1.FIFO

#include<stdio.h>

int main()

{

int i,j,n,a[50],frame[10],no,k,avail,count=0;

printf("\n Enter The Number Of Page:\n");

scanf("%d",&n);

printf("\n Enter The Page Number(refrence string):\n");

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

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

printf("\n Enter the Number Of Frame:");

scanf("%d",&no);

for(i=0;i<no;i++)

frame[i]=-1;

j=0;

printf("\t +ref string\t page frame\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;

}

3.LRU

#include<stdio.h>

void 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 thre 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<f;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("\n The no.of page faults is %d",c);

}

4.MFU

#include<stdio.h>

#define MAX 20

int frames[MAX],ref[MAX],mem[MAX][MAX],faults,

 sp,m,n,count[MAX];

void accept()

{

 int i;

 printf("Enter no.of frames:");

 scanf("%d", &n);

 printf("Enter no.of references:");

 scanf("%d", &m);

 printf("Enter reference string:\n");

 for(i=0;i<m;i++)

 {

  printf("[%d]=",i+1);

  scanf("%d",&ref[i]);

 }

}

void disp()

{

 int i,j;

 for(i=0;i<m;i++)

  printf("%3d",ref[i]);

 printf("\n\n");

 for(i=0;i<n;i++)

 {

  for(j=0;j<m;j++)

  {

   if(mem[i][j])

    printf("%3d",mem[i][j]);

   else

    printf("   ");

  }

  printf("\n");

 }

 printf("Total Page Faults: %d\n",faults);

}

int search(int pno)

{

 int i;

 for(i=0;i<n;i++)

 {

  if(frames[i]==pno)

   return i;

 }

 return -1;

}

int get\_mfu(int sp)

{

 int i,max\_i,max=-9999;

 i=sp;

 do

 {

  if(count[i]>max)

  {

   max = count[i];

   max\_i = i;

  }

  i=(i+1)%n;

 }while(i!=sp);

 return max\_i;

}

void mfu()

{

 int i,j,k;

 for(i=0;i<m && sp<n;i++)

 {

  k=search(ref[i]);

  if(k==-1)

  {

   frames[sp]=ref[i];

   count[sp]++;

   faults++;

   sp++;

   for(j=0;j<n;j++)

    mem[j][i]=frames[j];

  }

  else

   count[k]++;

 }

 sp=0;

 for(;i<m;i++)

 {

  k = search(ref[i]);

  if(k==-1)

  {

   sp = get\_mfu(sp);

   frames[sp] = ref[i];

   count[sp]=1;

   faults++;

   sp = (sp+1)%n;

   for(j=0;j<n;j++)

    mem[j][i] = frames[j];

  }

  else

   count[k]++;

 }

}

int main()

{

 accept();

 mfu();

 disp();

 return 0;

}

5.OPT

#include<stdio.h>

int n;

int findmax(int a[], int n);

int main() {

    int seq[30], fr[5], pos[5], find, flag, max, i, j, m, k, t, s;

    int count = 1, pf = 0, p = 0;

    float pfr;

    printf("Enter the maximum limit of seq: ");

    scanf("%d", &max);

    printf("\nEnter the sequence: ");

    for(i = 0; i < max; i++)

        scanf("%d", &seq[i]);

    printf("\nEnter the number of frames: ");

    scanf("%d", &n);

    fr[0] = seq[0];

    pf++;

    printf("%d\t", fr[0]);

    i = 1;

    while(count < n) {

        flag = 1;

        p++;

        for(j = 0; j < i; j++) {

            if(seq[i] == seq[j]) {

                flag = 0;

                break;

            }

        }

        if(flag != 0) {

            fr[count] = seq[i];

            printf("%d\t", fr[count]);

            count++;

            pf++;

        }

        i++;

    }

    for(i = count; i < max; i++) {

        flag = 1;

        for(j = 0; j < n; j++) {

            if(seq[i] == fr[j]) {

                flag = 0;

                break;

            }

        }

        if(flag != 0) {

            for(j = 0; j < n; j++) {

                pos[j] = -1;

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

                    if(fr[j] == seq[k]) {

                        pos[j] = k;

                        break;

                    }

                }

            }

            s = findmax(pos, n);

            fr[s] = seq[i];

            for(k = 0; k < n; k++) {

                printf("%d\t", fr[k]);

            }

            pf++;

            printf("\n");

        }

    }

    pfr = (float)pf / (float)max;

    printf("\nThe number of page faults is: %d", pf);

    printf("\nPage fault rate: %f\n", pfr);

    return 0;

}

int findmax(int a[], int n) {

    int max = -1, i, k = 0;

    for(i = 0; i < n; i++) {

        if(a[i] == -1) {

            return i;

        }

        if(a[i] > max) {

            max = a[i];

            k = i;

        }

    }

    return k;

}