LP-I Assignment No. 4-A

Name : Ravindra Dayaram Bagul
Roll No. : C31105
Batch : T-1

Problem Statement : Write a program to implementing paging simulation using First in first out (FIFO) page replacement algorithm

PROGRAM

```
#include<bits/stdc++.h>
using namespace std;

int present(int table_frame[], int nf, int page)
{
   for(int i=0; i<nf; i++)
      if(page == table_frame[i])
         return 1;
   return 0;
}

void printtable(int table_frame[], int nf)
{
   for(int i=0; i<nf; i++)
   {
      if(table_frame[i] == -1)
         cout<<"-- ";
      else
         cout<<" "<<table_frame[i]<<" ";
   }
   cout<<" >> ";
}

int main()
{
   //nf-number of frames
   int n,nf,i,pos=0;

   cout<<"Enter number of frames : ";
   cin>>nf;
   int table_frame[nf];
   for(i=0;i<nf;i++)
   {
      table_frame[i]=-1;
   }

   cout<<"Enter total number of page requests :";
   cin>>n;
   int pages[n];
   cout<<"\nEnter reference string :\n";
   for(i=0;i<n;i++)
   {
      cin>>pages[i];
```

```
    }

    int count1=0;
    cout<<"\nPosition of frame table after each request\n\n";
    for(i=0;i<n;i++)
    {
       cout<<"Page table after request from  "<<pages[i]<<" -> ";
       if(!present(table_frame,nf,pages[i]))
       {
          table_frame[pos] = pages[i];
          pos = (pos+1)%nf ;//considering it as a queue
          printtable(table_frame,nf);
          cout<<"Page Fault\n";
          count1++;
          continue;
       }

       printtable(table_frame,nf);
       cout<<"\n";

    }
    cout<<"\nNumber of page faults : "<<count1;
    cout<<"\nNumber of page Hit : "<<n-count1<<endl<<endl;
}
```

Output

```
Enter number of frames : 3
Enter total number of page requests :15

Enter reference string :
7 0 1 2 0 3 0 4 2 3 0 3 1 2 0

Position of frame table after each request

Page table after request from  7 ->  7 -- --  >> Page Fault
Page table after request from  0 ->  7  0 --  >> Page Fault
Page table after request from  1 ->  7  0  1  >> Page Fault
Page table after request from  2 ->  2  0  1  >> Page Fault
Page table after request from  0 ->  2  0  1  >>
Page table after request from  3 ->  2  3  1  >> Page Fault
Page table after request from  0 ->  2  3  0  >> Page Fault
Page table after request from  4 ->  4  3  0  >> Page Fault
Page table after request from  2 ->  4  2  0  >> Page Fault
Page table after request from  3 ->  4  2  3  >> Page Fault
Page table after request from  0 ->  0  2  3  >> Page Fault
Page table after request from  3 ->  0  2  3  >>
Page table after request from  1 ->  0  1  3  >> Page Fault
Page table after request from  2 ->  0  1  2  >> Page Fault
Page table after request from  0 ->  0  1  2  >>

Number of page faults : 12
Number of page Hit : 3
```

LP-I Assignment No. 4-B

Name : Ravindra Dayaram Bagul
Roll No. : C31105
Batch : T-1

Problem Statement : Write a program to implementing paging simulation using Least Recently used (LRU) page replacement algorithm.

PROGRAM

```
#include<bits/stdc++.h>
using namespace std;
int present(int table_frame[], int nf, int page)
{
   for(int i=0; i<nf; i++)
      if(page == table_frame[i])
         return 1;
   return 0;
}

void printtable(int table_frame[], int nf)
{
   for(int i=0; i<nf; i++)
   {
      if(table_frame[i] == -1)
         cout<<"-- ";
      else
         cout<<" "<<table_frame[i]<<" ";
   }
   cout<<" >> ";
}


int findpos(int table_frame[], int nf, int pages[], int curr, int np)
{
   for(int i=0; i<nf; i++)
      if(table_frame[i] == -1)
         return i;

   int pos[nf] = {0};
   for(int i=0; i<nf; i++)
   {
      pos[i] = -1e9;
      for(int j=curr-1; j>=0; j--)
         if(pages[j] == table_frame[i])
         {
            pos[i] = j;
            break;
         }
   }

   int min1 = 1000000, retPos = -1;
   for(int i=0; i<nf; i++)
```

```
        if(min1 > pos[i])
        {
            min1 = pos[i];
            retPos = i;
        }

    return retPos;
}

int main()
{
    //nf-number of frames
    int n,nf,i,pos=0;

    cout<<"Enter number of frames : ";
    cin>>nf;
    int table_frame[nf];
    for(i=0;i<nf;i++)
    {
        table_frame[i]=-1;
    }

    cout<<"Enter total number of page requests : ";
    cin>>n;
    int pages[n];
    cout<<"Enter pages : ";
    for(i=0;i<n;i++)
    {
        cin>>pages[i];
    }

    int count1=0;
    cout<<"\nPosition of frame table after each request : \n\n";
    for(i=0;i<n;i++)
    {
        cout<<"Page table after request from "<<pages[i]<<" -> ";
        if(!present(table_frame,nf,pages[i]))
        {
            int pos = findpos(table_frame,nf,pages,i,n);
            table_frame[pos]=pages[i];

            printtable(table_frame,nf);
            cout<<"page fault\n";
            count1++;
            continue;
        }
        printtable(table_frame,nf);
        cout<<"\n";

    }
    cout<<"\nNumber of page faults : "<< count1<<endl;
    cout<<"Number of page Hit : "<<n-count1<<endl<<endl;
}
```

