Experiment Number: 7

Problem Statement: **Pagers Algorithm**

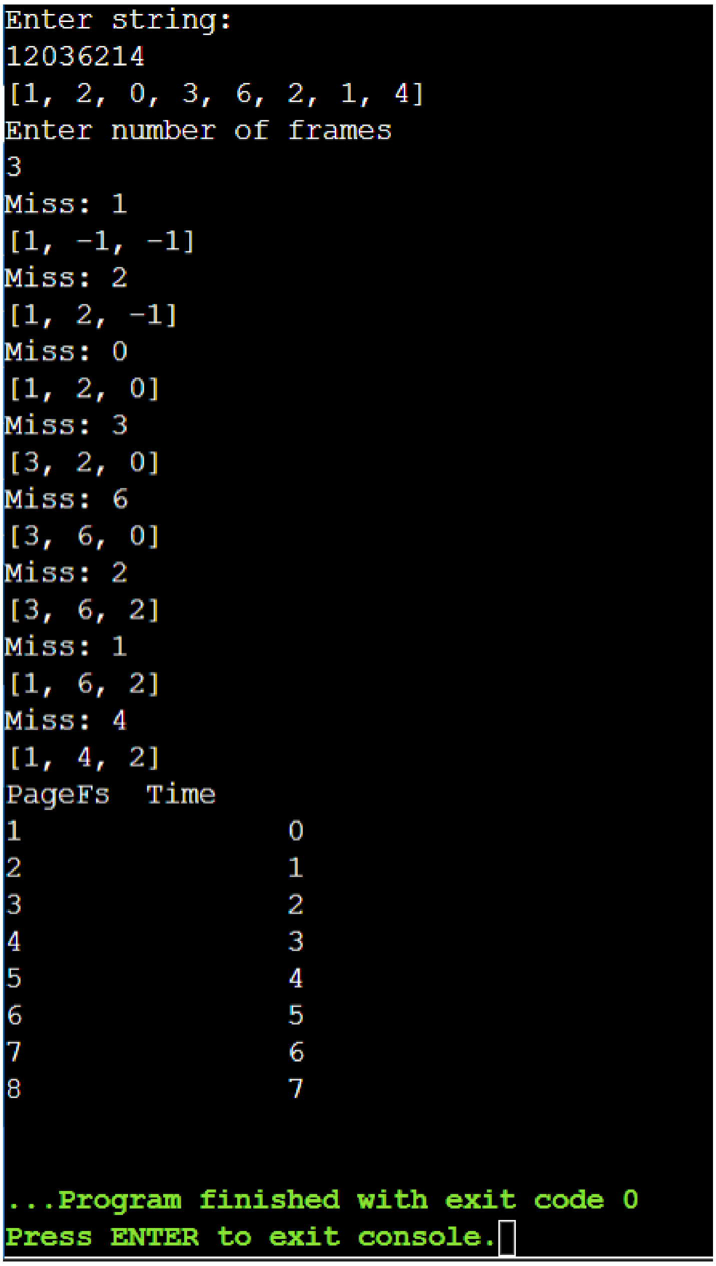
Name: Arnav Shah	Roll No. : 21
Class : AI_C	Batch : B2

