```
        p[repindex] = in[i];
        pgfaultcnt++;
        dispPages();
    } else
        printf("No page fault:");
    } dispPgFaultCnt();
}
int main()
{   int choice;
    while(1)
    {   printf("\n Page Replacement Algorithms
\n 1. Enter data \n 2. FIFO \n 3. optimal \n 4. LRU
\n 5. Exit \n Enter your choice:");
    scanf("%d", &choice);
    switch(choice)
    {  case 1: getdata();
        break;
      case 2: fifo();
        break;
      case 3: optimal();
        break;
      case 4: lru();
        break;
      default: return 0;
        break;
    }
}
```

OUTPUT

Page Replacement Algorithms

1. Enter data
2. FIFO
3. Optimal
4. LRU
5. Exit

Enter your choice: 2

Enter length of page reference ~~Sequence~~ length: 12

Enter the page reference: 1 2 3 4 1 2 5 1 2 3 4 5

Enter the no of frames: 3

For 1: 1

For 2: 1 2

For 3: 1 2 3

For 4: 2 3 4

For 1: 3 4 1

For 2: 4 1 2

For 5: 1 2 5

For 1: No page fault

For 2: No page fault

For 3: 2 5 3

For 4: 5 3 4

For 5: No page fault

Total no of page faults: 9

Menu

Enter your choice: 5

3/7/24

Q Write a CProgram to stimulate disk scheduling algorithms a) FCFS

→
```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int RQ[100], i, n, TotalHeadMoment = 0, initial;
  printf("Enter the number of Requests \n");
  scanf("%d", &n);
  printf("Enter the Requests sequence \n");
  for(i=0; i<n; i++)
      scanf("%d", &RQ[i]);
  printf("Enter initial head position \n");
  scanf("%d", &initial);
  for(i=0; i<n; i++)
  {
      TotalHeadMoment = TotalHeadMoment +
abs(RQ[i] - initial);
      initial = RQ[i];
  }
  printf("Total head moment is %d",
TotalHeadMoment);
  return 0;
}
```

OUTPUT

```
Enter the number of requests 5
Enter the requests sequence
  5  98  107  45  78
Enter initial head position 45
Total Head Movement is 237
```

b). SCAN

```
# include <stdio.h>
# include <stdlib.h>
int main ()
{ int RQ[100], i, j, n, TotalHeadMoment = 0,
  Initial, size, move;
  printf ("Enter the number of Requests \n");
  Scanf ("%d", &n);
  printf ("Enter the Requests sequence \n");
  for (i=0; i<n; i++)
  Scanf ("%d", &RQ[i]);
  printf ("Enter Initial head position \n");
  scanf ("%d", &size);
  printf ("Enter the head movement direction
for high 1 and for low 0 \n");
  Scanf ("%d", &move);
  for (j=0; j<n; i++)
  {  for (j=0; j<n-i-1; j++)
     {  if (RQ[j] > RQ[j+1])
        {  int temp;
           temp= RQ[j]
           RQ[j] = RQ[j+1];
           RQ[j+1] = temp;
        }
     }
  }
  int index;
```

```
for (i=0; i<n; i++)
{   if (initial < RQ[i])
      {   index = i;
          break;
      }
}
if (move == 1)
{  for (i= index; i<n; i++)
    {  TotalHeadMoment = TotalHeadMoment +
abs(RQ[i] - initial);
       initial = RQ[i];
    }
   TotalHeadMoment = TotalHeadMoment +
abs(size - RQ[i-1] - 1);
 initial = size - 1;
 for (i= index-1; i>=0; i--)
  {   TotalHeadMoment = TotalHeadMoment +
abs(RQ[i] - initial);
    initial = RQ[i];
  }
} else
{ for (i= index-1; i>=0; i--)
    {  TotalHeadMoment = TotalHeadMoment
 + abs(RQ[i] - initial);
    initial = RQ[i];
 }
TotalHeadMoment = TotalHeadMoment + abs(RQ[i+1] - 0);
```

```
initial = 0;
for (i= index; i<n; i++)
{   TotalHeadMoment = TotalHeadMoment +
abs (RQ[i] - initial);
    initial = RQ[i];
}
}
printf("Total head movement is %d", TotalHead
Moment);
return 0;
}
```

OUTPUT

Enter the number of requests &
Enter the requests sequence
98 183 37 122 14 124 65 67
Enter initial head position
53
Enter total disk size
200
Enter the head movement direction for high 1
and for low 0
1
Total Head Movement is 359

c) C - SCAN

→

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int RQ[100], i, j, n, TotalHeadMoment=0,
    initial, size, move;
    printf("Enter the number of Requests\n");
    scanf("%d", &n);
    printf("Enter the Requests Sequence\n");
    for(i=0; i<n; i++)
    scanf("%d", &RQ[i]);
    printf("Enter initial head position\n");
    scanf("%d", &initial);
    printf("Enter total disk size\n");
    scanf("%d", &size);
    printf("Enter the head movement direction
for high 1 and for low 0\n");
    scanf("%d", &move);
    for(i=0; i<n; i++)
    {
        for(j=0; j<n-i-1; j++)
        {
            if(RQ[j] > RQ[j+1])
            {
                int temp;
                temp = RQ[j];
                RQ[j] = RQ[j+1];
                RQ[j+1] = temp;
            }
        }
    }
}
```

```
int index;
for (i=0; i<n; i++)
{ if (initial < RQ[i])
    { index = i;
      break;
    }
}
if (move == 1)
{ for (i= index; i<n; i++
    { TotalHeadMoment = TotalHeadMoment +
abs (RQ[i] - initial);
  initial = RQ[i];
}
TotalHeadMoment = TotalHeadMoment + abs (size
 - RQ[i-1] -1);
TotalHeadMoment = TotalHeadMoment + abs (size-1-0);
 initial = 0;
for (i=0; i< index; i++)
{  TotalHeadMoment += abs (RQ[i] - initial);
    initial = RQ[i];
}
} else
{  for ( i = index -1 ; i>=0; i--)
    { TotalHeadMoment += abs(RQ[i] - initial);
      initial = RQ[i];
    }
```

```
TotalHeadMovement += abs(RQ[i+1]-0);
TotalHeadMovement = TotalHeadMovement +
 abs(size-1-0);
initial = size-1;
for(i=n-1;i>=index;i--)
 { TotalHeadMoment += abs(RQ[i]-initial);
    initial = RQ[i];
 }
}
printf("Total head movement is %d", TotalHead
Moment);
return 0;
}
```

OUTPUT

```
Enter the number of requests 5
Enter the requests sequence
 98  183  37  122  14
Enter the initial Head position
 53
Enter total disk size
 200
Enter the head movement direction for
high 1 and for low 0  1
Total head Movement is 359
```

IBM22CS029

15/5/24	FCFS and SJF
22/5/24	Pre-emptive & Round Robin
5/6/24	Rate Monotonic & Eorliest Deadline First
12/6/24	Producer & Consumer, Philosopher Dining
19/6/24	Banker's algorithm, Deadlock detection
3/7/24	Contigous Memory allocation Page replacement algorithm
10/7/24	Disk Scheduling

10

10

Quiz 3/7/24