**Assignment 6**

**Problems Statement:** Implement Page Replacement Algorithm with FIFO Page Replacement, Optimal Page Replacement, LRU Page Replacement Problems.

**FIFO Page Replacement Code:**

*#include*<bits/stdc++.h>

using namespace std;

int main(){

    int n,m,i,j,k,hit=0;

    cout<<"Enter number of frames\n";

    cin>>n;

    cout<<"Enter number of processes\n";

    cin>>m;

    vector<int> p(m);

    vector<int> hi(m);

    cout<<"Enter processes\n";

*for*(i=0;i<m;i++){

        cin>>p[i];

    }

    vector<vector<int>> a(n);

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

        a[i]=vector<int>(m,-1);

    }

    map <int, int> mp;

*for*(i=0;i<m;i++){

        vector<pair<int,int>> c;

*for*(auto q: mp){

            c.push\_back({q.second,q.first});

        }

        sort(c.begin(),c.end());

        bool hasrun=false;

*for*(j=0;j<n;j++){

*if*(a[j][i]==p[i]){

                hit++;

                hi[i]=1;

                mp[p[i]]++;

                hasrun=true;

*break*;

            }

*if*(a[j][i]==-1){

*for*(k=i;k<m;k++)

                    a[j][k]=p[i];

                mp[p[i]]++;

                hasrun=true;

*break*;

            }

        }

*if*(j==n||hasrun==false){

*for*(j=0;j<n;j++){

*if*(a[j][i]==c[c.size()-1].second){

                    mp.erase(a[j][i]);

*for*(k=i;k<m;k++)

                        a[j][k]=p[i];

                    mp[p[i]]++;

*break*;

                }

            }

        }

*for*(auto q:mp){

*if*(q.first!=p[i]){

                mp[q.first]++;

            }

        }

    }

    cout<<"Process ";

*for*(i=0;i<m;i++){

        cout<<p[i]<<" ";

    }

    cout<<'\n';

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

        cout<<"Frame "<<i<<" ";

*for*(j=0;j<m;j++){

*if*(a[i][j]==-1)

                cout<<"E ";

*else*

            cout<<a[i][j]<<" ";

        }

        cout<<'\n';

    }

*for*(i=0;i<m;i++){

*if*(hi[i]==0)

        cout<<"  ";

*else*

        cout<<hi[i]<<" ";

    }

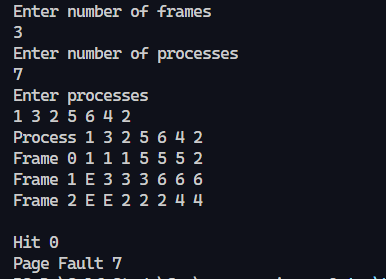
    cout<<"\n";

    cout<<"Hit "<<hit<<'\n'<<"Page Fault "<<m-hit<<'\n';

*return* 0;

}

**Output:**

****

**Text

Description automatically generated with medium confidence**

**Optimal Page Replacement Code:**

*#include*<bits/stdc++.h>

using namespace std;

int main(){

    int n,m,i,j,k;

    cout<<"Enter number of frames\n";

    cin>>n;

    cout<<"Enter number of processes\n";

    cin>>m;

    vector<int> p(m);

    cout<<"Enter processes\n";

*for*(i=0;i<m;i++){

        cin>>p[i];

    }

    vector<vector<int>> a(n,vector<int>(m,-1));

    map <int, int> mp;

*for*(i=0;i<m;i++){

        vector<int> op;

        vector<pair<int,int>> c;

*for*(auto q: mp){

            c.push\_back({q.second,q.first});

        }

*for*(int q=i+1;q<m;q++){

*for*(j=0;j<n;j++){

*if*(a[j][i]==p[q]){

                    op.push\_back(p[q]);

                }

            }

        }

        sort(op.begin(),op.end());

        op.erase(unique(op.begin(),op.end()),op.end());

        bool dontCall=true;

*if*(op.size()==n){

            dontCall=false;

        }

        sort(c.begin(),c.end());

        bool hasrun=false;

*for*(j=0;j<n;j++){

*if*(a[j][i]==p[i]){

                mp[p[i]]++;

                hasrun=true;

*break*;

            }

*if*(a[j][i]==-1){

*for*(k=i;k<m;k++)

                    a[j][k]=p[i];

                mp[p[i]]++;

                hasrun=true;

*break*;

            }

        }

*if*(j==n||hasrun==false){

*for*(j=0;j<n;j++){

*if*(dontCall==true){

*if*(a[j][i]==c[c.size()-1].second){

                        mp.erase(a[j][i]);

*for*(k=i;k<m;k++)

                            a[j][k]=p[i];

                        mp[p[i]]++;

*break*;

                    }

                }

*else* *if*(dontCall==false){

*if*(a[j][i]==op[op.size()-1]){

                        mp.erase(a[j][i]);

*for*(k=i;k<m;k++)

                            a[j][k]=p[i];

                        mp[p[i]]++;

*break*;

                    }

                }

            }

        }

*for*(auto q:mp){

*if*(q.first!=p[i]){

                mp[q.first]++;

            }

        }

    }

    int hit=0;

    vector<int> hitv(m);

*for*(i=1;i<m;i++){

*for*(j=0;j<n;j++){

*if*(p[i]==a[j][i-1]){

                hit++;

                hitv[i]=1;

*break*;

            }

        }

    }

    cout<<"Process ";

*for*(i=0;i<m;i++){

        cout<<p[i]<<" ";