1) First In First Out :-

```
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
class Replacement{
int f;
int[] runQ;
int pageF=0;
int time=0;
HashMap<Integer,Integer> map=new HashMap<>();
Replacement(int f){
this.f=f;
runQ=new int[f];
Arrays.fill(runQ,-1);
}
public void push(int m){
if(!contains(m)){
runQ[pageF%f]=m;
pageF++;
map.put(pageF,time);
System.out.println("Miss: "+m);
}else {
System.out.println("Hit: "+m);
}
time++;
```

```
System.out.println(Arrays.toString(runQ));
}
public boolean contains(int m){
for(int i=0;i< f;i++){
if(runQ[i]==m){
return true;
}
}
return false;
}
}
public class fifo {
public static void main(String[] args) {
Scanner in =new Scanner(System.in);
System.out.println("Enter string: ");
String str=in.nextLine();
int n=str.length();
int[] ref=new int[n];
for(int i=0;i<n;i++){
ref[i]=Integer.parseInt(String.valueOf(str.charAt(i)));
}
System.out.println(Arrays.toString(ref));
System.out.println("Enter number of frames");
int f= in.nextInt();
Replacement r=new Replacement(f);
for(int i=0;i<n;i++){
r.push(ref[i]);
}
System.out.println("PageFs\tTime");
for(Map.Entry<Integer,Integer> map :r.map.entrySet()){
System.out.println(map.getKey()+"\t\t"+ map.getValue());
}

}

}
```

Output :-

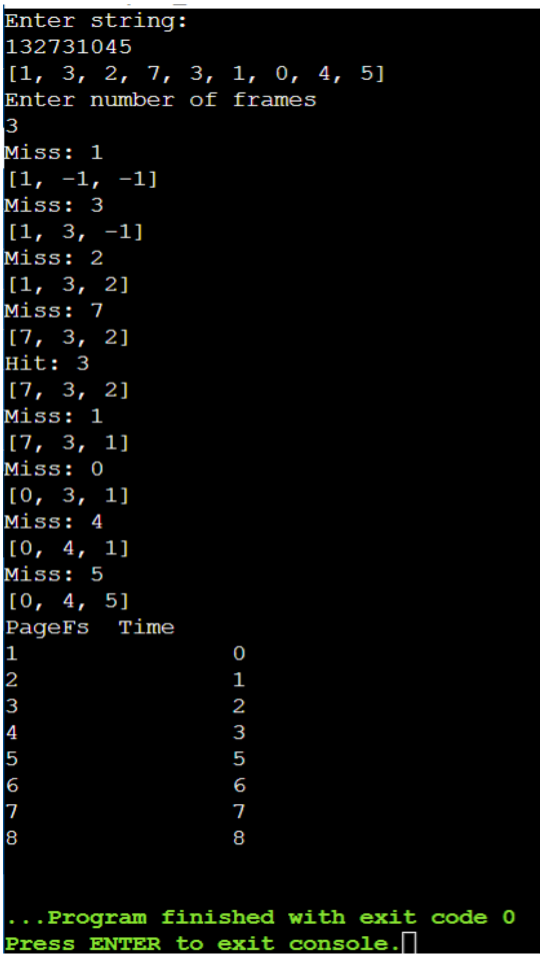


```
Enter string:
12036214
[1, 2, 0, 3, 6, 2, 1, 4]
Enter number of frames
3
Miss: 1
[1, -1, -1]
Miss: 2
[1, 2, -1]
Miss: 0
[1, 2, 0]
Miss: 3
[3, 2, 0]
Miss: 6
[3, 6, 0]
Miss: 2
[3, 6, 2]
Miss: 1
[1, 6, 2]
Miss: 4
[1, 4, 2]
PageFs  Time
1                0
2                1
3                2
4                3
5                4
6                5
7                6
8                7


...Program finished with exit code 0
Press ENTER to exit console.
```

