1. FIFO page replacement algorithm

**Input:**

//fifo

#include<iostream>

using namespace std;

int main(){

    int pages[20], frame[3], i, j=0, k, available, n, fs, count=0;

    cout<<"\nEnter the number of referenced strings: ";

    cin>>n;

    cout<<"\nEnter the referenced pages: ";

    for(i=1;i<=n;i++)

    cin>>pages[i];

    cout<<"\nEnter the frame size: ";

    cin>>fs;

    for(i=0;i<fs;i++)

        frame[i]=-1;

    cout<<"\n\nPages:\tFrame:\n";

    for(i=1;i<=n;i++){

        available=0;

        cout<<pages[i]<<"\t";

        for(k=0;k<fs;k++){

            if(frame[k]==pages[i])

            available=1;

        }

        if(available == 0){

            frame[j]=pages[i];

            j=(j+1)%fs;

            count++;

        }

        for(k=0;k<fs;k++)

            cout<<frame[k]<<" ";

        cout<<"\n";

    }

    cout<<"Total number of page faults are: "<<count;

}

**Output:**

Enter the number of referenced strings: 21

Enter the referenced pages: 2 1 5 6 5 7 8 4 1 2 3 6 3 2 5 9 8 4 2 1 4

Enter the frame size: 3

Pages: Frame:

2 2 -1 -1

1 2 1 -1

5 2 1 5

6 6 1 5

5 6 1 5

7 6 7 5

8 6 7 8

4 4 7 8

1 4 1 8

2 4 1 2

3 3 1 2

6 3 6 2

3 3 6 2

2 3 6 2

5 3 6 5

9 9 6 5

8 9 8 5

4 9 8 4

2 2 8 4

1 2 1 4

4 2 1 4

Total number of page faults are: 17

1. Least Recently Used (LRU) page replacement algorithm

**Input:**

//lru

#include<iostream>

using namespace std;

int main(){

    int pages[25],frame[3],i,j=0,k,available,n,fs,count=0,lastused[25],min,index;

    cout<<"Enter the number of referenced strings: ";

    cin>>n;

    cout<<"\nEnter the referenced pages: ";

    for(i=1;i<=n;i++)

    cin>>pages[i];

    cout<<"\nEnter the frame size: ";

    cin>>fs;

    for(i=0;i<fs;i++){

    frame[i]=-1;

    lastused[i]=0;

    }

    cout<<"\nPages:\tFrame:\n";

    for(i=1;i<=n;i++){

        available=0;

        cout<<pages[i]<<"\t";

        for(k=0;k<fs;k++){

        if(frame[k]==pages[i]){

            available=1;

            lastused[k]=i+1;

        }

        }

        if(available == 0){

            min=lastused[0];

            index=0;

            for(k=1;k<=fs;k++){

                if(lastused[k]<min){

                min=lastused[k];

                index=k;

                }

            }

            frame[index]=pages[i];

            lastused[index]=i+1;

            count++;

        }

        for(k=0;k<fs;k++)

            cout<<frame[k]<<" ";

        cout<<"\n";

    }

    cout<<"\nTotal number of page faults are: "<<count;

    return 0;

}

**Output:**

Enter the number of referenced strings: 21

Enter the referenced pages: 3 4 3 2 1 2 6 4 5 7 4 1 3 7 7 5 4 1 3 8 5

Enter the frame size: 3

Pages: Frame:

3 3 -1 -1

4 3 4 -1

3 3 4 -1

2 3 4 2

1 3 1 2

2 3 1 2

6 6 1 2

4 6 4 2

5 6 4 5

7 7 4 5

4 7 4 5

1 7 4 1

3 3 4 1

7 3 7 1

7 3 7 1

5 3 7 5

4 4 7 5

1 4 1 5

3 4 1 3

8 8 1 3

5 8 5 3

Total number of page faults are: 17

1. Optimal page replacement algorithm

**Input:**

//optimal

#include <iostream>

using namespace std;

int main() {

    int pages[25], frame[3], n, fs, count = 0;

    cout << "Enter the number of referenced strings: ";

    cin >> n;

    cout << "\nEnter the referenced pages:";

    for (int i = 0; i < n; i++) {

    cin >> pages[i];

    }

    cout << "\nEnter the frame size: ";

    cin >> fs;

    for (int i = 0; i < fs; i++) {

    frame[i] = -1;

    }

    cout << "\nPages:\tFrame:\n";

    for (int i = 0; i < n; i++) {

        bool available = false;

        for (int j = 0; j < fs; j++) {

            if (frame[j] == pages[i]) {

                available = true;

                break;

            }

        }

        if (!available) {

            count++;

            int index = -1;

            for (int j = 0; j < fs; j++) {

                if (frame[j] == -1) {

                index = j;

                break;

                }

            }

            if (index == -1) {

                int farthest = -1;

                for (int j = 0; j < fs; j++) {

                    int k;

                    for (k = i + 1; k < n; k++) {

                    if (frame[j] == pages[k]) {

                        if (k > farthest) {

                            farthest = k;

                            index = j;

                        }

                        break;

                    }

                    }

                    if (k == n) {

                        index = j;

                        break;

                    }

                }

            }

            frame[index] = pages[i];

        }

        cout << pages[i] << "\t";

        for (int j = 0; j < fs; j++) {

            cout << frame[j] << " ";

        }

        cout << endl;

    }

    cout << "\nTotal number of page faults: " << count << endl;

    return 0;

}

**Output:**

Enter the number of referenced strings: 20

Enter the referenced pages:7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Enter the frame size: 3

Pages: Frame:

7 7 -1 -1

0 7 0 -1

1 7 0 1

2 2 0 1

0 2 0 1

3 2 0 3

0 2 0 3

4 2 4 3

2 2 4 3

3 2 4 3

0 2 0 3

3 2 0 3

2 2 0 3

1 2 0 1

2 2 0 1

0 2 0 1

1 2 0 1

7 7 0 1

0 7 0 1

1 7 0 1

Total number of page faults: 9