    }

    cout<<'\n';

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

        cout<<"Frame "<<i<<" ";

*for*(j=0;j<m;j++){

*if*(a[i][j]==-1)

                cout<<"E ";

*else*

            cout<<a[i][j]<<" ";

        }

        cout<<'\n';

    }

    cout<<"HIT     ";

*for*(i=0;i<hitv.size();i++){

*if*(hitv[i]==0)

            cout<<"  ";

*else*

        cout<<hitv[i]<<" ";

    }

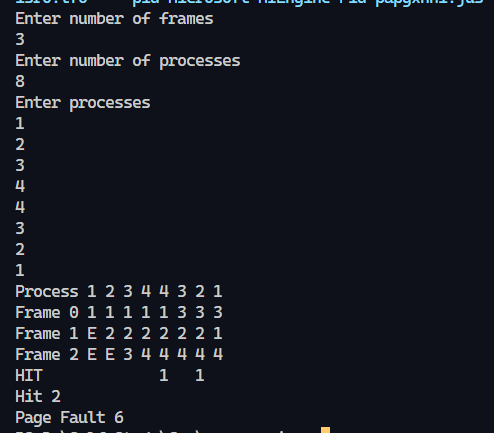
    cout<<"\n";

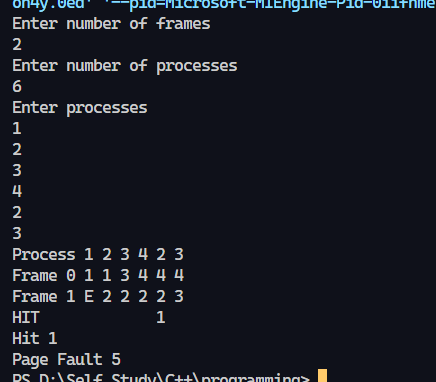
    cout<<"Hit "<<hit<<'\n'<<"Page Fault "<<m-hit<<'\n';

*return* 0;

}

**Output:**

****

****

**LRU Page Replacement Code:**

*#include*<bits/stdc++.h>

using namespace std;

int main(){

    int n,m,i,j,k,hit=0;

    cout<<"Enter number of frames\n";

    cin>>n;

    cout<<"Enter number of processes\n";

    cin>>m;

    vector<int> p(m);

    vector<int> hi(m);

    cout<<"Enter processes\n";

*for*(i=0;i<m;i++){

        cin>>p[i];

    }

    vector<vector<int>> a(n);

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

        a[i]=vector<int>(m,-1);

    }

    map <int, int> mp;

*for*(i=0;i<m;i++){

        vector<pair<int,int>> c;

*for*(auto q: mp){

            c.push\_back({q.second,q.first});

        }

        sort(c.begin(),c.end());

        bool hasrun=false;

*for*(j=0;j<n;j++){

*if*(a[j][i]==p[i]){

                hit++;

                hi[i]=1;

                mp[p[i]]=1;

                hasrun=true;

*break*;

            }

*if*(a[j][i]==-1){

*for*(k=i;k<m;k++)

                    a[j][k]=p[i];

                mp[p[i]]++;

                hasrun=true;

*break*;

            }

        }

*if*(j==n||hasrun==false){

*for*(j=0;j<n;j++){

*if*(a[j][i]==c[c.size()-1].second){

                    mp.erase(a[j][i]);

*for*(k=i;k<m;k++)

                        a[j][k]=p[i];

                    mp[p[i]]++;

*break*;

                }

            }

        }

*for*(auto q:mp){

*if*(q.first!=p[i]){

                mp[q.first]++;

            }

        }

    }

    cout<<"Process ";

*for*(i=0;i<m;i++){

        cout<<p[i]<<" ";

    }

    cout<<'\n';

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

        cout<<"Frame "<<i<<" ";

*for*(j=0;j<m;j++){

*if*(a[i][j]==-1)

                cout<<"E ";

*else*

            cout<<a[i][j]<<" ";

        }

        cout<<'\n';

    }

*for*(i=0;i<m;i++){

*if*(hi[i]==0)

        cout<<"  ";

*else*

        cout<<hi[i]<<" ";

    }

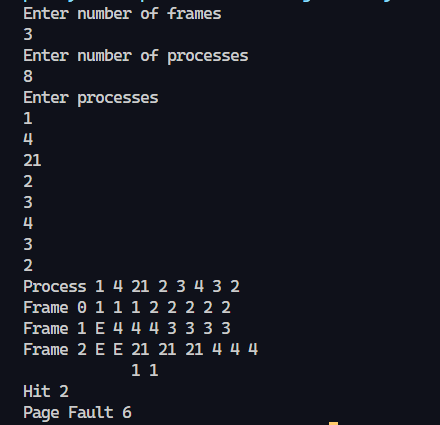
    cout<<"\n";

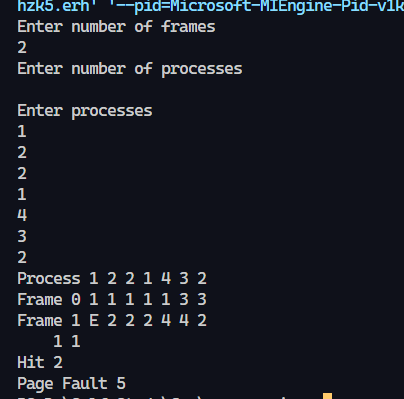
    cout<<"Hit "<<hit<<'\n'<<"Page Fault "<<m-hit<<'\n';

*return* 0;

}

**Output:**

****

****