```
***************************************************
*  PROGRAMMED BY : ALI ESHGHI
*  CLASS         : CS1B
*  SECTION       : MW: 7:30p - 9:50p
*  LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
***************************************************

*************************************
* WELCOME TO THE SHEEP LIST MANAGER *
*************************************

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 1
Sheep name: Fluffy
Sheep Age:  1


The Sheep...
Sheep Name: Fluffy
Sheep Age:  1
Has been added

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 2
NAME           AGE
-------------- ---
Fluffy          1


Is at the front of the list!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 1
Sheep name: Maa
Sheep Age:  3


The Sheep...
Sheep Name: Maa
Sheep Age:  3
Has been added

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 4
There are 2 sheeps in the list

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 5
<output using the array>
NAME           AGE
-------------- ---
Fluffy          1
Maa             3


<output using the linked list>
NAME           AGE
-------------- ---
Fluffy          1
Maa             3
There are 2 sheeps in the list

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 1
Sheep name: Baa Baa
Sheep Age:  2


```

```
The Sheep...
Sheep Name: Baa Baa
Sheep Age:  2
Has been added

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 5
<output using the array>
NAME           AGE
-------------- ---
Fluffy          1
Maa             3
Baa Baa         2


<output using the linked list>
NAME           AGE
-------------- ---
Fluffy          1
Maa             3
Baa Baa         2
There are 3 sheeps in the list

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 4
There are 3 sheeps in the list

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 3
Who are you looking for? Baa Baa

NAME           AGE
-------------- ---

```

```
Baa Baa         2

Has Been Found!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 6

The list has been cleared!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 6

The list is empty

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 5

The list is empty


SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 4

The list is empty


SHEEP LIST MANAGER
```

```
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 3

There are no sheep to be found!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 2
Nobody is in front -the list is empty!

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 7

**** The number 7 is an invalid entry     ****
**** Please input a number between 0 or 6 ****

SHEEP LIST MANAGER
1 - Add Sheep
2 - Output 1st Sheep
3 - Find Sheep
4 - List Size
5 - Output List
6 - Clear list
0 - Exit
Enter a command: 0
```

```
/************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
*************************************************************/

#ifndef MYHEADER_H_
#define MYHEADER_H_

#include<iostream>
#include<iomanip>
#include<string>
#include<fstream>
#include<limits>
#include<sstream>
#include "ClassHeader.h"
using namespace std;

/********************************************************
* CONSTANTS
* ----------------------------------------------------
* USED FOR CLASS HEADING - ALL WILL BE OUTPUT
* ----------------------------------------------------
* Type     : Program Type
* LAB_NUM  : Lab Number (specific to this lab)
* LAB_NAME : Title of the Lab
*********************************************************/

const string NAME = "LAB 13 - ARRAYS AND LINKED LIST (OOP)";
const char   TYPE    = 'L';
const int    NUM     = 13 ;
const string CLASS   = "CS1B";
const string SECTION = "MW: 7:30p - 9:50p";

/************************************************************
 * Function - PrintHeaderFile
 * ----------------------------------------------------------
 * This function will output the class heading to the screen.
 *
 * return type - nothing
 *               the function is void type
 ************************************************************/
void PrintHeader();

/************************************************************
 * Function - Menu
 * ----------------------------------------------------------
 * This function will outputs the menu and prompts the user
 * to choose an option from the menu
 *
 * return type - integer
 *               the function is int type
 ************************************************************/
int Menu();

#endif /* MYHEADER_H_ */

```

```cpp
/************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
*************************************************************/
#include "MyHeader.h"
#include "Classheader.h"

/************************************************************
 * LAB 13 - ARRAYS AND LINKED LIST (OOP)
 *_________________________________________________________
 * This program will use the arrays and linked list as an
 * objects of a class and gets the information base of the
 * users choice and add those information to the list and
 * prompts a menu for the user who can have more options
 *_________________________________________________________
 * INPUT:menuOption -> user input for the menu
 *       name       -> name of a new sheep
 *       age    -> age of a new sheep
 *
 *
 * OUTPUT:name      -> name of the sheeps in the list
 *        age       -> age of the sheeps in the list
 *        size      -> how many sheeps are in the list
 *        search-> user input name to search in the list
 *        if the list is empty or no
 ************************************************************/

int main()
{

    /*************
     *  VARIABLE *
     *************/
    int     menuOption;
    Animal  sheep;
    string  name;
    string  search;
    int     age;
    int     size;


    //this function will output the class header
    PrintHeader();



    cout << "*************************************" << endl;
    cout << "* WELCOME TO THE SHEEP LIST MANAGER *" << endl;
    cout << "*************************************" << endl << endl;

    //do while loop for menu
    do
    {
        //this function prompts the user the menu and gets the user input
        menuOption = Menu();


        //first option that adds the sheep to the menu
        if(menuOption == 1)
        {
            cin.ignore(10000,'\n');
            cout << "Sheep name: ";
            getline(cin,name);
            cout << "Sheep Age:  ";
            cin  >> age;

            sheep.AddSheep(name, age);
            sheep.AddSheepLinkedList(name, age);
        }

        //second option that displays the first sheep in the list
        else if(menuOption == 2)
        {
            sheep.DisplayFirstSheep();
        }

        //third option that searches for a specific sheep
        else if(menuOption == 3)
        {
            cin.ignore(10000,'\n');
            cout << "Who are you looking for? ";
            getline(cin,search);
            sheep.FindSheep(search);
        }

        //fourth option that outputs how many sheeps are in the list
        else if(menuOption == 4)
        {
            size = sheep.ListSize();
            cout << "There are " << size << " sheeps in the list"
                                 << endl << endl;
        }

        //fifth option that displays the members of the list
        else if(menuOption == 5)
        {
            sheep.Display();

            size = sheep.ListSize();
            cout << "There are " << size << " sheeps in the list"
                                 << endl << endl;
        }

        //sixth option that clears the list (deconstruction)
        else if(menuOption == 6)
        {
            sheep.~Animal();

        }


    }while(menuOption != 0);


    return 0;


}

