Name:- Rahul Baser

Roll no:- 75

Practical no 8

Aim:-

Write C programs to implement different disk scheduling algorithms and to demonstrate different memory management schemes.(First Fit and SSTF)

Code:-(SSTF)

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int request[100], max, head;
    int totalHeadMov = 0;
    int m, d, ind;
    int c = 0;
    int a[100];
    int b[100];
    int k = 0;

    printf("\nEnter the Number of Request :");
    scanf("%d", &max);
```

```
printf("\nEnter pending reqeust :");
for(int i = 0; i < max; i++){
    scanf("%d", &request[i]);
}

printf("\nEnter starting head point :");
scanf("%d", &head);

while(max != c){
    m = 9000;
    for(int i = 0; i < max; i++){
        d = abs(request[i] - head);
        if(m > d){
            m = d;
            ind = i;
        }
    }

    a[k] = request[ind];
    b[k] = m;
    totalHeadMov += m;
    k++;
    head = request[ind];
    request[ind] = 9000;
    c++;

}
```

```
    printf("\n----------------------------------------");
    printf("\nFrom\t\tTo\t\tHead Mov\n");
    printf("----------------------------------------\n");
    for(int i = 0; i < k-1; i++){
        printf(" %d\t\t%d\t\t%d\n", a[i], a[i+1], b[i]);
    }
    printf("----------------------------------------\n");
    printf("\t  Total Head Movement : %d\n",totalHeadMov);
    printf("----------------------------------------\n");
    //93 176 42 148 27 14 180
    return 0;
}
```

```
Enter the Number of Request :7

Enter pending reqeust :93 176 42 148 27 14 180

Enter starting head point :55

-----------------------------------------
From            To              Head Mov
-----------------------------------------
 42             27              13
 27             14              15
 14             93              13
 93             148             79
 148            176             55
 176            180             28
-----------------------------------------
          Total Head Movement : 207
-----------------------------------------
```

Code:-(First Fit)

```
#include <stdio.h>
#include <stdlib.h>

int main(){
int n,m;
printf("\nEnter the number of process:-\n");
scanf("%d",&n);
int *process=(int *)calloc(n,sizeof(int));
printf("\nEnter the number of memory slots:-\n");
scanf("%d",&m);
int *memory=(int *)calloc(n,sizeof(int));
int flag=0;
int c=0;
int i,j;
printf("Enter the processes space required:-\n");
for(i=0;i<n;i++)
{
  scanf("%d",&process[i]);
}
printf("\nEnter the space which is given by the memory:-\n");
for(i=0;i<m;i++)
{
  scanf("%d",&memory[i]);
}
for(i=0;i<n;i++)
```

```
{
  for(j=0;j<m;j++)
  {
  flag=0;
    if(process[i]<=memory[j])
    {
      c=c+abs(process[i]-memory[j]);
      printf("\nProcess %d  %d ->>> slot %d size %d remainf fragments are
%d",(i+1),process[i],j+1,memory[j],abs(process[i]-memory[j]));
      memory[j]=0;
      flag=1;
      break;
    }
  }
  if(flag==0)
   {
      printf("\nProcess %d has no memory allocated!!!!!!",i+1);
    }
}
printf("\nTotal Fragmentation is %d\n",c);
}
```

```
Enter the number of process:-
4

Enter the number of memory slots:-
5
Enter the processes space required:-
20
60
70
40

Enter the space which is given by the memory:-
133
150
30
120
35

Process 1  20 ->>> slot 1 size 133 remainf fragments are 113
Process 2  60 ->>> slot 2 size 150 remainf fragments are 90
Process 3  70 ->>> slot 4 size 120 remainf fragments are 50
Process 4 has no memory allocated!!!!!!
Total Fragmentation is 253
```