

LAB ASSIGNMENT – 5

1) FIFO Page Replacement Program:-

```
package PageReplacement;
import java.io.*;
public class FIFO {
    public static void main(String[] args) throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int frames, pointer = 0, hit = 0, fault = 0, ref_len;
        int buffer[];
        int reference[];
        int mem_layout[][];

        System.out.println("Please enter the number of Frames: ");
        frames = Integer.parseInt(br.readLine());

        System.out.println("Please enter the length of the Reference string: ");
        ref_len = Integer.parseInt(br.readLine());

        reference = new int[ref_len];
        mem_layout = new int[ref_len][frames];
        buffer = new int[frames];
        for(int j = 0; j < frames; j++)
            buffer[j] = -1;

        System.out.println("Please enter the reference string: ");
        for(int i = 0; i < ref_len; i++)
        {
            reference[i] = Integer.parseInt(br.readLine());
        }
        System.out.println();
        for(int i = 0; i < ref_len; i++)
        {
            int search = -1;
            for(int j = 0; j < frames; j++)
            {
                if(buffer[j] == reference[i])
                {
                    search = j;
                    hit++;
                    break;
                }
            }
            if(search == -1)
            {
                buffer[pointer] = reference[i];
                fault++;
                pointer++;
                if(pointer == frames)
                    pointer = 0;
            }
            for(int j = 0; j < frames; j++)
                mem_layout[i][j] = buffer[j];
        }
        for(int i = 0; i < frames; i++)
        {
            for(int j = 0; j < ref_len; j++)
                System.out.printf("%3d ", mem_layout[j][i]);
            System.out.println();
        }
        System.out.println("The number of Hits: " + hit);
        System.out.println("Hit Ratio: " + (float)((float)hit/ref_len));
        System.out.println("The number of Faults: " + fault);
    }
}
```

2) LRU Page Replacement Program:-

```
package PageReplacement;
import java.io.*;
import java.util.*;

public class LRU {

    public static void main(String[] args) throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int frames, pointer = 0, hit = 0, fault = 0, ref_len;
        Boolean isFull = false;
        int buffer[];
        ArrayList<Integer> stack = new ArrayList<Integer>();
        int reference[];
        int mem_layout[][];

        System.out.println("Please enter the number of Frames: ");
        frames = Integer.parseInt(br.readLine());

        System.out.println("Please enter the length of the Reference string: ");
        ref_len = Integer.parseInt(br.readLine());

        reference = new int[ref_len];
        mem_layout = new int[ref_len][frames];
        buffer = new int[frames];
        for(int j = 0; j < frames; j++)
            buffer[j] = -1;

        System.out.println("Please enter the reference string: ");
        for(int i = 0; i < ref_len; i++)
        {
            reference[i] = Integer.parseInt(br.readLine());
        }
        System.out.println();
        for(int i = 0; i < ref_len; i++)
        {
            if(stack.contains(reference[i]))
            {
                stack.remove(stack.indexOf(reference[i]));
            }
            stack.add(reference[i]);
            int search = -1;
            for(int j = 0; j < frames; j++)
            {
                if(buffer[j] == reference[i])
                {
                    search = j;
                    hit++;
                    break;
                }
            }
            if(search == -1)
            {
                if(isFull)
                {
                    int min_loc = ref_len;
                    for(int j = 0; j < frames; j++)
                    {
                        if(stack.contains(buffer[j]))
                        {
                            int temp = stack.indexOf(buffer[j]);
                            if(temp < min_loc)
                            {
                                min_loc = temp;
                                pointer = j;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }
}

    buffer[pointer] = reference[i];
    fault++;
    pointer++;
    if(pointer == frames)
    {
        pointer = 0;
        isFull = true;
    }
}
for(int j = 0; j < frames; j++)
    mem_layout[i][j] = buffer[j];
}

for(int i = 0; i < frames; i++)
{
    for(int j = 0; j < ref_len; j++)
        System.out.printf("%3d ", mem_layout[j][i]);
    System.out.println();
}
System.out.println("The number of Hits: " + hit);
System.out.println("Hit Ratio: " + (float)((float)hit/ref_len));
System.out.println("The number of Faults: " + fault);
}
}

