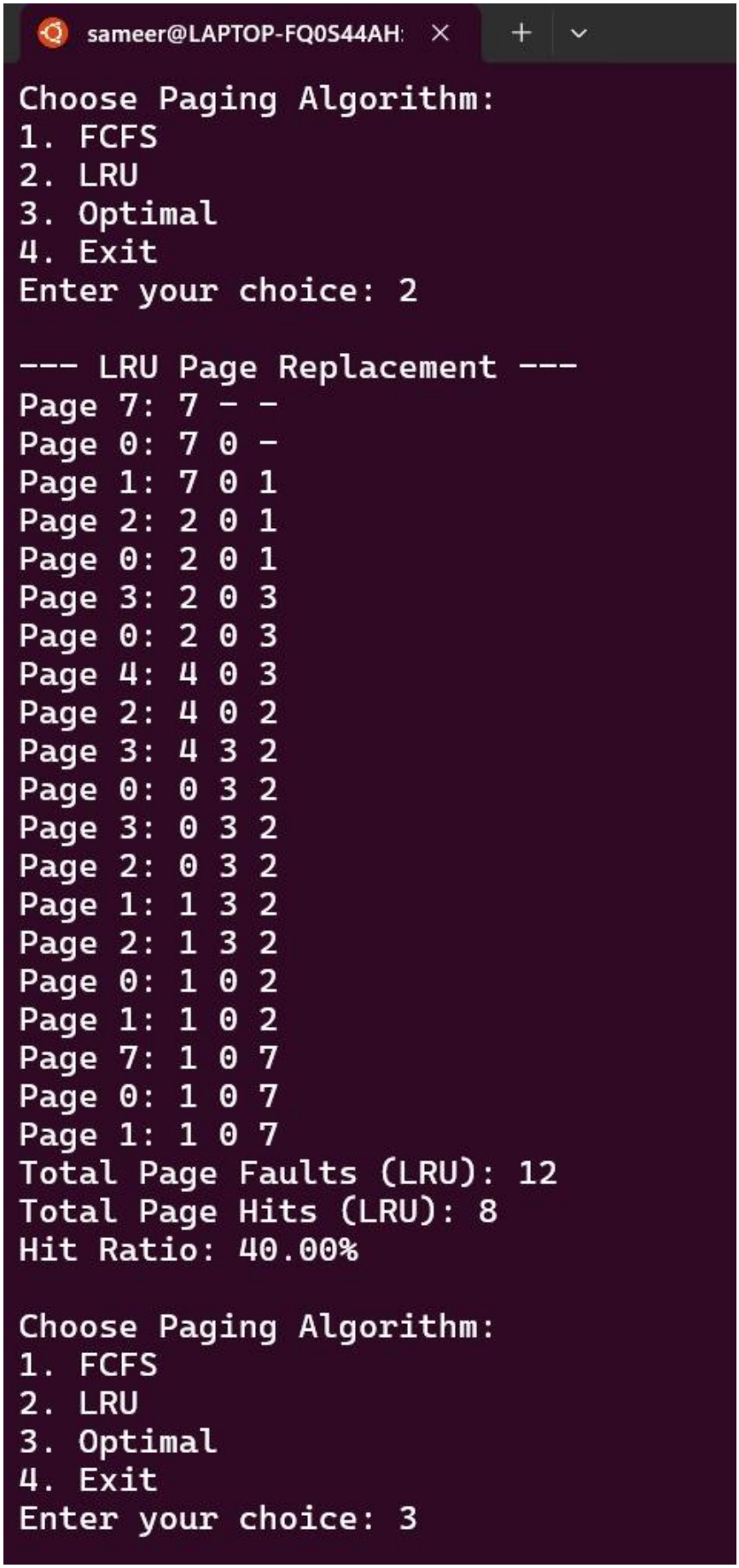
```
        }
        if (j == n)
            return i;
    }
    return (Found == -1) ? 0 : Found;
}

void Optimal(int PageSeq[]) {
    printf("\n--- Optimal Page Replacement ---\n");
    int Frame[MAX_FRAMES];
    int faults = 0;

    for (int i = 0; i < Size; i++) {
        Frame[i] = -1;
    }

    for (int i = 0; i < n; i++) {
        if (!isHit(Frame, PageSeq[i])) {
            faults++;
            int j;
            for (j = 0; j < Size; j++) {
                if (Frame[j] == -1) {
                    Frame[j] = PageSeq[i];
                    break;
                }
            }

            if (j == Size) {
```