2) LRU :-

```
import java.util.*;
class ReplacementLRU {
int f;
int[] runQ;
int pageF = 0;
int time = 0;
HashMap<Integer, Integer> map = new HashMap<>();
HashMap<Integer, Integer> page = new HashMap<>();
ReplacementLRU(int f) {
this.f = f;
runQ = new int[f];
Arrays.fill(runQ, -1);
}
public void push(int m) {
if (!contains(m)) {
if (pageF < f) {
runQ[pageF % f] = m;
pageF++;
page.put(pageF,time);
} else {
pageF++;
int lruIndex = 0;
int lruTime = map.get(runQ[0]);
for (int i = 1; i < f; i++) {
if (map.get(runQ[i]) < lruTime) {
lruTime = map.get(runQ[i]);
lruIndex = i;
}
}
runQ[lruIndex] = m;
page.put(pageF,time);
}
map.put(m, time);
System.out.println("Miss: " + m);
} else {
map.put(m, time);
System.out.println("Hit: " + m);
}
time++;
System.out.println(Arrays.toString(runQ));
}
public boolean contains(int m) {
for (int i = 0; i < f; i++) {
if (runQ[i] == m) {
return true;
}
}
return false;
}
}
public class lru {
public static void main(String[] args) {
Scanner in = new Scanner(System.in);
System.out.println("Enter string: ");
String str = in.nextLine();
int n = str.length();
int[] ref = new int[n];
for (int i = 0; i < n; i++) {
ref[i] = Integer.parseInt(String.valueOf(str.charAt(i)));
}
System.out.println(Arrays.toString(ref));
System.out.println("Enter number of frames");

int f = in.nextInt();

ReplacementLRU r = new ReplacementLRU(f);

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

r.push(ref[i]);

}

System.out.println("PageFs\tTime");

for (Map.Entry<Integer, Integer> page : r.page.entrySet()) {

System.out.println(page.getKey() + "\t\t" + page.getValue());

}

}

}
```

Output :-



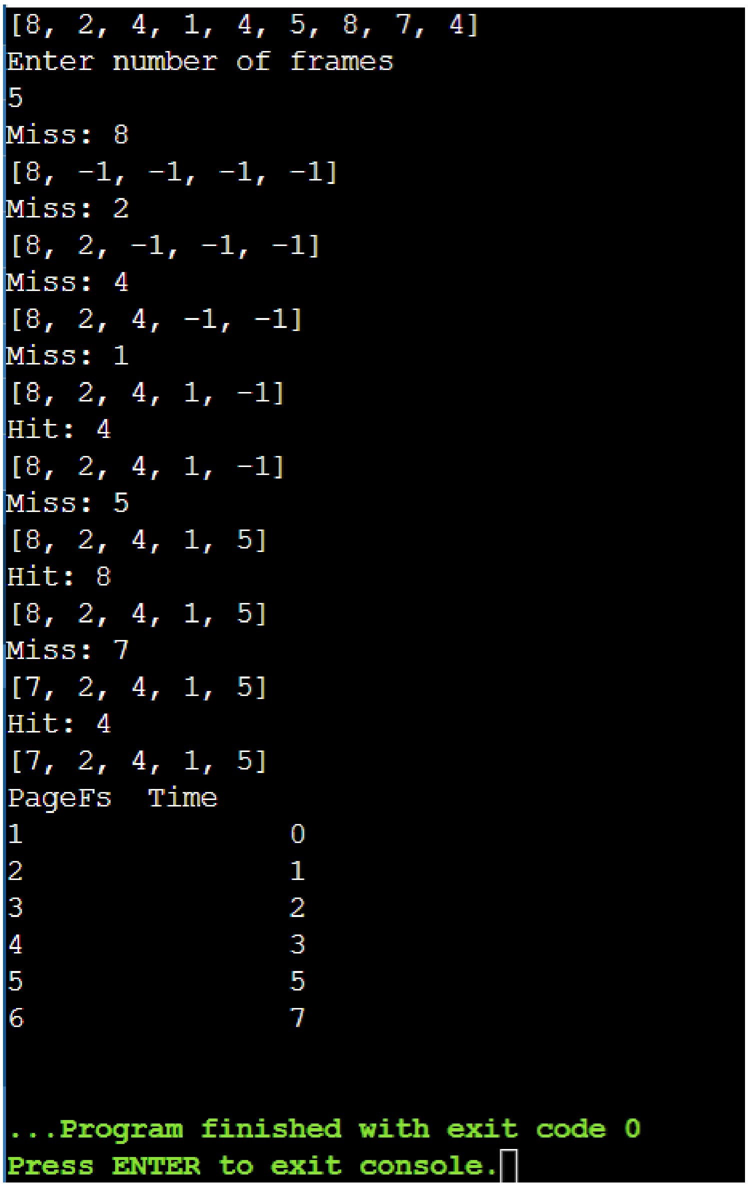
```
Enter string:
132731045
[1, 3, 2, 7, 3, 1, 0, 4, 5]
Enter number of frames
3
Miss: 1
[1, -1, -1]
Miss: 3
[1, 3, -1]
Miss: 2
[1, 3, 2]
Miss: 7
[7, 3, 2]
Hit: 3
[7, 3, 2]
Miss: 1
[7, 3, 1]
Miss: 0
[0, 3, 1]
Miss: 4
[0, 4, 1]
Miss: 5
[0, 4, 5]
PageFs  Time
1               0
2               1
3               2
4               3
5               5
6               6
7               7
8               8


...Program finished with exit code 0
Press ENTER to exit console.
```

