# Assignment No.6

**Name :** Jay Pawar

# Batch : B1

**Roll no. :** 22

# FIFO:

## Code:

#include<bits/stdc++.h>

using namespace std;

int main()

{

string s;

cout<<"Enter The String:"; cin>>s;

int b=0; int c = 0; int a[3];

for(int i=0;i<s.length();i++)

{

if(a[0]!=int(s.at(i)) && a[1]!=int(s.at(i)) && a[2]!=int(s.at(i)))

{

a[c] = int(s.at(i)); b++;

c++;

if(c==3)

c=0;

}

}

cout<<b<<endl; return 0;

}

## Output:

**OPTIMAL:**

## CODE:

#include<stdio.h>

void optimal(int string[20],int n,int size)

{

//Creating array for block storage int frames[n];

//Initializing each block with -1 for (int i=0;i<n;i++)

frames[i]=-1;

//Index to insert element int index=-1;

//Counters

int page\_miss=0; int page\_hits=0; int full=0;

for (int i=0;i<size;i++)

{

int symbol=string[i]; int flag=0;

for(int j=0;j<n;j++)

{

if (symbol==frames[j])

{

flag=1; break;

}

}

if (flag==1)

{

printf("\nSymbol: %d Frame: ",symbol); for (int j=0;j<n;j++)

printf("%d ",frames[j]); page\_hits+=1;

}

else

{

if (full==0)

{

index=(index+1)%n; frames[index]=symbol; page\_miss+=1;

printf("\nSymbol: %d Frame: ",symbol); for (int j=0;j<n;j++)

printf("%d ",frames[j]);

//Frames filled or not if (i==n-1)

full=1;

}

else

{

//First find the index to replace with int pos=-1;

int index=-1; for(int j=0;j<n;j++)

{

//Whether symbol in frame found or not in future int found=0;

for (int k=i+1;k<size;k++)

{

//If symbol exists in cached string if (frames[j]==string[k])

{

found=1; if (pos<k)

{

pos=k; index=j;

}

break;

}

}

//Symbol does not exist in cached frame if (found==0)

{

pos=size; index=j;

}

}

//Now assign symbol in lru position frames[index]=symbol;

printf("\nSymbol: %d Frame: ",symbol); for (int j=0;j<n;j++)

printf("%d ",frames[j]);

}

}

}

printf("\nPage hits: %d",page\_hits); printf("\nPage misses: %d",page\_miss);

}

//Main function int main(void)

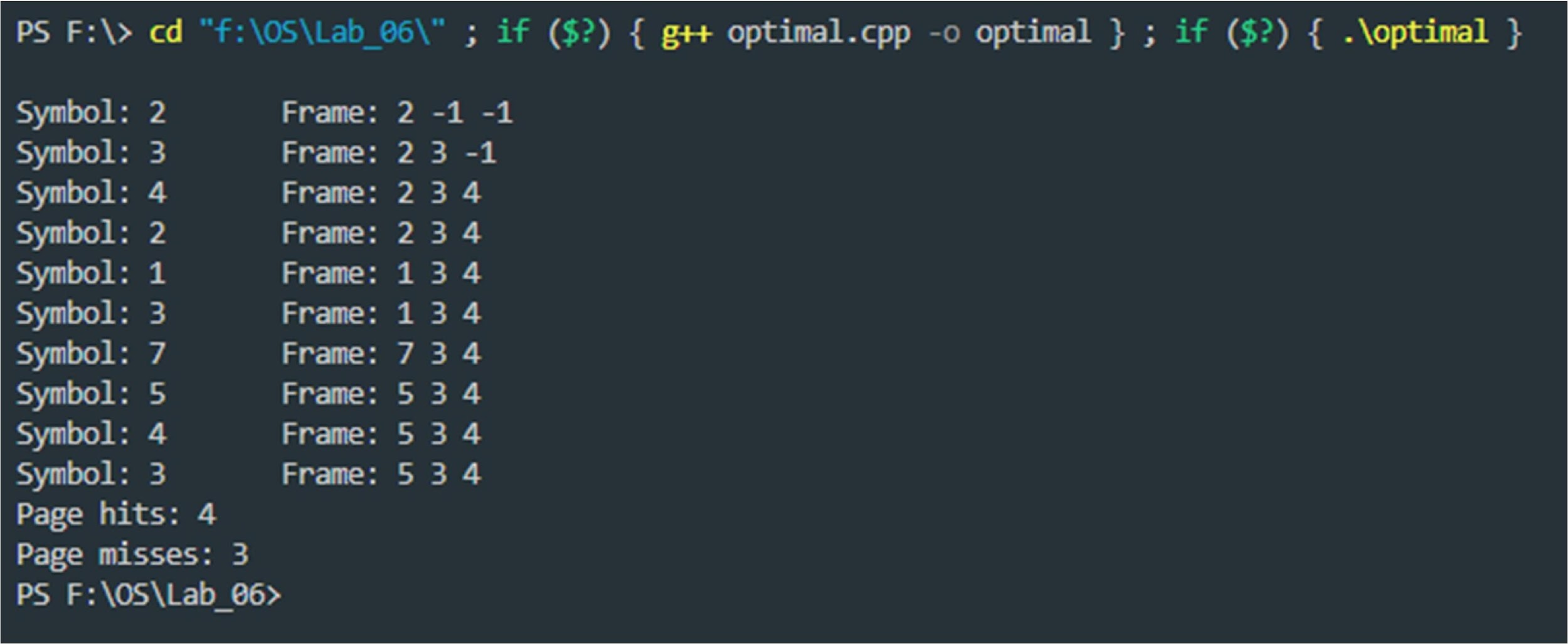
{

int string[]={2, 3, 4, 2, 1, 3, 7, 5, 4, 3};

int no\_frames=3;

int size=sizeof(string)/sizeof(int); optimal(string,no\_frames,size); return 0;

}

**OUTPUT:**

# LRU:

## CODE:

#include<bits/stdc++.h>

int findLRU(int time[], int n)

{

int i, minimum = time[0], pos = 0; for (i = 1; i < n; ++i)

{

if (time[i] < minimum)

{

minimum = time[i]; pos = i;

}

}

return pos;

}

int main()

{

int no\_of\_frames, no\_of\_pages, frames[10], pages[30], counter = 0, time[10], flag1, flag2, i, j, pos, faults = 0;

printf("Enter number of frames: ");

scanf("%d", &no\_of\_frames); printf("Enter number of pages: "); scanf("%d", &no\_of\_pages); printf("Enter reference string: "); for (i = 0; i < no\_of\_pages; ++i)

{

scanf("%d", &pages[i]);

}

for (i = 0; i < no\_of\_frames; ++i)

{

frames[i] = -1;

}

for (i = 0; i < no\_of\_pages; ++i)

{

flag1 = flag2 = 0;

for (j = 0; j < no\_of\_frames; ++j)

{

if (frames[j] == pages[i])

{

counter++; time[j] = counter; flag1 = flag2 = 1; break;

}

}

if (flag1 == 0)

{

for (j = 0; j < no\_of\_frames; ++j)

{

if (frames[j] == -1)

{

counter++; faults++;

frames[j] = pages[i]; time[j] = counter; flag2 = 1;

break;

}

}

}

if (flag2 == 0)

{

pos = findLRU(time, no\_of\_frames); counter++;

faults++;

frames[pos] = pages[i]; time[pos] = counter;

}

printf("\n");

for (j = 0; j < no\_of\_frames; ++j)

{

printf("%d\t", frames[j]);

}

}

printf("\n\nTotal Page Faults = %d", faults); return 0;

}

## OUTPUT: