SCAN (algoritmo do elevador):É um algoritmo de escalonamento de disco para determinar o movimento do braço e da cabeça do disco na manutenção de solicitações de leitura na direção preferida. A partir de posições requisitadas, o algoritmo trata o acesso seguinte como sendo aquele localizado mais próximo à posição atual da cabeça de leitura. Possui características semelhantes ao SSTF.

#include<stdio.h>  
  
int main()  
{  
    int i,j,sum=0,n;  
    int d[20];  
    int disk;   //loc of head  
    int temp, max;       
    int dloc;   //loc of disk in array  
      
    printf("Enter number of location: ");  
    scanf("%d", &n);  
      
    printf("\nEnter position of head: ");  
    scanf("%d", &disk);  
      
    printf("\nEnter elements of disk queue:\n");      
    for(i=0 ; i<n ; i++)  
    {  
        printf("\nElement n %d: ", i+1);  
        scanf("%d", &d[i]);  
    }  
          
    d[n] = disk;  
    n = n+1;  
      
     // sorting disk locations  
    for(i=0 ; i<n ; i++)     
        for(j=i ; j<n ; j++)  
            if(d[i] > d[j])  
            {  
                temp = d[i];  
                d[i] = d[j];  
                d[j] = temp;  
            }  
      
    max = d[n];  
      
    // to find loc of disc in array  
    for(i=0 ; i<n ; i++)     
        if(disk == d[i])  
        {  
            dloc = i;       
            i = n; //condition break for  
        }      
                
          
    for(i=dloc ; i>=0 ; i--)  
        printf("%d -->",d[i]);  
          
    printf("0 -->");  
    for(i=dloc+1 ; i<n ; i++)  
        printf("%d-->", d[i]);  
      
    sum = disk + max;  
    printf("\n\nmovement of total %d", sum);  
}

FCFS (first-come, first-served): Muito parecido com outros algorítmos FIFO (First In, First Out). A princípio é escolhido a primeira posição do vetor de requisições (ou seja, a primeira requisição feita para movimentação do braço). Logo em seguida, busca-se a próxima posição do vetor de requisições (independente da distância para a posição atual), e assim sucessivamente, seguindo a ordem do vetor, até a última requisição ser atendida.

#include<stdio.h>  
  
void fcfs(int size)  
{  
    int i, j, result;  
    int arrival[size];  
    int burst[size];  
    int waiting[size];  
      
    /\* initialize all values with 0 \*/  
    for(i=0; i < size; ++i)      
        waiting[i]=0;      
  
    /\* input the arrival time and burst time for each process \*/  
    for(i=0; i < size; ++i) {  
        printf("\nEnter arrival time for process %d : ", i+1);  
        scanf("%d", &arrival[i]);  
  
        printf("Enter burst time for process %d : ", i+1);  
        scanf("%d", &burst[i]);  
    }  
  
    /\* first come first serve \*/  
    for(i=1; i<size; ++i) {  
        result=0;  
          
        for(j=0; j<i; ++j)      
            result+= burst[j];  
          
        /\* waiting time = starting time - arrival time \*/  
        waiting[i] = result - arrival[i];      
    }  
  
    /\* print the waiting time \*/  
    printf("\nWaiting Time:\t");  
    for(i=0; i<size; ++i)  
         printf("%d\t", waiting[i]);  
  
    /\* average waiting time \*/  
    for(i=0; i<size; ++i)               
        result+=waiting[i];  
      
    result /= size;      
    printf("\nAverage Waiting Time:\t%d\n", result);      
}  
  
int main()  
{  
    int size;  
      
    printf("size: ");  
    scanf("%d", &size);  
      
    if(size > 0)  
        fcfs(size);  
}