FIFO --

```
#include<stdio.h>
int main(){
   int frames, pages, i, j, hit=0, fault=0, counter=0;
   int reference_string[100], mem_layout[100][100];
   printf("\nEnter the number of frames: ");
   scanf("%d",&frames);
   printf("\nEnter the number of pages: ");
   scanf("%d",&pages);
   printf("\nEnter the reference string:");
   for(i=0;i<pages;i++){
      scanf("%d",&reference_string[i]);
   }
   for(i=0;i<frames;i++){
      mem_layout[i][0]=-1;
   }
   for(i=0;i<pages;i++){
      hit=0;
      for(j=0;j<frames;j++){
         if(mem_layout[j][0]==reference_string[i]){
            hit=1;
            mem_layout[j][i+1]=1;
            break;
         }
      }
      if(hit==0){
         mem_layout[counter][0]=reference_string[i];
         fault++;
         for(j=0;j<frames;j++){
            mem_layout[counter][j+1]=0;
         }
         counter++;
         if(counter==frames){
            counter=0;
         }
      }
      printf("\n");
      for(j=0;j<frames;j++){
         printf("%d\t",mem_layout[j][i]);
      }
   }
   printf("\nTotal Page Faults:%d",fault);
   return 0;
}
```

LRU--

```
#include<stdio.h>
#define MAX_PAGES 100

int findLRU(int time[], int n){
   int i, min = time[0], pos = 0;
   for(i=1;i<n;i++){
      if(time[i]<min){
         min = time[i];
         pos = i;
```

```
    }
  }
  return pos;
}

int main(){
  int pages[MAX_PAGES], frames, n, i, j, k, faults = 0, pos;
  printf("Enter the number of frames: ");
  scanf("%d", &frames);
  printf("Enter the number of pages: ");
  scanf("%d", &n);
  printf("Enter the reference string:");
  for(i=0;i<n;i++){
    scanf("%d", &pages[i]);
  }
  int mem[frames], time[frames];
  for(i=0;i<frames;i++){
    mem[i] = -1;
    time[i] = 0;
  }
  for(i=0;i<n;i++){
    for(j=0;j<frames;j++){
      if(mem[j]==pages[i]){
        time[j] = i+1;
        break;
      }
    }
    if(j==frames){
      pos = findLRU(time, frames);
      mem[pos] = pages[i];
      time[pos] = i+1;
      faults++;
    }
    printf("\n");
    for(k=0;k<frames;k++){
      printf("%d\t", mem[k]);
    }
  }
  printf("\nTotal Page Faults:%d", faults);
  return 0;
}
```

OPTIMAL--

```
#include<stdio.h>
#include<limits.h>
int main(){
  int frames, pages, i, j, k, l, hit=0, fault=0, max_dist, max_frame, flag;
  int reference_string[100], mem_layout[100][100], distance[100];
  printf("\nEnter the number of frames: ");
  scanf("%d",&frames);
  printf("\nEnter the number of pages: ");
  scanf("%d",&pages);
  printf("\nEnter the reference string:");
  for(i=0;i<pages;i++){
    scanf("%d",&reference_string[i]);
  }
  for(i=0;i<frames;i++){
```

```
      mem_layout[i][0]=-1;
   }
   for(i=0;i<pages;i++){
      hit=0;
      for(j=0;j<frames;j++){
         if(mem_layout[j][0]==reference_string[i]){
            hit=1;
            break;
         }
      }
      if(hit==0){
         fault++;
         flag=0;
         for(j=0;j<frames;j++){
            if(mem_layout[j][0]==-1){
               mem_layout[j][0]=reference_string[i];
               flag=1;
               break;
            }
         }
         if(flag==0){
            for(j=0;j<frames;j++){
               distance[j]=INT_MAX;
               for(k=i+1;k<pages;k++){
                  if(reference_string[k]==mem_layout[j][0]){
                     distance[j]=k-i;
                     break;
                  }
               }
            }
            max_dist=-1;
            for(j=0;j<frames;j++){
               if(distance[j]>max_dist){
                  max_dist=distance[j];
                  max_frame=j;
               }
            }
            mem_layout[max_frame][0]=reference_string[i];
         }
      }
      for(j=0;j<frames;j++){
         mem_layout[j][i+1]=mem_layout[j][i];
      }
   }
   printf("\nTotal Page Faults:%d",fault);
   return 0;
}
```