

**NAME:** *Hariprasad K K*

**REG NO:** *19BCE7079*

**COURSE :** *Operating System*

**DATE :** *13-11-2021*

**CODE:**

**FIL21.C**

```
#include<stdio.h>
int main()
{
    int no_of_frames, no_of_pages, frames[10], pages[30], temp[10],
    flag1, flag2, flag3, i, j, k, pos, max, faults = 0;
    printf("Enter number of frames: ");
    scanf("%d", &no_of_frames);

    printf("Enter number of pages: ");
    scanf("%d", &no_of_pages);

    printf("Enter page 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]){
                flag1 = flag2 = 1;
                break;
            }
        }

        if(flag1 == 0){
            for(j = 0; j < no_of_frames; ++j){
                if(frames[j] == -1){
                    faults++;
                }
            }
        }
    }
}
```

```

        frames[j] = pages[i];
        flag2 = 1;
        break;
    }
}

if(flag2 == 0){
    flag3 = 0;

    for(j = 0; j < no_of_frames; ++j){
        temp[j] = -1;

        for(k = i + 1; k < no_of_pages; ++k){
            if(frames[j] == pages[k]){
                temp[j] = k;
                break;
            }
        }
    }

    for(j = 0; j < no_of_frames; ++j){
        if(temp[j] == -1){
            pos = j;
            flag3 = 1;
            break;
        }
    }

    if(flag3 == 0){
        max = temp[0];
        pos = 0;

        for(j = 1; j < no_of_frames; ++j){
            if(temp[j] > max){
                max = temp[j];
                pos = j;
            }
        }
    }

    frames[pos] = pages[i];
    faults++;
}

printf("\n");

for(j = 0; j < no_of_frames; ++j){
    printf("%d\t", frames[j]);
}

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

return 0;
}

```

## FIL20.C

```
#include<stdio.h>
int main()
{
int i,j,n,a[50],frame[10],no,k,avail,count=0;
    printf("\n ENTER THE NUMBER OF PAGES:\n");
scanf("%d",&n);
    printf("\n ENTER THE PAGE NUMBER :\n");
    for(i=1;i<=n;i++)
        scanf("%d",&a[i]);
    printf("\n ENTER THE NUMBER OF FRAMES :");
    scanf("%d",&no);
for(i=0;i<no;i++)
    frame[i]= -1;
    j=0;
    printf("\tref string\t page frames\n");
for(i=1;i<=n;i++)
    {
        printf("%d\t\t",a[i]);
        avail=0;
        for(k=0;k<no;k++)
            if(frame[k]==a[i])
                avail=1;
        if (avail==0)
        {
            frame[j]=a[i];
            j=(j+1)%no;
            count++;
            for(k=0;k<no;k++)
                printf("%d\t",frame[k]);
        }
        printf("\n");
    }
    printf("Page Fault Is %d",count);
    return 0;
}
```

## FILL22.C

```
#include<stdio.h>
int main()
{
int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];
printf("Enter no of pages:");
scanf("%d",&n);
printf("Enter the reference string:");
for(i=0;i<n;i++)
    scanf("%d",&p[i]);
printf("Enter no of frames:");
```

```

scanf("%d",&f);
q[k]=p[k];
printf("\n\t%d\n",q[k]);
c++;
k++;
for(i=1;i<n;i++)
{
    c1=0;
    for(j=0;j<f;j++)
    {
        if(p[i]!=q[j])
            c1++;
    }
    if(c1==f)
    {
        c++;
        if(k<f)
        {
            q[k]=p[i];
            k++;
            for(j=0;j<k;j++)
                printf("\t%d",q[j]);
            printf("\n");
        }
        else
        {
            for(r=0;r<f;r++)
            {
                c2[r]=0;
                for(j=i-1;j<n;j--)
                {
                    if(q[r]!=p[j])
                        c2[r]++;
                    else
                        break;
                }
            }
            for(r=0;r<f;r++)
                b[r]=c2[r];
            for(r=0;r<f;r++)
            {
                for(j=r;j<f;j++)
                {
                    if(b[r]<b[j])
                    {
                        t=b[r];
                        b[r]=b[j];
                        b[j]=t;
                    }
                }
            }
        }
    }
}

