28.1 LAB: Find max in list

This section has been set as optional by your instructor.

Given the IntNode class, define the FindMax() function to return the largest value in the list or -99 if the list is empty. Assume all values in the list are non-negative.

Ex: If the list contains:

```
head -> 14 -> 191 -> 186 -> 181
```

FindMax(headNode) returns 191.

Ex: If the list contains:

```
head ->
```

FindMax(headNode) returns -99.

LAB ACTIVITY | 28.1.1: LAB: Find max in list | 0 / 10

main.cpp — Load default template...

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

class IntNode {
public:
   // Constructor
   IntNode(int dataInit);

   // Get node value
   int GetNodeData();

   // Get pointer to next node
   IntNode* GetNext();

   /* Insert node after this node.
```


Develop mode | Submit mode

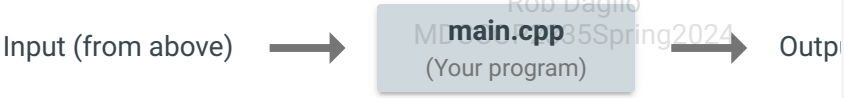
Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first

box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program



Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```

28.2 LAB: Index of list item

This section has been set as optional by your instructor.

Given the IntNode class, define the IndexOf() function to return the index of parameter `target` or -1 if not found.

Note: The first index after the head node is 0.

Ex: If the list contains:

```
head -> 14 -> 191 -> 22 -> 99
```


IndexOf(headNode, 22) returns 2.

Ex: If the list contains:

```
head ->
```

IndexOf(headNode, 22) returns -1.

LAB ACTIVITY | 28.2.1: LAB: Index of list item | 0 / 10

main.cpp

```
1 Loading latest submission...
```


Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.cpp** (Your program) → Outp

Program output displayed here

Coding trail of your work What is this?

Retrieving signature

28.3 LAB: Is list sorted

This section has been set as optional by your instructor.

Given the IntNode class, define the IsSorted() function that takes the head node of a linked list as a parameter and determines if the numbers in the list are in ascending order. IsSorted() returns true if the list is in ascending order, has only one item, or is empty; otherwise, isSorted() returns false.

Ex: If the list contains:

```
head -> 14 -> 19 -> 22 -> 99
```

IsSorted(headNode) returns true.

Ex: If the list contains:

```
head -> 14 -> 19 -> 22 -> 99 -> 14 -> 100
```

IsSorted(headNode) returns false.

Ex: If the list contains:

```
head ->
```

IsSorted(headNode) returns true.

LAB ACTIVITY | 28.3.1: LAB: Is list sorted | 0 / 10

main.cpp — Load default template...

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

class IntNode {
public:
   // Constructor
   IntNode(int dataInit);

   // Get node value
   int GetNodeData();

   // Get pointer to next node
```

```
13    IntNode* GetNext();
14
15    /* Insert node after this node.
```

Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → main.cpp (Your program) → Outp

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working on this zyLab.
```

28.4 LAB: List count

This section has been set as optional by your instructor.

Given the IntNode class, define the GetCount() function that returns the number of items in the list not including the head node.

Ex: If the list contains:

```
head -> 14 -> 19 -> 4
```

GetCount(headNode) returns 3.

Ex: If the list contains:

```
head ->
```

GetCount(headNode) returns 0.

539740.3879454.qx3zqy7

LAB ACTIVITY | 28.4.1: LAB: List count | 0 / 10

main.cpp — Load default template...

