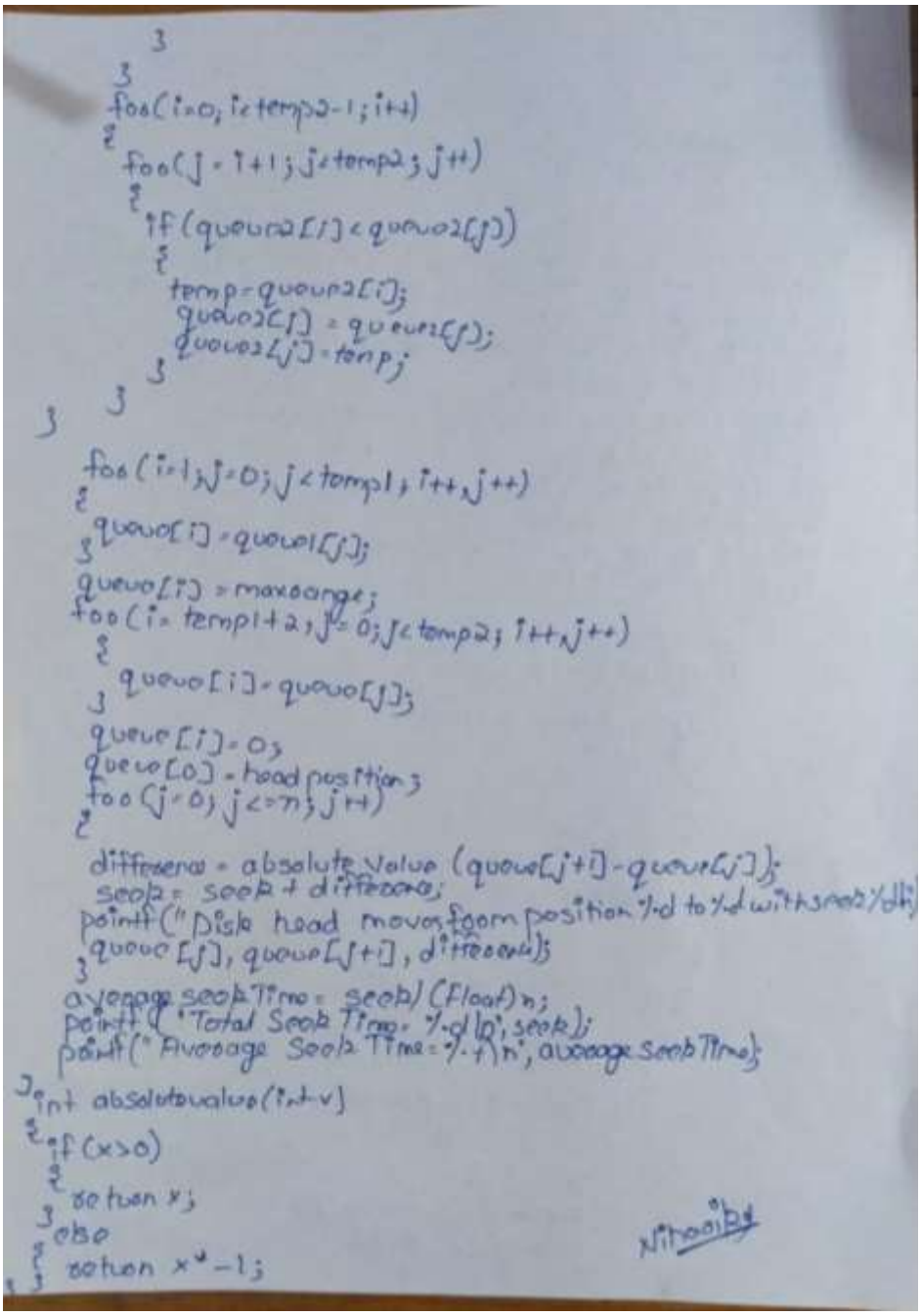
Name - Nihaarika Sharma
Course - BSC.IT A
Student id - 20052076
University Roll no - 2023071

```
#include<stdio.h>
int absoluteValue(int);
void main()
{
 int queue[25],n,headposition,i,j,k,seek=0,maxrange,
 difference, temp, queue1[20], queue2[20], temp1=0, temp2=0;
 float averageSeekTime;
 printf("Enter the maximum range of disk:");
 scanf("%d",&maxrange);
 printf("Enter the number of queue requests:");
 scanf("%d",&n);
 printf("Enter the inital 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++;
       }
       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;
           }
```

