```cpp
#include<iostream>
#include<fstream>
#include<cstdlib>
#include <sstream>
#include<string>

using namespace std;


typedef int item;
struct integers{
    item data;
    integers* next;
};
integers* headlist;
typedef integers* intptr;


void delete_integer(int data,intptr &headlist);
void insert_item(int data,intptr &headlist);
void max_value(intptr &headlist,int x);
void bubble_sort(intptr &headlist);
void swap(intptr &headlist);
void greater_thold(intptr &headlist,int threshold,int x);
void average_list(intptr &headlist);
bool if_ordered(intptr &headlist);
void add_to_list(item number,intptr &headlist);
void read_file(int x);
void write_file(int x,intptr &headlist);

int main(){

    ifstream infile;
    infile.open("CommandFile.txt");

    if(!infile.is_open()){
        cout<<"could not open file"<<endl;
        exit(EXIT_FAILURE);
        }
```

```
    int file_number;
    char command;

    while(infile>>command){
        if(command=='r'){
            infile>>file_number;
            read_file(file_number);
        }
        else if(command=='s'){
            bubble_sort(headlist);
        }
        else if(command=='w'){
            write_file(file_number,headlist);
        }
        else if(command=='i'){
            int inserted_number;
            infile>>inserted_number;
            insert_item(inserted_number,headlist);
        }
        else if(command=='d'){
            int deleted_number;
            infile>>deleted_number;
            delete_integer(deleted_number,headlist);
        }
        else if(command=='e'){
            int threshold;
            infile>>threshold;
            greater_thold(headlist,threshold,file_number);
        }
        else if(command=='m'){
            max_value(headlist,file_number);
        }
        else if(command=='a'){
            average_list(headlist);
        }
    }
    infile.close();

return 0;
```

```
}



void insert_item(int data,intptr &headlist){

    bool sorted=false;
    bool found=false;

    intptr searchptr,lastptr,newptr;

 newptr=new integers;
 newptr->data=data;
 newptr->next=NULL;

 sorted=if_ordered(headlist);


if(sorted==true){

    if(headlist==NULL){
        headlist=newptr;
        return;
    }
    else if(headlist->data>=data){
        newptr->next=headlist;
        headlist=newptr;
    }
    else{
        found=false;
        searchptr=headlist;
        lastptr=headlist;
        while((searchptr!=NULL)&&(! found)){
                if(searchptr->data>=data){
                    found=true;
                }
                else{
                    lastptr=searchptr;
                    searchptr=searchptr->next;
```

```
                }
        }
        newptr->next=searchptr;
        lastptr->next=newptr;
    }
}
else{
    newptr->next=headlist;
    headlist=newptr;

}

}




void delete_integer(int data,intptr &headlist){

bool found=false;
intptr searchptr,lastptr,oldptr;

if(headlist==NULL){
            return;
}
else if(headlist->data==data){
        oldptr=headlist;
        headlist=headlist->next;
        delete oldptr;
}
else{
        found=false;
        searchptr=headlist;
        lastptr=headlist;
while((searchptr!=NULL)&&(! found)){
        if(searchptr->data==data){
            found=true;
            lastptr->next=searchptr->next;
            delete searchptr;
        }
```

```
        else{
            lastptr=searchptr;
            searchptr=searchptr->next;
        }
     }
   }
}



void max_value(intptr &headlist,int x){
    intptr searchptr;
    int max=headlist->data;
    searchptr=headlist;

while(searchptr!=NULL){
    if(searchptr->data>=max){
        max=searchptr->data;
    }
    searchptr=searchptr->next;
  }
        ofstream outfile;
        string a;
        stringstream ss;
        ss<<x;
        ss>>a;

        string filename="outfile_";
        filename+=a;
        filename+=".txt";
        outfile.open(filename.c_str(),ios::app);


        if(!outfile.is_open()){
            cout<<"could not open file"<<endl;
            exit(EXIT_FAILURE);
            }

         outfile<<"maximum value is:"<<max<<endl;
```

```
        outfile.close();
}




void swap(intptr &headlist){
    intptr secondptr=headlist->next;
    int temp;

temp=secondptr->data;
secondptr->data=headlist->data;
headlist->data=temp;
}




void bubble_sort(intptr &headlist){

    intptr countptr=headlist;
    intptr secondptr=headlist;
    int counter=0;

    while(countptr!=NULL){
        countptr=countptr->next;
        counter++;
    }

for(int i=1;i<counter;i++){
    for(int j=1;j<counter;j++){
        if(secondptr->data>secondptr->next->data){
            swap(secondptr);
        }
        secondptr=secondptr->next;
    }
    secondptr=headlist;
    }
```

```
}



void  greater_thold(intptr &headlist,int threshold,int x){

    intptr searchptr=headlist;
    int counter=0;
    while(searchptr!=NULL){
        if(searchptr->data>threshold){
            counter++;
        }
        searchptr=searchptr->next;
    }

        ofstream outfile;
        string a;
        stringstream ss;
        ss<<x;
        ss>>a;

        string filename="outfile_";
        filename+=a;
        filename+=".txt";
        outfile.open(filename.c_str(),ios::app);

        if(!outfile.is_open()){
            cout<<"could not open file"<<endl;
            exit(EXIT_FAILURE);
            }
         outfile<<"number of elements with value with greater than T
is:"<<counter<<endl;

        outfile.close();

}
```

```
void average_list(intptr &headlist){


intptr firstptr,secondptr;
firstptr=headlist;
secondptr=headlist->next;

if(firstptr==NULL){
    return;
}


while(secondptr!=NULL){

        intptr newptr;
        int average=0;

        newptr=new integers;
        newptr->data=average;
        newptr->next=NULL;

        newptr->data=(firstptr->data+secondptr->data)/2;
        newptr->next=secondptr;
        firstptr->next=newptr;


        firstptr=newptr->next;
        secondptr=firstptr->next;

        }
}



bool if_ordered(intptr &headlist){

    intptr firstptr,secondptr;
    firstptr=headlist;
    secondptr=headlist;
```

```
    bool ordered=true;

    while((secondptr!=NULL)&&(ordered==true)){
        if(secondptr->data<firstptr->data){
            ordered=false;
        }
        else{
            firstptr=secondptr;
            secondptr=secondptr->next;
        }
    }
    return ordered;
}



void add_to_list(item number,intptr &headlist){

intptr newptr=new integers;

newptr->data=number;
newptr->next=headlist;
headlist=newptr;
}


void read_file(int x){

    ifstream infile;
    string a;
    stringstream ss;
    ss<<x;
    ss>>a;

    string filename="data_";
    filename+=a;
    filename+= ".txt";
    infile.open(filename.c_str());
```

```
    if(!infile.is_open()){
    cout<<"could not open file"<<endl;
    exit(EXIT_FAILURE);
    }
    int n=0;

    while(infile>>n){

        add_to_list(n,headlist);
    }
    infile.close();

}


void write_file(int x,intptr &headlist){

    ofstream outfile;
    string a;
    stringstream ss;
    ss<<x;
    ss>>a;

    string filename="outfile_";
    filename+=a;
    filename+=".txt";
    outfile.open(filename.c_str(),ios::app);

    if(!outfile.is_open()){
        cout<<"could not open file"<<endl;
        exit(EXIT_FAILURE);
        }

    intptr tempHeadlist=headlist;
    while(headlist!=NULL){

     outfile<<headlist->data<<endl;
      headlist=headlist->next;
        }
```

```
headlist=tempHeadlist;

outfile.close();
}
```