Name: Aakash A. Joshi
Class: TE computer
Roll no.: 0077
Subject: SPOS
Batch: T4

Assignment no. 7.1(FIFO)

Code:

```
import java.util.HashSet;
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

public class fifo {
   // Method to find page faults using FIFO
   static int pageFaults(int pages[], int n, int capacity) {
      HashSet<Integer> s = new HashSet<>(capacity);
      Queue<Integer> indexes = new LinkedList<>();
      int page_faults = 0;

      for (int i = 0; i < n; i++) {
         if (s.size() < capacity) {
            if (!s.contains(pages[i])) {
               s.add(pages[i]);
               page_faults++;
               indexes.add(pages[i]);
            }
         } else {
            if (!s.contains(pages[i])) {
               int val = indexes.peek();
               indexes.poll();
               s.remove(val);
               s.add(pages[i]);
               indexes.add(pages[i]);
               page_faults++;
            }
         }

         // Print page access and frames
         System.out.print("Page " + pages[i] + " -> ");
         for (int page : indexes) {
            System.out.print(page + " ");
         }
         System.out.println();
      }

      return page_faults;
   }

   public static void main(String args[]) {
```

```
        Scanner input = new Scanner(System.in);

        System.out.print("Enter the number of pages: ");
        int n = input.nextInt();
        int pages[] = new int[n];

        System.out.println("Enter the page numbers:");
        for (int i = 0; i < n; i++) {
            pages[i] = input.nextInt();
        }

        System.out.print("Enter the number of frames: ");
        int capacity = input.nextInt();

        input.close();

        int pageFaults = pageFaults(pages, n, capacity);
        System.out.println("\nTotal Page Faults: " + pageFaults);
    }
}
```

Output:

```
Enter the number of pages: 14
Enter the page numbers:
7
0
1
2
0
3
0
4
2
3
0
3
2
1
Enter the number of frames: 3
Page 7 -> 7
Page 0 -> 7 0
Page 1 -> 7 0 1
Page 2 -> 0 1 2
Page 0 -> 0 1 2
Page 3 -> 1 2 3
Page 0 -> 2 3 0
Page 4 -> 3 0 4
Page 2 -> 0 4 2
Page 3 -> 4 2 3
Page 0 -> 2 3 0
Page 3 -> 2 3 0
Page 2 -> 2 3 0
Page 1 -> 3 0 1
```

Total Page Faults: 11

Name: Aakash A. Joshi
Class: TE computer
Roll no.: 0077
Subject: SPOS
Batch: T4

Assignment no. 7.2(LRU)

Code:

```
import java.io.*;
public class lru{
      public static void main(String args[])throws IOException
      {
                BufferedReader obj=new BufferedReader(new
InputStreamReader(System.in));
                int f,page=0,ch,pgf=0,n,chn=0;
                boolean flag;
                int pages[];          //pgf-page fault

               System.out.println("LRU");
               int pt=0;
         System.out.println("enter no. of frames: ");
                f=Integer.parseInt(obj.readLine());
               int frame[]=new int[f];


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

               System.out.println("enter the no of pages ");
               n=Integer.parseInt(obj.readLine());

              pages=new int[n];
               System.out.println("enter the page no ");

               for(int j=0;j<n;j++)
               pages[j]=Integer.parseInt(obj.readLine());

               int pg=0;
               for(pg=0;pg<n;pg++)
               {
                       page=pages[pg];
                       flag=true;
                       for(int j=0;j<f;j++)
                       {
                               if(page==frame[j])
                               {
                                       flag=false;
```

```
                        break;
                    }
                }
                int temp,h=3,i;
                if(flag)
            {
                if( frame[1]!=-1 && frame[2]!=-1 && frame[0]!=-1)
                    {
                        temp=pages[pg-3];
                        if(temp==pages[pg-2] || temp==pages[pg-1])
                            temp=pages[pg-4];

                        for(i=0;i<f;i++)
                            if(temp==frame[i])
                                break;
                        frame[i]=pages[pg];
                    }
                    else
                    {
                        if(frame[0]==-1)
                            frame[0]=pages[pg];
                        else if(frame[1]==-1)
                            frame[1]=pages[pg];
                        else if(frame[2]==-1)
                            frame[2]=pages[pg];
                    }

                    System.out.print("frame :");
                    for(int j=0;j<f;j++)
                    System.out.print(frame[j]+"  ");
                    System.out.println();
                    pgf++;
                }
                else
                {
                    System.out.print("frame :");
                    for(int j=0;j<f;j++)
                    System.out.print(frame[j]+"  ");
                    System.out.println();
                }

            }//for

        System.out.println("Page fault:"+pgf);

    }
}
```