```
1 Loading latest submission...
```

Develop mode | **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.cpp** (Your program) → Outp

Program output displayed here

Coding trail of your work [What is this?](#)

```
History of your effort will appear here once you begin working
on this zyLab.
```

28.5 LAB: Warm up: Contacts

This section has been set as optional by your instructor.

A linked list is built in this lab. Make sure to keep track of the head node.

Step 1: Define class `ContactNode` per the following specifications:

- Private data members
  - `string contactName`
  - `string contactPhoneNumber`
  - `ContactNode* nextNodePtr`
- Constructor with parameters for name followed by phone number (1 pt)
- Public member functions
  - `GetName()` - Accessor (1 pt)
  - `GetPhoneNumber()` - Accessor (1 pt)
  - `InsertAfter(ContactNode* newNode)` - Insert newNode after the current node (2 pts)
  - `GetNext()` - Accessor (1 pt)
  - `PrintContactNode()` - Output contactName and contactPhoneNumber of the current node according to the format shown in the example below.

Ex: If the name is Roxanne Hughes and the phone number is 443-555-2864, PrintContactNode() outputs:

```
Name: Roxanne Hughes
Phone number: 443-555-2864
```


Step 2: Define main() to read the name and phone number for three contacts and output each

contact. Create three ContactNodes and use the nodes to build a linked list. (2 pts)

Ex: If the input is:

```
Roxanne Hughes
443-555-2864
Juan Alberto Jr.
410-555-9385
Rachel Phillips
310-555-6610
```


the output is:

```
Person 1: Roxanne Hughes, 443-555-2864
Person 2: Juan Alberto Jr., 410-555-9385
Person 3: Rachel Phillips, 310-555-6610
```

Step 3: Output the linked list. Use a loop to call each node's PrintContactNode(). (2 pts)

Ex:

```
CONTACT LIST
Name: Roxanne Hughes
Phone number: 443-555-2864

Name: Juan Alberto Jr.
Phone number: 410-555-9385

Name: Rachel Phillips
Phone number: 310-555-6610
```


539740.3879454.qx3zqy7

LAB ACTIVITY | 28.5.1: LAB: Warm up: Contacts | 0 / 10

main.cpp Load default template...

```
1 #include <iostream>
```

```
using namespace std;

class ContactNode {
public:
   /* Declare member functions here */

private:
   /* Declare data members here */

};

/* Define member functions here */

int main() {
```


Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → **main.cpp** (Your program) → Outp

Program output displayed here

Coding trail of your work What is this?

History of your effort will appear here once you begin working on this zyLab.

28.6 LAB: Capitalizing vowels

Complete `CapVowels()`, which takes a string (char*) as a parameter and returns a new string (char*) containing the string parameter with the first occurrence of each of the five English vowels (a, e, i, o, and u) capitalized.

Hints: Begin `CapVowels()` by copying the string parameter to a newly allocated string. Use `strlen()` to obtain the length of a string.

Ex: If the input is:

```
management
```

the output is:


```
Original: management
Modified: mAnagEment
```


LAB ACTIVITY | 28.6.1: LAB: Capitalizing vowels | 0 / 10

main.cpp — Load default template...

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

// Return a newly allocated copy of original
// with the first occurrence of each vowel capitalized
char* CapVowels(char* original) {
   /*
      A new string must be allocated because
      a statically declared array (char result[50]) cannot be returned.
   */

   /* Type your code here. */

}
```

Develop mode | **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program Input (from above) → **main.cpp** (Your program) → Outp

Program output displayed here

Coding trail of your work What is this?

History of your effort will appear here once you begin working on this zyLab.

28.7 LAB: Swap two numbers

Complete the `Swap()` member function in the Number class to exchange the values of the num member variable of two Number objects, num1 and num2.

Ex: If num1 is 19 and num2 is 178, calling `num1.Swap(num2)` will swap the values so that num1 becomes 178 and num2 becomes 19.

