

## FIFO PAGE REPLACEMENT :

```
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);
}
}

```

### output:-

```

Please enter the number of Frames:
3
Please enter the length of the Reference string:
20
Please enter the reference string:
7
0
1
2
0
3
0
4
2
3
0
3
2
1
2
0
1
7
0
1

```

7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7
7	7																
-1	0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1
0	0																
-1	-1	1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2
2	1																

```

The number of Hits: 5
Hit Ratio: 0.25
The number of Faults: 15
-----

```

# LRU Page Replacement algorithm in java

## code in Java:

```
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);
}

```

### output:-

```

Please enter the number of Frames:
3
Please enter the length of the Reference string:
20
Please enter the reference string:
7
0
1
2

```

```

0
3
0
4
2
3
0
3
2
1
2
0
1
7
0
1
  7   7   7   2   2   2   2   4   4   4   0   0   0   1   1   1   1   1   1
1
-1   0   0   0   0   0   0   0   0   0   3   3   3   3   3   3   0   0   0   0
0
-1  -1   1   1   1   3   3   3   2   2   2   2   2   2   2   2   2   7   7
7
The number of Hits: 8
Hit Ratio: 0.4
The number of Faults: 12
-----

```

# Optimal Page Replacement algorithm in java

## code in Java:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class OptimalReplacement {

    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:
20
```

20  
Please enter the reference string:

1 1 1 1 1 1 1 1 6 6 6 6 6 6 6 6 6 2 4

	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	7	4
4	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

-1	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1
1	1	1	2	2	2	5	5	5	5	5	5	5	2	2	2	2	2	2	2

[illegible]

```
The number of Hits: 11
Hit Ratio: 0.55
```

```
The number of Faults: 9
-----
```