Output:
LRU
enter no. of frames:

```
3
enter the no of pages
8
enter the page no
3
1
2
1
6
5
1
3
frame :3   -1   -1
frame :3    1   -1
frame :3    1    2
frame :3    1    2
frame :6    1    2
frame :6    1    5
frame :6    1    5
frame :3    1    5
Page fault:6
```

Name: Aakash A. Joshi
Class: TE computer
Roll no.: 0077
Subject: SPOS
Batch: T4

Assignment no. 7.3(Optimal)

Code:

```
import java.util.Scanner;

public class optimal {

   static boolean search(int key, int[] fr) {
      for (int i = 0; i < fr.length; i++)
         if (fr[i] == key)
            return true;
      return false;
   }

   static int predict(int pg[], int[] fr, int pn, int index) {
      int res = -1, farthest = index;
      for (int i = 0; i < fr.length; i++) {
         int j;
         for (j = index; j < pn; j++) {
            if (fr[i] == pg[j]) {
               if (j > farthest) {
                  farthest = j;
                  res = i;
               }
               break;
            }
         }

         if (j == pn)
            return i;
      }

      return (res == -1) ? 0 : res;
   }

   static void optimalPage(int pn, int fn) {
   Scanner scanner = new Scanner(System.in);

   int[] fr = new int[fn];
   int[] pg = new int[pn];
   int hit = 0;
   int index = 0;

   System.out.println("Enter page numbers:");
   for (int i = 0; i < pn; i++) {
```

```
            pg[i] = scanner.nextInt();
        }

        System.out.println("No. of frames: " + fn);

        System.out.println("\nPage Simulation:");
        System.out.println("----------");

        for (int i = 0; i < pn; i++) {
            System.out.print("Page " + pg[i] + " -> ");
            if (search(pg[i], fr)) {
                hit++;
                System.out.println("Hit");
                continue;
            }

            if (index < fn) {
                fr[index++] = pg[i];
                System.out.print("Miss [");
            } else {
                int j = predict(pg, fr, pn, i + 1);
                fr[j] = pg[i];
                System.out.print("Miss [");
            }

            // Print the current frames
            for (int k = 0; k < fn; k++) {
                if (k != 0) {
                    System.out.print(", ");
                }
                System.out.print(fr[k]);
            }
            System.out.println("]");
        }

        System.out.println("\nNo. of hits = " + hit);
        System.out.println("No. of misses = " + (pn - hit));
        scanner.close();
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the number of pages:");
        int pn = scanner.nextInt();

        System.out.println("Enter the number of frames:");
        int fn = scanner.nextInt();

        optimalPage(pn, fn);
        scanner.close();
    }
```

```
}
```

Output:

```
Enter the number of pages:
10
Enter the number of frames:
3
Enter page numbers:
4
7
6
1
7
6
1
2
7
2
No. of frames: 3

Page Simulation:
----------
Page 4 -> Miss [4, 0, 0]
Page 7 -> Miss [4, 7, 0]
Page 6 -> Miss [4, 7, 6]
Page 1 -> Miss [1, 7, 6]
Page 7 -> Hit
Page 6 -> Hit
Page 1 -> Hit
Page 2 -> Miss [2, 7, 6]
Page 7 -> Hit
Page 2 -> Hit

No. of hits = 5
No. of misses = 5
```