CSCI 2170 Test 2 **Name _______________________**

You are on the honor system and you are to do this exam with no help from any other person and the internet. When you sign your name below, you are indicating that you have adhered to these restrictions.

Email me (Cen.Li@mtsu.edu) in case you have questions.

1. For the **paper test (70 pts)**
   - include the following line at the top of your answer sheet, with your name filled in.

   I, ____, worked on all of the problems on this test completely on my own without any assistance from any other person. I did not use my book, my notes, and the internet.

   - Just type the answers, label each answer with the question number.
   - Save the file and export it as a PDF file before submitting to the D2L Dropbox labelled "Test 2".
   - The paper test is due April 9th (Thursday) at 4:00 pm. Make sure to submit the answer to D2L dropbox before due time.

2. For the **programming test (30 pts)**, you need to write, compile and run the program on the "ranger" system and submit the program using the "handin" command:

```
handin  test2 list.cpp
```

The programming test is due at April 9th (Thursday) at 6:00 pm. Make sure to handin the program before the due time.

CSCI 2170 Test 2 (Total 70 pts) Name_____________________

1. (6 pts, 2pts each) Multiple choice question:

a) Given the following structure and variables:

```
struct   Student {
  string     name;
  string     email;
};
Student       Peter, *p;
```

Which of the following statements makes the pointer p point to the variable ***Peter***?

(a) p = *Peter;
(b) &P = Peter;
(c) p = &Peter;
(d) *p = Peter;
(e) *p = &Peter;

b) Which of the following statements is NOT valid C++ code? :

(a) int ptr=&num1;
(b) int ptr=int *num1;
(c) float num1=&ptr2;
(d) all of these are valid
(e) all of these are invalid

c) For a linked list, in an insertion or deletion routine: how many pointers are you required to create for use during the traversal process?

(a) Two: one for the node under inspection and one for the previous node
(b) Two: one for the node under inspection and one for the next node
(c) One: for the node being inserted or deleted
(d) Three: one for the node under inspection, one for the next node, and one for the following node

2. (16 pts) Show the output of the following 4 program segments

(a) (3 pts) const int SIZE=8;

```
        int values[SIZE] = {10, 10, 14, 16, 6, 25, 5, 8};
        int index;
        index=0;
        res = values[index];
        for (int j=1; j<SIZE; j++)  {
                if (values[j] > res) {
                        res = values[j];
                        index = j;
                        cout << index << res << endl;
                }
        }
        cout << index << " " << values[index] << endl;
```

(b) (4 pts) const int SIZE=10;

```
int values[SIZE] = {0, 10, 14, 6, 6, 5, 5, 8};
int index=0;
int count=8;
bool flag = false;
int num = 6;
int result;
```

```
while (index < count && !flag) {
        if (values[index] == num)
        {
                flag = true;
                result = index;
        }
        else
                cout << “!”;

        index++;
}
cout << result << endl;
```

(c) (4 pts)

```
const int SIZE = 10;   // maximum number of items to store in array

void MyFunc(int arr[], int pos, int &size);
int main(){
        int array[SIZE];
        int position=2;
        int aSize=6;

        for (int i=0; i<aSize; i++)
            array[i] = i*10;

        MyFunc(array,  position, aSize);

        // display values after the insertion
        for (int i=0; i<aSize; i++)
            cout << array[i] << " ";

        return 0;
}
void MyFunc(int arr[], int pos, int &size) {
        for (int i=pos; i<size-1; i++)  {
             arr[i] = arr[i+1];
        }
        size--
}
```

(d) (5 pts)

```
        typedef   int * intPtr;
        int  main()   {
             intPtr   p, q;
             int    x, y;

             p = & x;
             *p = 10;
             q = & y;
             y=*p+10;
             cout << *p << "  " << *q << "  " << x  << " " << y << endl;

             q = p;
             *q = 20;
        cout << *p << "  " << *q << "  " << x  << " " << y << endl;

         q = new int;
         *q = x+y;
         x = x+*q;
         cout << *p << "  " << *q << "  "  << x  << " " << y << endl;

         return 0;
}
```

3. (10 points) A class grade book keeps records of 25 students' 15 lab grades in the following array structure:

```
const int NUM_STUDENTS = 25;
const int NUM_LABS=15;
int grades[NUM_STUDENTS][NUM_LABS];
```

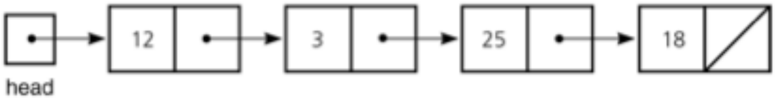
For the following problems you may assume that the lab grade for each of the 25 students for each of the 15 labs has been recorded in the "grades" array.

(a) (5 pts) Show the C++ statements to compute the average lab grade for <u>student number 10</u>

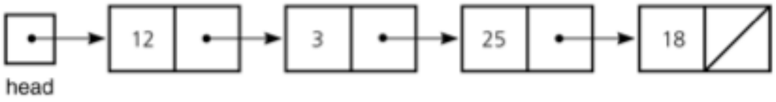
(b) (5 pts) Show the C++ statements to find which student scores the lowest for <u>lab 10</u>. If there are multiple students scoring the same lowest lab score, display all these students.

4. (6 pts) (a) Show the C++ statements needed to dynamically create an array of size 10 of type double. Initialize each array element to 2.0. (declare all the variables as needed)
(b) Show the C++ statement(s) to release the memory allocated dynamically in the previous question.

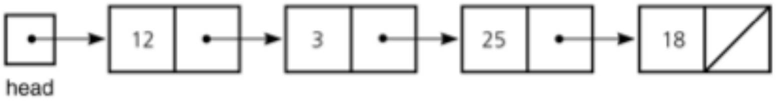
5. (12 pts, 4pts each) For each of the 4 questions a-d shown below, only the head of the **linked list** is given. You may declare additional pointers as needed. Show the code for that <u>specific linked list</u> only. Do not need to write the code for a general problem.



a. Show a C++ loop statement that computes the sum of all the values in the linked list.



b. Write the C++ statements to **remove the first node (node with data 12) from the list and free its memory**



c. Show the C++ statements needed to insert a new node with value 10 in between the two nodes having values 12 and 3 in the linked list shown above. Show the code for this particular case only, do not write the code for a general case.

d. Write the C++ loop statements needed to remove all the nodes in the list and free the memory of all the nodes back to the memory pool (i.e., the heap).

5. (10 pts) We wish to create a database of all the restaurants in Murfreesboro. For each restaurant, we need to store information about its name, phone number, food type, the average price for a meal.
   a. Show the declaration of a C++ struct type to represent a restaurant.
   b. Declare an array of the struct type defined in part a. This array can be used to store information of ALL the restaurants in Murfreesboro (max 1000 restaurants).
   c. Read restaurant information from a data file already opened using *myIn* of ifstream type, and store the information in the array declared part (b).
      The format of the restaurants in the data file are as the following. After all the records are read, your code should also display the total number of restaurants read.

      Blue Coast Burrito
      (615) 433-0922
      Mexican
      10.85

6. (10 pts) Show the C++ function **ListDelete** for **sorted linked list.** The function removes a node from the list that has data matching the data to be deleted. The head of the linked list and the data to be deleted are passed in the function as parameters. If the data to be deleted is not in the list, a message should be displayed. Show the complete function definition.