539740.3879454.qx3zqy7

LAB ACTIVITY | 28.7.1: LAB: Swap two numbers 0 / 10

main.cpp Load default template...

```
#include <iostream>

using namespace std;

class Number {
   private:
      int num;

   public:
      void SetNumber(int n) {
         num = n;
      }

      int GetNumber() {
         return num;
```


Develop mode Submit mode

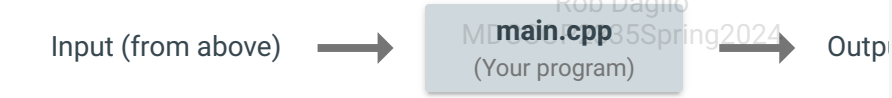
Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first

box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program



Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```

28.8 LAB: State ID renewal (Copy constructor)

Given class `StateID`, implement the copy constructor (in `StateID.cpp`).

`StateID` contains a pointer to an array of strings with length 5, which contains the state identification card's information. The information stored in the array should be in order of **Name, State, Identification Number, Date of Birth,** and **Expiration Date.**

In `main.cpp`, create a new `StateID` object as a backup of the old state ID (copy `id` to the new object). `main.cpp` then sets `id`'s state, identification number, and expiration date to "Pennsylvania", "87654321", and "1/25/2028" respectively.

Print both the renewed and old state ID card's information using each ID's own print member function.

The output of `main.cpp` is:


```
Old ID:
Name: John Doe
State: Nebraska
IDN: N01234567
DOB: 1/23/1980
EXP: 1/24/2023
```

```
New ID:
Name: John Doe
State: Pennsylvania
IDN: 87654321
DOB: 1/23/1980
EXP: 1/25/2028
```


539740.3879454.qx3zqy7

LAB ACTIVITY 28.8.1: LAB: State ID renewal (Copy constructor) 0 / 10

Downloadable files

`main.cpp` , `StateID.h` , and `StateID.cpp` **Download**

Current file: main.cpp Load default template...

```
1  #include "StateID.h"
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
7  int main() {
8     string info[] = {"John Doe", "Nebraska", "N01234567", "1/23/1980", "1/24/
9     StateID id(info);
10
11    // TODO: Create backup and call copy constructor.
12
13    id.SetState("Pennsylvania");
14    id.SetIdn("87654321");
15    id.SetExp("1/25/2028");
```

Develop mode **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program Input (from above) → main.cpp (Your program) → Outp

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```

28.9 LAB: State ID renewal (Copy assignment)

Given class `StateID`, overload the assignment operator (in `StateID.cpp`) so as to allow a StateID object to be assigned to another StateID object.

`StateID` contains a pointer to an array of strings with length 5, which contains the state identification card's information. The information stored in the array should be in order of **Name, State, Identification Number, Date of Birth,** and **Expiration Date.**

In `main.cpp`, create a new `StateID` object as a backup of the old state ID (copy `id` to the new object). `main.cpp` then sets `id`'s state, identification number, and expiration date to "Pennsylvania", "87654321", and "1/25/2028" respectively.

Print both the renewed and old state ID card's information using each ID's own print member function.

The output of `main.cpp` is:

```
Old ID:
Name: John Doe
State: Nebraska
IDN: N01234567
DOB: 1/23/1980
EXP: 1/24/2023

New ID:
Name: John Doe
State: Pennsylvania
IDN: 87654321
DOB: 1/23/1980
EXP: 1/25/2028
```

LAB ACTIVITY | 28.9.1: LAB: State ID renewal (Copy assignment) 0 / 10

Downloadable files

`main.cpp` , `StateID.h` , and `StateID.cpp` **Download**

Current file: main.cpp **Load default template...**

```
1  #include "StateID.h"
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
7  int main() {
8     string info[] = {"John Doe", "Nebraska", "N01234567", "1/23/1980", "1/24/
9     StateID id(info);
10    StateID oldId;
11
12    // TODO: Assign id to oldId.
13
14    id.SetState("Pennsylvania");
15    id.SetIdn("87654321");
```

Develop mode | **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above) → main.cpp (Your program) → Outp

Program output displayed here

Coding trail of your work What is this?

```
History of your effort will appear here once you begin working
on this zyLab.
```


©zyBooks 01/31/24 18:15 1939727
Rob Daglio
MDCCOP2335Spring2024