```cpp
// simulation.cpp
// Finnian Allen & Makoto Kewish
// This program runs a simulation of a magnetic disk
// that takes in requests and serves them as a queue
// December 2, 2018

#include "queue.h"
#include "itemtype.h"
#include <iostream>
#include <fstream>
using namespace std;

// This function opens and closes the input file and simulates the program
// Pre: No preconditions
// Post: Calls all necessary functions to perform program
// usage: simulate()
void simulate();

// This function opens the input file and displays a message accordingly
// Pre: there must be a file present
// Post: the file is opened and a relavent message is displayed
// usage: openFile(infile)
void openFile(ifstream& infile);

// This function closes the input file
// Pre: The file must have been opened
// Post: The file is closed
// usage: closeFile(infile)
void closeFile(ifstream& infile);

// This function takes in the input and adds it to the queue
// Pre: the file must have been opened and the queue must have been initialized
// Post: the input is added to the queue and printed to the screen
// usage: makeRequest(infile, disk)
void makeRequest(ifstream& infile, Queue& disk);

// This function takes in the queue and determines who is served and serves them
// Pre: queue most have been initialized
// Post: What is being served is output and the queue is served
// usage: serve(disk);
void serve(Queue& disk);

int main()
{

   simulate();

   return 0;
}

// This function opens and closes the input file and simulates the program
// Pre: No preconditions
// Post: Calls all necessary functions to perform program
// usage: simulate()
void simulate()
{
   ifstream infile;
   Queue disk;

   openFile(infile);
   makeRequest(infile, disk);
   serve(disk);
   makeRequest(infile, disk);
   makeRequest(infile, disk);
   serve(disk);
   makeRequest(infile, disk);
   makeRequest(infile, disk);
   makeRequest(infile, disk);
   serve(disk);
```

```
70      serve(disk);
71      makeRequest(infile, disk);
72      serve(disk);
73
74      closeFile(infile);
75      }
76
77  // This function opens the input file and displays a message accordingly
78  // Pre: there must be a file present
79  // Post: the file is opened and a relavent message is displayed
80  // usage: openFile(infile)
81  void openFile(ifstream& infile)
82  {
83      infile.open("requests.dat");
84      if(infile.is_open())
85      {
86          cout << "File opened successfully" << endl;
87      } else {
88          cout << "File failed to open" << endl;
89      }
90  }
91
92  // This function closes the input file
93  // Pre: The file must have been opened
94  // Post: The file is closed
95  // usage: closeFile(infile)
96  void closeFile(ifstream& infile)
97  {
98      infile.close();
99  }
100
101 // This function takes in the input and adds it to the queue
102 // Pre: the file must have been opened and the queue must have been initialized
103 // Post: the input is added to the queue and printed to the screen
104 // usage: makeRequest(infile, disk)
105 void makeRequest(ifstream& infile, Queue& disk)
106 {
107    bool isNotFull;
108
109    ItemType newItem;
110    infile >> newItem;
111    disk.lineUp(newItem, isNotFull);
112    cout << "Reading request: " << endl << disk << endl;
113 }
114
115 // This function takes in the queue and determines who is served and serves them
116 // Pre: queue most have been initialized
117 // Post: What is being served is output and the queue is served
118 // usage: serve(disk);
119 void serve(Queue& disk)
120 {
121     bool isNotEmpty;
122     cout << "Serving: " << disk.getWhoIsServed() << endl;
123     disk.getServed(isNotEmpty);
124     cout << endl;
125 }
```