```

```cpp
/*************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
**************************************************************/

#include"MyHeader.h"

/*************************************************************
 * Function - Menu
 * ----------------------------------------------------------
 * This function will outputs the menu and prompts the user
 * to choose an option from the menu
 *
 * return type - integer
 *               the function is int type
 *************************************************************/

int Menu()
{

    /*************
     *  VARIABLE *
     *************/

    int menuOption;
    bool checkInp;

    /*******************
     *  INITIALIZATION *
     *******************/

    checkInp = false;

    //do while loop for input check
    do
    {

    cout << "SHEEP LIST MANAGER" << endl;
    cout << "1 - Add Sheep" << endl;
    cout << "2 - Output 1st Sheep" << endl;
    cout << "3 - Find Sheep" << endl;
    cout << "4 - List Size" << endl;
    cout << "5 - Output List" << endl;
    cout << "6 - Clear list" << endl;
    cout << "0 - Exit" << endl;
    cout << "Enter a command: ";



    //CHECKS FOR THE CHAR INPUT

    if (!(cin >> menuOption))
    {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');

        cout << endl;
        cout << "**** Please input a NUMBER between 0 or 6 ****";
        cout << endl << endl;

        checkInp = false;

    }

    //CHECKS FOR THE RANGE ERROR

    else if (menuOption > 6 || menuOption < 0 )
    {

        cout << endl;
        cout << "**** The number "              << menuOption
        << " is an invalid entry     ****" << endl;
        cout << "**** Please input a number between 0 or 6 ****";
        cout << endl << endl;

        checkInp = false;

    }

    checkInp = true;

    }while(!checkInp);


    return menuOption;
}



```

PrintHeader.cpp

```cpp
/*************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
*************************************************************/

#include "MyHeader.h"

/*************************************************************
 * Function - PrintHeaderFile
 * ----------------------------------------------------------
 * This function will output the class heading to the screen.
 *
 * return type - nothing
 *               the function is void type
 ************************************************************/

void PrintHeader()
{
    cout << left;
    cout << "***************************************************\n"        ;
    cout << "*  PROGRAMMED BY : ALI ESHGHI"                           ;
    cout << "\n*  "     << setw(14) << "CLASS"     << ": " << CLASS       ;
    cout << "\n*  "     << setw(14) << "SECTION"   << ": " << SECTION     ;
    cout << "\n*  LAB #"<< setw(9)  << NUM  << ": " << NAME           ;
    cout << "\n***************************************************\n\n"    ;
    cout << right;
}


```

```
/************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
************************************************************/

#ifndef CLASSHEADER_H_
#define CLASSHEADER_H_

#include<iostream>
#include<iomanip>
#include<string>
#include<fstream>
#include<limits>
#include<sstream>
using namespace std;

const int AR_SIZE = 50;


class Animal
{
    //public part of the class that is available for outside of the class
    public:
        //constructor
        Animal();

        //decosntructor
        ~Animal();

        //method for adding a new sheep and its age to parallel arrays
        void AddSheep(string name, int age);

        //method for adding a new sheep to a linked list
        void AddSheepLinkedList(string name, int age);

        //method for showing the first sheep from the list
        void DisplayFirstSheep();

        //method that returns the size of the list of the sheeps
        int ListSize() const;

        //method  for outputting the objects
        void Display() const;

        //method for finding the sheep in the list
        void FindSheep(string) const;


    //private part only available for the class
    private:
        string nameAr[AR_SIZE];
        int    ageAr[AR_SIZE];
        string name;
        int    age;
        int    listSize;

        struct SheepNode
        {
            string     sheepName;
            int        sheepAge;
            SheepNode  *next;
        };
        SheepNode *head;

};



#endif /* CLASSHEADER_H_ */

