A2.

```c
# include < stdio.h >
int absolute value (int);
void main ()
{
int
queue (25), n, headposistion , i, j, k, seek = 0,
max range ,
difference, temp, queue1 (20), queue2 (20),
temp1, 0, temp2 = 0;
float average seek Time;
printf ("enter the maximum range of
disk: ");
scanf ("%d", & max range);
printf ("enter the number of queue requesti"
scanf ("%d", &n);
printf ("enter the intial head position");
scanf ("%d", & headposition);
printf (enter the disk positions to be
read (queue). ");
for (i=1; i <=n; i++)
{
scanf ("%d", &temp);
if (temp > headposition)
{

queue1 (temp1) = temp;
temp1++;
3
```

```
else
  {
  queue2[temp2] = temp;
    temp2++;
  }
  }
for (i = 0; i < temp1 - 1; i++)
  {
  for (j = i-1; j < temp1; j++)
  {

    if queue1 [i] > queue1(j))
    {
    temp = queue1(i);
    queue1 (i) = queue1(j);
    queue1(j) = temp
    }
    }
    }

for (i = 0; i < temp2 - 1; i++)
  {
for (j = j+1; j < temp2; j++)
  {
  if (queue2(i) < queue2(j))
  {
```

temp = queue 2(i):
queue 2(i) = queue2(j):
queue 2(j) = temp:

   3

   3

   3

for (i = 1, j = 0; j < temp1, i++, j++)
{
    queue (i) = queue 1(j))

   3
    queue (i) = max range;
    for (i = temp 1 + 2, j = 0; j < temp 2; i++, j++)
{
    queue (i) = queue 2(j);

   3
    queue (i) = 0
    queue (0) = headposition:
    for j = 0; j <= n; j++)
{
    difference =
    absolute value (queue (i + 1) - queue (j)):
    seek = seek + difference

```c
absolute val
    printf ("Disk head moves from position
%d to %d with seek %d\n",
    queue[i], queue[i+1], difference):
}

average Seek Time = seek / (float) n;
print("Total Seek Time = %d\n", seek).
    printf ("Average Seek Time =%f\n", average
Seek Time):
}
int absolute Value (int x)
{
if (x > 0)
  {
return x;
  }
else
{
return x*-1;
}
}
```