3) Optimal :-

```
import java.util.*;
class ReplacementOptimal {
int f;
int[] runQ;
int pageF = 0;
int time = 0;
HashMap<Integer, Integer> map = new HashMap<>();
ReplacementOptimal(int f) {
this.f = f;
runQ = new int[f];
Arrays.fill(runQ, -1);
}
public void push(int m, int[] futRef) {
if (!contains(m)) {
if (pageF < f) {
runQ[pageF % f] = m;
pageF++;
} else{
pageF++;
int farthest = -1;
int replaceIndex = -1;
for (int i = 0; i < f; i++) {
int currentP = runQ[i];
boolean found = false;
for (int j = time; j < futRef.length; j++) {
if (currentP == futRef[j]) {
found = true;
if (j > farthest) {
farthest = j;
replaceIndex = i;
}
break;
}
}
if (!found) {
replaceIndex = i;
break;
}
}
runQ[replaceIndex] = m;
}
map.put(pageF, time);
System.out.println("Miss: " + m);
} else {
System.out.println("Hit: " + m);
}
time++;
System.out.println(Arrays.toString(runQ));
}
public boolean contains(int m) {
for (int i = 0; i < f; i++) {
if (runQ[i] == m) {
return true;
}
}
return false;
}
}
public class optimal {
public static void main(String[] args) {
Scanner in = new Scanner(System.in);
```

```
System.out.println("Enter string: ");

String str = in.nextLine();

int n = str.length();

int[] ref = new int[n];

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

ref[i] = Integer.parseInt(String.valueOf(str.charAt(i)));

}

System.out.println(Arrays.toString(ref));

System.out.println("Enter number of frames");

int f = in.nextInt();

int[] futRef = Arrays.copyOfRange(ref, 1, n);

ReplacementOptimal r = new ReplacementOptimal(f);

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

r.push(ref[i], futRef);

}

System.out.println("PageFs\tTime");

for (Map.Entry<Integer, Integer> map : r.map.entrySet()) {

System.out.println(map.getKey() + "\t\t" + map.getValue());

}

}

}
```

Output :-



```
[8, 2, 4, 1, 4, 5, 8, 7, 4]
Enter number of frames
5
Miss: 8
[8, -1, -1, -1, -1]
Miss: 2
[8, 2, -1, -1, -1]
Miss: 4
[8, 2, 4, -1, -1]
Miss: 1
[8, 2, 4, 1, -1]
Hit: 4
[8, 2, 4, 1, -1]
Miss: 5
[8, 2, 4, 1, 5]
Hit: 8
[8, 2, 4, 1, 5]
Miss: 7
[7, 2, 4, 1, 5]
Hit: 4
[7, 2, 4, 1, 5]
PageFs  Time
1               0
2               1
3               2
4               3
5               5
6               7


...Program finished with exit code 0
Press ENTER to exit console.
```