```

```
/************************************************************
* AUTHOR    : Ali Eshghi
* STUDENT ID    : 1112261
* LAB #13       : LAB 13 - ARRAYS AND LINKED LIST (OOP)
* CLASS         : CS 1B
* SECTION       : MW - 7:30 pm - 9:50 pm
* DUE DATE      : 12/3/2019
************************************************************/

#include "MyHeader.h"
#include "ClassHeader.h"


Animal::Animal()/*** CONSTRUCTOR ***/
{
    /*******************
     *  INITIALIZATION *
     *******************/

    name.clear();
    age  = 0;
    listSize = 0;
    head = NULL;
}

Animal::~Animal()    /*** DESTRUCTOR ***/
{

    /*************
     *  VARIABLE *
     *************/

    SheepNode *sheepPtr;

    if(head != NULL)
    {
        //clear the list
        sheepPtr = head;
        while(sheepPtr != NULL)
        {
            head = head -> next;
            delete sheepPtr;

            sheepPtr = head;
        }

        for(int i = 0; i < AR_SIZE; i++)
        {
            nameAr[i] = ' ';
        }
        cout << "The list has been cleared!" << endl << endl;
    }

    else if(head == NULL)
    {
```

```cpp
        cout << "\nThe list is empty" << endl << endl;
    }

}


//method for adding a new sheep and its age to parallel arrays
void Animal::AddSheep(string name, int age)
{
    if(listSize < AR_SIZE)
    {
        nameAr[listSize] = name;
        ageAr[listSize]  = age;

        listSize++;

        cout << endl << endl;
        cout << "The Sheep..."           << endl;
        cout << "Sheep Name: "   << name << endl;
        cout << "Sheep Age:  "   << age  << endl;
        cout << "Has been added" << endl << endl;
    }

    else
    {
        cout << "could not add new animal, list is full..." << endl;
    }

}

//method for adding a new sheep to a linked list
void Animal::AddSheepLinkedList(string name, int age)
{
    /*************
     *  VARIABLE *
     *************/

    SheepNode *newSheepNode;
    SheepNode *tail;

    /*******************
     *  INITIALIZATION *
     *******************/

    newSheepNode = new SheepNode;

    /*** ADD TO THE TAIL ***/

    //check if there is memory for new node
    if(newSheepNode != NULL)
    {
        newSheepNode -> sheepName = name;
        newSheepNode -> sheepAge  = age;


        //check if list is empty
        if(head != NULL)
        {
            tail = head;

            //find the tail
            while(tail != NULL)
            {
                tail = tail -> next;
            }

            tail -> next = newSheepNode;
        }

        else
        {
            head = newSheepNode;
        }

        listSize++;
    }

    else
    {
        cout << "Could not add to the list - out of memory";
    }
}

//method that returns the size of the list of the sheeps
int Animal::ListSize() const
{
    if(head != NULL)
    {
        return listSize;
    }
    else if(head == NULL)
    {
        cout << "\nThe list is empty" << endl << endl;
        return 0;
    }

}

//method for showing the first sheep from the list
void Animal::DisplayFirstSheep()
{
    if(head != NULL)
    {
        cout << left;
        cout << setw(15) << "NAME" << "AGE" <<endl;
        cout << setw(15) << "--------------" << "---" << endl;
        cout << setw(16) << nameAr[0] << ageAr[0];
        cout << endl << endl << endl;

        cout << "Is at the front of the list!" << endl << endl;
    }

    else if(head == NULL)
    {
        cout << "Nobody is in front -the list is empty!" << endl << endl;
    }


}

//method for finding the sheep in the list
void Animal::FindSheep(string search) const
{
    /*************
     *  VARIABLE *
     *************/

    int   i;
    bool  stat;

    /******************
     *  INITIALIZATION *
     ******************/

    i    = 0;
    stat = false;

    if(head != NULL)
    {
        while(i < AR_SIZE && stat)
        {
            if(nameAr[i] == search)
            {
                stat = true;
            }

            else
            {
                i++;
            }

            if(stat == true)
            {
                cout << setw(15) << "NAME" << "AGE" <<endl;
                cout << setw(15) << "--------------" << "---" << endl;
                cout << setw(16) << nameAr[i] << ageAr[i] << endl << endl;
                cout << "Has Been Found";
            }
        }
    }

    else if(head == NULL)
    {
        cout << "There are no sheep to be found!" << endl << endl;
    }
```

```cpp

}


//method  for outputting the objects
void Animal::Display() const
{
    /*************
     *  VARIABLE *
     *************/

    SheepNode *sheepPtr;

    if(head != NULL)
    {
        cout << "<output using the array>" << endl;
        cout << left;
        cout << setw(15) << "NAME" << "AGE" <<endl;
        cout << setw(15) << "---------------" << "---" << endl;

        for(int index = 0; index < listSize; index++)
        {
            cout << setw(16) << nameAr[index] << ageAr[index] << endl;
        }
        cout << endl << endl;


        cout << "<output using the linked list>" << endl;
        cout << left;
        cout << setw(15) << "NAME" << "AGE" <<endl;
        cout << setw(15) << "---------------" << "---" << endl;

        for(sheepPtr = head -> next; sheepPtr != NULL; sheepPtr = sheepPtr -> next)
        {
            cout << setw(16) << sheepPtr->sheepName << sheepPtr->sheepAge << endl;
        }

    }

    else if(head == NULL)
    {
        cout << "\nThe list is empty" << endl << endl;
    }

}




```