Output

```
Enter number of frames : 4
Enter total number of page requests : 20
Enter pages : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Position of frame table after each request :

Page table after request from 7 ->  7 -- -- --  >> page fault
Page table after request from 0 ->  7  0 -- --  >> page fault
Page table after request from 1 ->  7  0  1 --  >> page fault
Page table after request from 2 ->  7  0  1  2  >> page fault
Page table after request from 0 ->  7  0  1  2  >>
Page table after request from 3 ->  3  0  1  2  >> page fault
Page table after request from 0 ->  3  0  1  2  >>
Page table after request from 4 ->  3  0  4  2  >> page fault
Page table after request from 2 ->  3  0  4  2  >>
Page table after request from 3 ->  3  0  4  2  >>
Page table after request from 0 ->  3  0  4  2  >>
Page table after request from 3 ->  3  0  4  2  >>
Page table after request from 2 ->  3  0  4  2  >>
Page table after request from 1 ->  3  0  1  2  >> page fault
Page table after request from 2 ->  3  0  1  2  >>
Page table after request from 0 ->  3  0  1  2  >>
Page table after request from 1 ->  3  0  1  2  >>
Page table after request from 7 ->  7  0  1  2  >> page fault
Page table after request from 0 ->  7  0  1  2  >>
Page table after request from 1 ->  7  0  1  2  >>

Number of page faults : 8
Number of page Hit : 12
```

LP-I Assignment No. 4-C

Name : Ravindra Dayaram Bagul
Roll No. : C31105
Batch : T-1

Problem Statement : Write a program to implementing paging simulation using Optimal page replacement algorithm.

PROGRAM

```
#include<bits/stdc++.h>
using namespace std;

int present(int table_frame[], int nf, int page)
{
   for(int i=0; i<nf; i++)
      if(page == table_frame[i])
         return 1;
   return 0;
}

void printtable(int table_frame[], int nf)
{
   for(int i=0; i<nf; i++)
   {
      if(table_frame[i] == -1)
         cout<<"-- ";
      else
         cout<<" "<<table_frame[i]<<" ";
   }
   cout<<" >> ";
}

int findpos(int table_frame[],int nf,int pages[],int curr,int np)
{
   int i,j;
   for(i=0;i<nf;i++)
   {
      if(table_frame[i] == -1)
         return i;
   }

   int pos[nf]={0};
   for(i=0;i<nf;i++)
   {
      pos[i]=1e9;
      for(j=curr+1;j<np;j++)
      {
         if(pages[j]==table_frame[i])
         {
            pos[i]=j;
            break;
         }
```

```
        }
    }

    int max1=-1;
    int returnpos=-1;
    for(i=0;i<nf;i++)
    {
        if(pos[i]>max1)
        {
            max1=pos[i];
            returnpos=i;
        }
    }

    return returnpos;
}

int main()
{
    //nf-number of frames
    int n,nf,i,pos=0;

    cout<<"\nEnter number of frames : ";
    cin>>nf;
    int table_frame[nf];
    for(i=0;i<nf;i++)
    {
        table_frame[i]=-1;
    }

    cout<<"Enter total number of page requests : ";
    cin>>n;
    int pages[n];
    cout<<"Enter pages : ";
    for(i=0;i<n;i++)
    {
        cin>>pages[i];
    }

    int count1=0;
    cout<<"Position of frame table after each request : \n";
    for(i=0;i<n;i++)
    {
        cout<<"Page table after request from "<< pages[i]<<" -> ";
        if(!present(table_frame,nf,pages[i]))
        {
            int pos = findpos(table_frame,nf,pages,i,n);
            table_frame[pos]=pages[i];

            printtable(table_frame,nf);
            cout<<"page fault\n";
            count1++;
            continue;
```

```
        }
        printtable(table_frame,nf);
        cout<<"\n";

    }
    cout<<"\nNumber of page faults : "<<count1<<endl;
    cout<<"Number of page Hit : "<<n-count1<<endl<<endl;
}
```

Output

```
Enter number of frames : 4
Enter total number of page requests : 20
Enter pages : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Position of frame table after each request :

Page table after request from 7 ->  7 -- -- --  >> page fault
Page table after request from 0 ->  7  0 -- --  >> page fault
Page table after request from 1 ->  7  0  1 --  >> page fault
Page table after request from 2 ->  7  0  1  2  >> page fault
Page table after request from 0 ->  7  0  1  2  >>
Page table after request from 3 ->  3  0  1  2  >> page fault
Page table after request from 0 ->  3  0  1  2  >>
Page table after request from 4 ->  3  0  4  2  >> page fault
Page table after request from 2 ->  3  0  4  2  >>
Page table after request from 3 ->  3  0  4  2  >>
Page table after request from 0 ->  3  0  4  2  >>
Page table after request from 3 ->  3  0  4  2  >>
Page table after request from 2 ->  3  0  4  2  >>
Page table after request from 1 ->  1  0  4  2  >> page fault
Page table after request from 2 ->  1  0  4  2  >>
Page table after request from 0 ->  1  0  4  2  >>
Page table after request from 1 ->  1  0  4  2  >>
Page table after request from 7 ->  1  0  7  2  >> page fault
Page table after request from 0 ->  1  0  7  2  >>
Page table after request from 1 ->  1  0  7  2  >>

Number of page faults : 8
Number of page Hit : 12
```