```

```

    }
    for (r=0; r<f; r++)
    {
        if (c2[r]==b[0])
            q[r]=p[i];
        printf("\t%d", q[r]);
    }
    printf("\n");
}

}

printf("\nThe no of page faults is %d", c);
}

```

## SCREENSHOTS:

```

1 // Page Faults in Set Associative Cache
2 //
3 // Parameters:
4 // - n: Number of pages (0 to 1023)
5 // - k: Number of frames (0 to 1023)
6 // - r: Reference string (array of integers)
7 // - f: Frame set (array of integers)
8 //
9 // Returns:
10 // - Total page faults
11 //
12 // Author:
13 // - [Your Name]
14 //
15 // Date:
16 // - [Date]
17 //
18 // Version:
19 // - 1.0
20 //
21 // License:
22 // - MIT
23 //
24 // Usage:
25 // - gcc -std=c99 -O2 -DDEBUG -Dn=1024 -Dk=1024 -Dr="{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,9
```

```

#include<stdio.h>
int main()
{
    int q[10],p[10],c=0,i=1,k,r,t,j,k=0,n,r,t,s[20],z[10];
    printf("Enter no of pages:-");
    scanf("%d",&n);
    printf("Enter the reference string:-");
    for(i=0;i<n;i++)
        scanf("%d",&p[i]);
    printf("Enter no of frames:-");
    scanf("%d",&r);
    q[k]=p[k];
    printf("%d\t%d\t",q[k]);
    i++;
    for(i=1;i<n;i++)
    {
        c++;
        for(j=0;j<r;j++)
        {
            if(p[i]==q[j])
            {
                c++;
            }
            if(c==r)
            {
                q[j]=p[i];
                k++;
                for(j=k;j<n;j++)
                {
                    printf("%d\t",q[j]);
                }
                printf("\n");
            }
            else
            {
                for(i=0;i<r;i++)
                {
                    if(r==i)
                    {
                        c[r]=p[i];
                        for(j=0;j<n;j++)
                        {
                            if(q[j]==p[i])
                            {
                                c[j]++;
                                break;
                            }
                        }
                    }
                    if(c[j]==r)
                    {
                        for(i=0;i<r;i++)
                        {
                            if(r==i)
                            {
                                c[r]=p[i];
                                for(j=0;j<n;j++)
                                {
                                    if(q[j]==p[i])
                                    {
                                        c[j]++;
                                        break;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

File Edit View Search Terminal Help
vlt-app@kali:~$ gcc ./a.out
./a.out: file not recognized: File truncated
collect2: error: ld returned 1 exit status
vlt-app@kali:~$ gcc ./a.out
vlt-app@kali:~$ ./a.out
Enter the number of pages:
4
Enter the page number :
1
2
3
4
Enter the number of frames is
ref string      page frames
1      1      -1      -1      -1
2      1      2      -1      -1
3      1      2      1      -1
4      1      2      1      0
vlt-app@kali:~$ gcc ./a.out
vlt-app@kali:~$ ./a.out
Enter number of frames: 3
Enter number of pages: 4
Enter page reference string: 3
1
2
3
4
1      -1      -1
2      2      -1
3      2      0
4      2      0
Total Page Faults = vlt-app@kali:~$
vlt-app@kali:~$ gcc ./a.out
vlt-app@kali:~$ gcc ./a.out
./a.out: warning: return type defaults to 'int' [-Wimplicit-int]
main()
{
}
vlt-app@kali:~$ gcc ./a.out
gcc: error: ./a.out: no such file or directory
Compilation terminated.
vlt-app@kali:~$ gcc ./a.out
vlt-app@kali:~$ ./a.out
Enter no of pages: 4
Enter the reference string:
1
2
3
4
Enter no of frames: 3
1
2
3
4