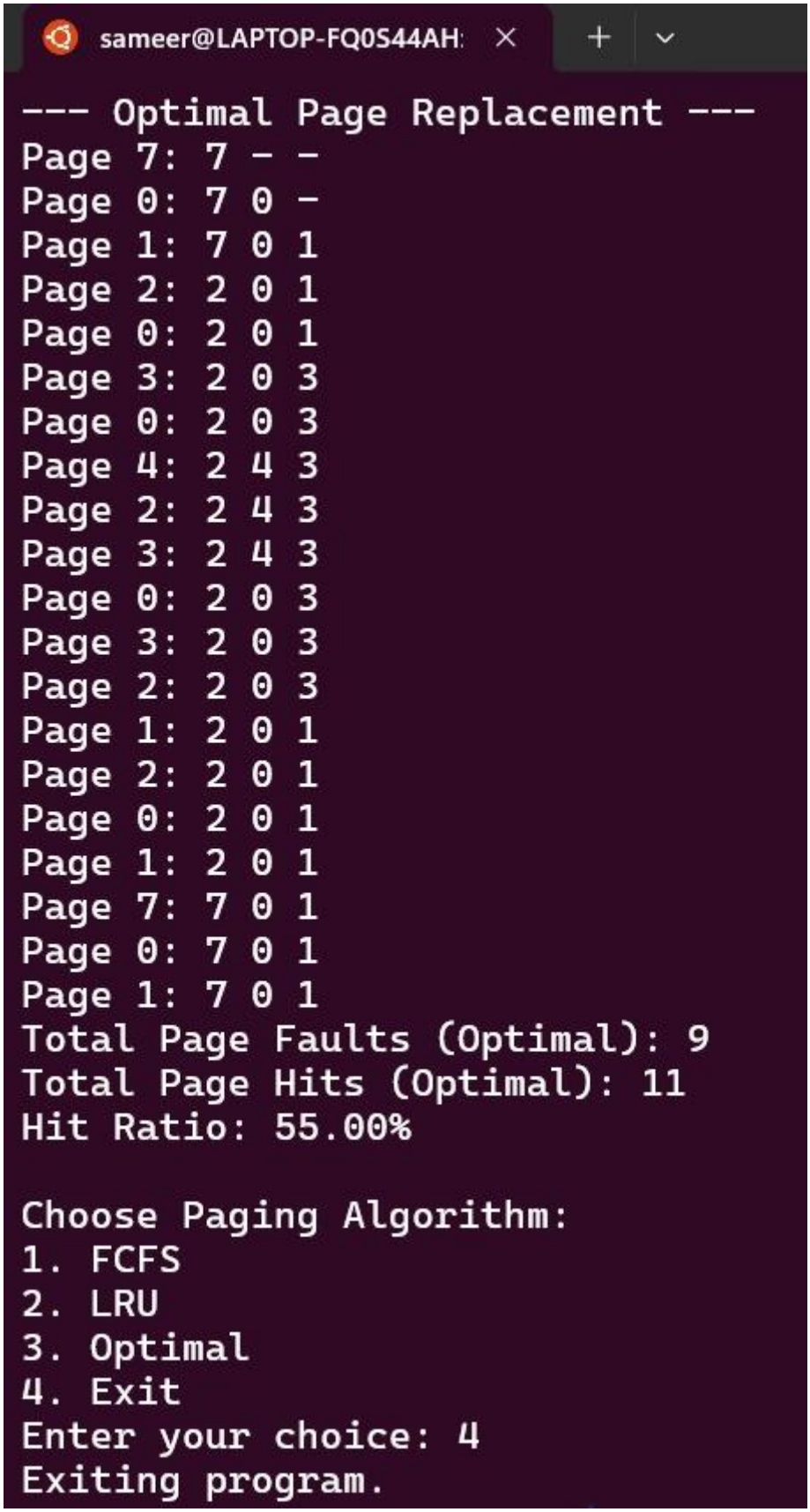
```
            int idx = predict(PageSeq, Frame, i + 1);
            Frame[idx] = PageSeq[i];
        }
    }

    printf("Page %d: ", PageSeq[i]);
    for (int j = 0; j < Size; j++) {
        Frame[j] == -1 ? printf("- ") : printf("%d ", Frame[j]);
    }
    printf("\n");
  }
  printf("Total Page Faults (Optimal): %d\n", faults);
  printf("Total Page Hits (Optimal): %d\n", n - faults);
  printf("Hit Ratio: %.2f%%\n", ((float)(n - faults) / n) * 100);
}

int main() {
  int PageSeq[MAX_PAGES], choice;

  printf("Enter Number of Pages: ");
  scanf("%d", &n);
  printf("Enter The Page Reference String:\n");
  for (int i = 0; i < n; i++)
    scanf("%d", &PageSeq[i]);
  printf("Enter The Number of Frames: ");
  scanf("%d", &Size);

  do {
```

```
        printf("\nChoose Paging Algorithm:\n");
        printf("1. FCFS\n");
        printf("2. LRU\n");
        printf("3. Optimal\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                FCFS(PageSeq);
                break;
            case 2:
                LRU(PageSeq);
                break;
            case 3:
                Optimal(PageSeq);
                break;
            case 4:
                printf("Exiting program.\n");
                break;
            default:
                printf("Invalid choice! Try again.\n");
        }
    } while (choice != 4);

    return 0;
}
```