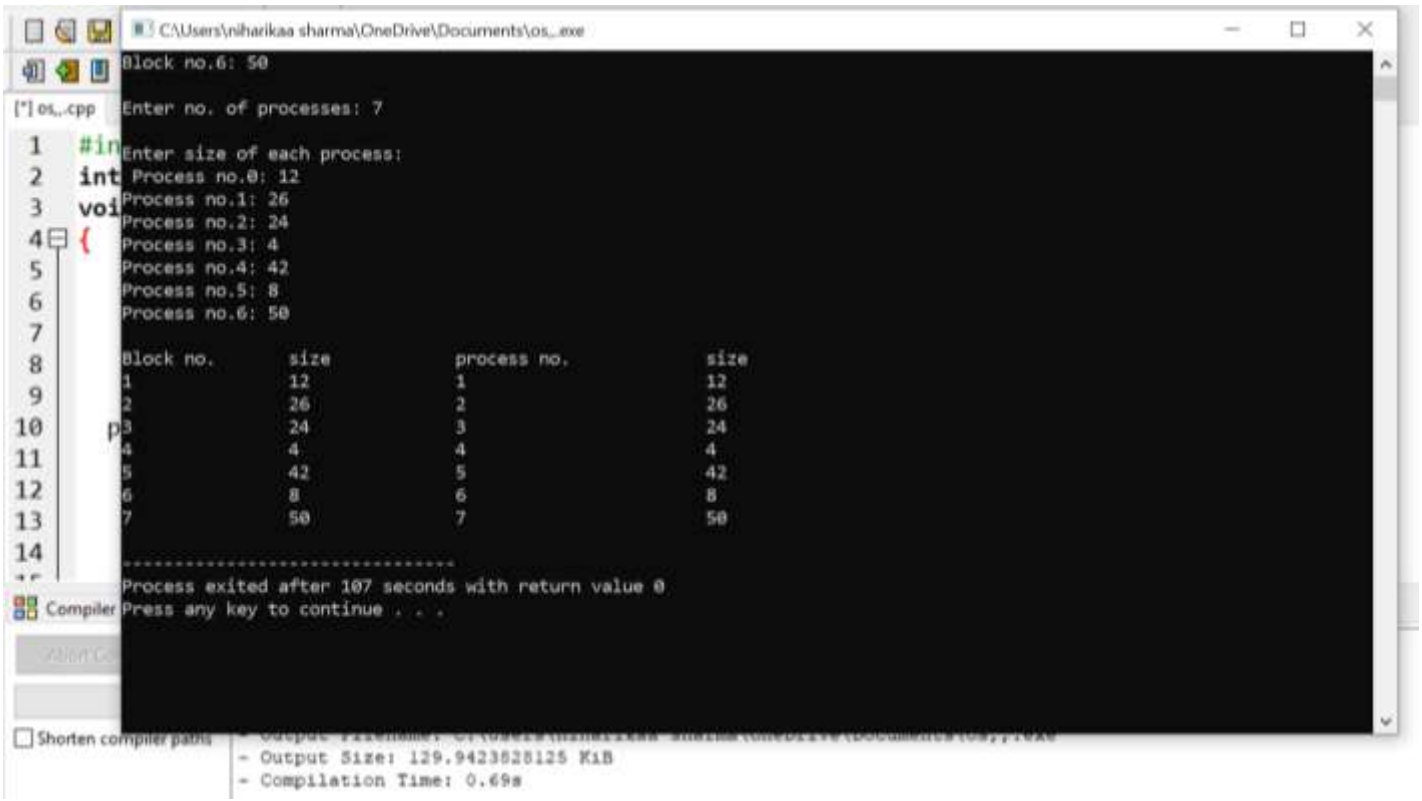
Nihaarika



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

    for(i=1,j=0; j<temp1; i++,j++)
    {
    queue[i] = queue1[j];
    }
    queue[i] = maxrange;
    for(i=temp1+2, j=0; j<temp2; i++,j++)
    {
      queue[i]=queue2[j];
    }
    queue[i]=0;
    queue[0]=head position;
    for(j=0; j<=n; j++)
    {
      difference = absolute value (queue[j+1]-queue[j]);
      seek = seek + difference;
      printf("Disk head moves from position %d to %d with seek %d\n",
      queue[j], queue[j+1], difference);
    }
    average seek Time = seek/(float)n;
    printf("Total Seek Time= %d\n", seek);
    printf("Average Seek Time= %f\n", average seek Time);
}
int absolutevalue(int v)
{
  if(x>0)
  {
    return x;
  }
  else
  {
    return x*-1;
  }
}
```

Nitesh Bk


C:\Users\niharikaa sharma\OneDrive\Documents\os_.exe
Block no.6: 50
Enter no. of processes: 7
Enter size of each process:
Process no.0: 12
Process no.1: 26
Process no.2: 24
Process no.3: 4
Process no.4: 42
Process no.5: 8
Process no.6: 50
Block no. size process no. size
1 12 1 12
2 26 2 26
3 24 3 24
4 4 4 4
5 42 5 42
6 8 6 8
7 50 7 50
Process exited after 107 seconds with return value 0
Press any key to continue . . .
Compiler
Shorten compiler paths
- Output Size: 129.9423828125 KiB
- Compilation Time: 0.69s