```

```

#include<stdio.h>
int main()
{
    int no_of_frames, no_of_pages, frames[10], pages[10], temp[10], flag1, flag2, flag3, i, j, k, pos, max, faults = 0;
    printf("Enter number of frames: ");
    scanf("%d", &no_of_frames);

    printf("Enter number of pages: ");
    scanf("%d", &no_of_pages);

    printf("Enter page 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]){
                flag1 = flag2 = 1;
                break;
            }
        }

        if(flag1 == 0){
            for(j = 0; j < no_of_frames; ++j){
                if(frames[j] == -1){
                    faults++;
                    frames[j] = pages[i];
                    flag2 = 1;
                    break;
                }
            }
        }

        if(flag2 == 0){
            flag3 = 0;

            for(j = 0; j < no_of_frames; ++j){
                temp[j] = -1;

                for(k = 1 + j; k < no_of_pages; ++k){
                    if(frames[j] == pages[k]){
                        temp[j] = k;
                        break;
                    }
                }
            }

            if(flag3 == 0){
                max = temp[0];
                pos = 0;

                for(j = 1; j < no_of_frames; ++j){
                    if(temp[j] < max){
                        max = temp[j];
                        pos = j;
                    }
                }

                if(flag3 == 0){
                    max = temp[pos];
                    pos = 0;

                    for(j = 1; j < no_of_frames; ++j){
                        if(temp[j] < max){
                            max = temp[j];
                            pos = j;
                        }
                    }
                }

                frames[pos] = pages[i];
                faults++;

                printf("\n");

                for(j = 0; j < no_of_frames; ++j){
                    printf("%d\t", frames[j]);
                }
            }
        }

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

        return 0;
    }
}

```

```

        break;
    }
}

if(flag2 == 0){
    flag3 = 0;

    for(j = 0; j < no_of_frames; ++j){
        temp[j] = -1;

        for(k = 1 + j; k < no_of_pages; ++k){
            if(frames[j] == pages[k]){
                temp[j] = k;
                break;
            }
        }
    }

    for(j = 0; j < no_of_frames; ++j){
        if(temp[j] == -1){
            pos = j;
            flag3 = 1;
            break;
        }
    }

    if(flag3 == 0){
        max = temp[0];
        pos = 0;

        for(j = 1; j < no_of_frames; ++j){
            if(temp[j] < max){
                max = temp[j];
                pos = j;
            }
        }

        frames[pos] = pages[i];
        faults++;

        printf("\n");

        for(j = 0; j < no_of_frames; ++j){
            printf("%d\t", frames[j]);
        }
    }
}

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

return 0;
}

```



[illegible]

```

File Edit View Search Format Help
vlt-appcelabs-4i-g cat /lib/c
#include <stdio.h>

int main()
{
    int i,j,a[50],Frame[30],no,k,avail,count=0;
    printf("\n ENTER THE NUMBER OF PAGES:(a*)");
    scanf("%d",&a);
    printf("\n ENTER THE PAGE NUMBER :-(a*)");
    for(i=1;i<=a;i++)
        scanf("%d",&a[i]);
    printf("\n ENTER THE NUMBER OF FRAMES :(-)");
    scanf("%d",&no);
    for(i=0;i<no;i++)
        Frame[i]=-1;
    while(1)
    {
        printf("\nref string(a page frames(a*))");
        for(i=0;i<no;i++)
        {
            printf("%d\t",a[i]);
            avail=0;
            for(k=0;k<no;k++)
            {
                avail=1;
                if (a[k]==-1)
                {
                    Frame[i]=a[i];
                    i=i+1;no--;
                    count++;
                    for(k=0;k<no;k++)
                        printf("%d\t",Frame[k]);
                }
                printf("\n");
            }
            printf("Page Fault is %d",count);
            return 0;
        }
    }
}

vlt-appcelabs-4i-g gcc ./a.out
./a.out: file not recognized: file truncated
collect2: error: ld returned 1 exit status
vlt-appcelabs-4i-g gcc /lib/c
vlt-appcelabs-4i-g ./a.out

ENTER THE NUMBER OF PAGES:
4

ENTER THE PAGE NUMBER :
2
2
2
2

ENTER THE NUMBER OF FRAMES :4
ref string      page frames
a      1      -1      -1      -1
2      1      2      -1      -1

```