```

3) OPTIMAL Page Replacement Program :-

```
package PageReplacement;
import java.io.*;
public class Optimal {
    public static void main(String[] args) throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int frames, pointer = 0, hit = 0, fault = 0, ref_len;
        boolean isFull = false;
        int buffer[];
        int reference[];
        int mem_layout[][];

        System.out.println("Please enter the number of Frames: ");
        frames = Integer.parseInt(br.readLine());

        System.out.println("Please enter the length of the Reference string: ");
        ref_len = Integer.parseInt(br.readLine());

        reference = new int[ref_len];
        mem_layout = new int[ref_len][frames];
        buffer = new int[frames];
        for(int j = 0; j < frames; j++)
            buffer[j] = -1;

        System.out.println("Please enter the reference string: ");
        for(int i = 0; i < ref_len; i++)
        {
            reference[i] = Integer.parseInt(br.readLine());
        }
        System.out.println();
        for(int i = 0; i < ref_len; i++)
        {
            int search = -1;
            for(int j = 0; j < frames; j++)
            {
                if(buffer[j] == reference[i])
                {
                    search = j;
                    hit++;
                    break;
                }
            }
            if(search == -1)
            {
                if(isFull)
                {
                    int index[] = new int[frames];
                    boolean index_flag[] = new boolean[frames];
                    for(int j = i + 1; j < ref_len; j++)
                    {
                        for(int k = 0; k < frames; k++)
                        {
                            if((reference[j] == buffer[k]) && (index_flag[k] == false))
                            {
                                index[k] = j;
                                index_flag[k] = true;
                                break;
                            }
                        }
                    }
                }
                int max = index[0];
                pointer = 0;
                if(max == 0)
                    max = 200;
                for(int j = 0; j < frames; j++)
```

```

        {
            if(index[j] == 0)
                index[j] = 200;
            if(index[j] > max)
            {
                max = index[j];
                pointer = j;
            }
        }
    }
    buffer[pointer] = reference[i];
    fault++;
    if(!isFull)
    {
        pointer++;
        if(pointer == frames)
        {
            pointer = 0;
            isFull = true;
        }
    }
    for(int j = 0; j < frames; j++)
        mem_layout[i][j] = buffer[j];
}

for(int i = 0; i < frames; i++){
    for(int j = 0; j < ref_len; j++)
        System.out.printf("%3d ",mem_layout[j][i]);
    System.out.println();
}

System.out.println("The number of Hits: " + hit);
System.out.println("Hit Ratio: " + (float)((float)hit/ref_len));
System.out.println("The number of Faults: " + fault);
}
}

```

OUTPUT

Please enter the number of Frames: 3

Please enter the length of the Reference string: 18

Please enter the reference string:

1
2
3
4
2
1
5
6
1
2
3
7
6
3
2
1
3
6

1	1	1	4	4	4	4	6	6	6	6	7	7	7	7	1	1	1
-1	2	2	2	2	1	1	1	1	2	2	2	6	6	6	6	3	3
-1	-1	3	3	3	3	5	5	5	5	3	3	3	3	2	2	2	6

The number of Hits: 3

Hit Ratio: 0.16666667

The number of Faults: 15

Output 1 : - FIFO

```

Please enter the number of Frames: 3
Please enter the length of the Reference string: 18
Please enter the reference string:
1
2
3
4
2
1
5
6
1
2
3
7
6
3
2
1
3
6

```

```

    1  1  1  4  4  4  5  5  5  2  2  2  6  6  6  1  1  1
   -1  2  2  2  2  2  2  6  6  6  3  3  3  3  3  3  3  3
   -1 -1  3  3  3  1  1  1  1  1  1  7  7  7  2  2  2  6

```

```

The number of Hits: 4
Hit Ratio: 0.22222222
The number of Faults: 14

```

Output :- LRU

Please enter the number of Frames: 3
Please enter the length of the Reference string: 18
Please enter the reference string:

1
2
3
4
2
1
5
6
1
2
3
7
6
3
2
1
3
6

1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3
-1	2	2	2	2	2	2	2	2	2	2	7	7	7	2	1	1	1
-1	-1	3	4	4	4	5	6	6	6	6	6	6	6	6	6	6	6

The number of Hits: 8
Hit Ratio: 0.44444445
The number of Faults: 10

Output 3 : Optimal