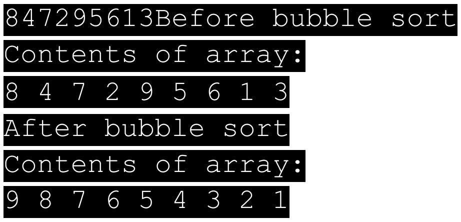
```
#include <iostream>
#include <fstream>
using namespace std;

//Reads data from a text file into an array
int readData(int * &arr) {
// Open the file to get the data
int size;
ifstream inputFile;
inputFile.open("data.txt");
// The first line of the file is the size of the array
inputFile >> size;

//Allocate the memory for pointer arr
    arr =  new int[size];
// Read the rest of the data into the array
for (int i = 0; i < size; i++) {
    inputFile>>*(arr +i);
    cout<<*(arr+i);
} // End for loop
inputFile.close();
return size;
} // End readData

void bsort(int * arr, int size) {
    int i,j;
    int temp;

    for(i = 1; i<size; i++){

        for(j = 0; j<size-i; j++){

            if(*(arr + j)< *(arr + j + 1)){
                temp = *(arr + j);
                *(arr + j) = *(arr + j + 1);
                *(arr + j + 1) = temp;
            }
        }
    }
}

void writeToConsole(int * arr, int size) {
cout << "Contents of array:" << endl;
for (int i = 0; i < size; i++) {
```

```
cout << *(arr + i) << " ";
} // End for loop
} // End writeToConsole


int main() {
    int * arr;
    int size = readData(arr);
  cout << "Before bubble sort" << endl;
  writeToConsole(arr, size);
  cout<<endl;
  bsort(arr, size);
  cout << "After bubble sort" << endl;
  writeToConsole(arr, size);

      return 0;
}
```

////////////////////OUTPUT//////////////////



```
847295613Before bubble sort
Contents of array:
8 4 7 2 9 5 6 1 3
After bubble sort
Contents of array:
9 8 7 6 5 4 3 2 1
```

PART 2

```
//
//  main.cpp
//  Lab6b
//
//  Created by Michael Hannigan on 10/12/20.
//

// linklist.cpp
// linked list
#include <iostream>
using namespace std;
////////////////////////////////////////////////////////////////
struct link //one element of list
{
int data; //data item
link* next; //pointer to next link
};
```

```
////////////////////////////////////////////////////////////////
class linklist //a list of links
{
private:
link* first; //pointer to first link
public:
linklist() //no-argument constructor
{ first = NULL; } //no first link
void additem(int d); //add data item (one link)
void display(); //display all links
};
//--------------------------------------------------------------
void linklist::additem(int d) //add data item
{
link* newlink = new link; //make a new link
newlink->data = d; //give it data
newlink->next = first; //it points to next link
first = newlink; //now first points to this
}
//--------------------------------------------------------------
void linklist::display() //display all links
{
   link* current = first;
   while(current!=NULL){
      cout<<current->data<<endl;
      current = current->next;
   }
}
////////////////////////////////////////////////////////////////
int main()
{
linklist li; //make linked list

li.additem(25); //add four items to list
li.additem(36);
li.additem(49);
li.additem(64);

li.display(); //display entire list
return 0;
}

///////////////////////OUTPUT///////////////////
64
49
36
25
```