Output :


sameer@LAPTOP-FQ0S44AH:
sameer@LAPTOP-FQ0S44AH:~/122B1B258/omkar_shinde$ gcc OSL8.c
sameer@LAPTOP-FQ0S44AH:~/122B1B258/omkar_shinde$ ./a.out
Enter Number of Pages: 20
Enter The Page Reference String:
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
Enter The Number of Frames: 3
Choose Paging Algorithm:
1. FCFS
2. LRU
3. Optimal
4. Exit
Enter your choice: 1
--- FCFS Page Replacement ---
Page 7: 7 - -
Page 0: 7 0 -
Page 1: 7 0 1
Page 2: 2 0 1
Page 0: 2 0 1
Page 3: 2 3 1
Page 0: 2 3 0
Page 4: 4 3 0
Page 2: 4 2 0
Page 3: 4 2 3
Page 0: 0 2 3
Page 3: 0 2 3
Page 2: 0 2 3
Page 1: 0 1 3
Page 2: 0 1 2
Page 0: 0 1 2
Page 1: 0 1 2
Page 7: 7 1 2
Page 0: 7 0 2
Page 1: 7 0 1
Total Page Faults (FCFS): 15
Total Page Hits (FCFS): 5
Hit Ratio: 25.00%


sameer@LAPTOP-FQ0S44AH:
Choose Paging Algorithm:
1. FCFS
2. LRU
3. Optimal
4. Exit
Enter your choice: 2
--- LRU Page Replacement ---
Page 7: 7 - -
Page 0: 7 0 -
Page 1: 7 0 1
Page 2: 2 0 1
Page 0: 2 0 1
Page 3: 2 0 3
Page 0: 2 0 3
Page 4: 4 0 3
Page 2: 4 0 2
Page 3: 4 3 2
Page 0: 0 3 2
Page 3: 0 3 2
Page 2: 0 3 2
Page 1: 1 3 2
Page 2: 1 3 2
Page 0: 1 0 2
Page 1: 1 0 2
Page 7: 1 0 7
Page 0: 1 0 7
Page 1: 1 0 7
Total Page Faults (LRU): 12
Total Page Hits (LRU): 8
Hit Ratio: 40.00%
Choose Paging Algorithm:
1. FCFS
2. LRU
3. Optimal
4. Exit
Enter your choice: 3


sameer@LAPTOP-FQ0S44AH:
--- Optimal Page Replacement ---
Page 7: 7 - -
Page 0: 7 0 -
Page 1: 7 0 1
Page 2: 2 0 1
Page 0: 2 0 1
Page 3: 2 0 3
Page 0: 2 0 3
Page 4: 2 4 3
Page 2: 2 4 3
Page 3: 2 4 3
Page 0: 2 0 3
Page 3: 2 0 3
Page 2: 2 0 3
Page 1: 2 0 1
Page 2: 2 0 1
Page 0: 2 0 1
Page 1: 2 0 1
Page 7: 7 0 1
Page 0: 7 0 1
Page 1: 7 0 1
Total Page Faults (Optimal): 9
Total Page Hits (Optimal): 11
Hit Ratio: 55.00%
Choose Paging Algorithm:
1. FCFS
2. LRU
3. Optimal
4. Exit
Enter your choice: 4
Exiting program.