



Name_____

Userid (UD email address)_____

Please circle your section number (or lab time)

010 (Mon 09:05) 013 (Wed 09:05) 080 (Mon 7pm)

011 (Mon 10:10) 014 (Wed 10:10)

012 (Mon 11:15) 015 (Wed 11:15)

Answer the multiple choice questions on a “Scantron Form”

Bubble in ONLY your Unix userid and your answers

DO NOT bubble in your id number or section

If you bubble in your SSN, the computer will **reject your form!!!**

Answer the remaining questions directly on the exam paper.

General Instructions

- The programming questions start with number 11. You may want to tackle them first, since they may take more time.
- DO NOT WRITE YOUR NAME ON ANY PAGE EXCEPT THIS ONE!
- You have two hours. **Pace yourself**, and pay attention to the point values.
- Read *all* the directions *carefully* on each problem.
- Good luck.

Questions 1 through 2 should be answered together.

1. (2 pts) Suppose that you want to pass an array of integer values to a function that will sort those values in descending order. The function is an ordinary function, not a member function of a class. Which of the following is a reasonable function prototype for such a function?
 - (a) `void sortAscending(int a[], int count) const;`
 - (b) `void sortAscending(const int * const a, int count);`
 - (c) `void sortAscending(int *a, const int count);`
 - (d) `void sort(int a[], int count[]);`
2. (2 pts) (Continuation of question 1.) Suppose the following code appeared in a main program.

```
int myArray[] = {3, 16, 89, 12};
```

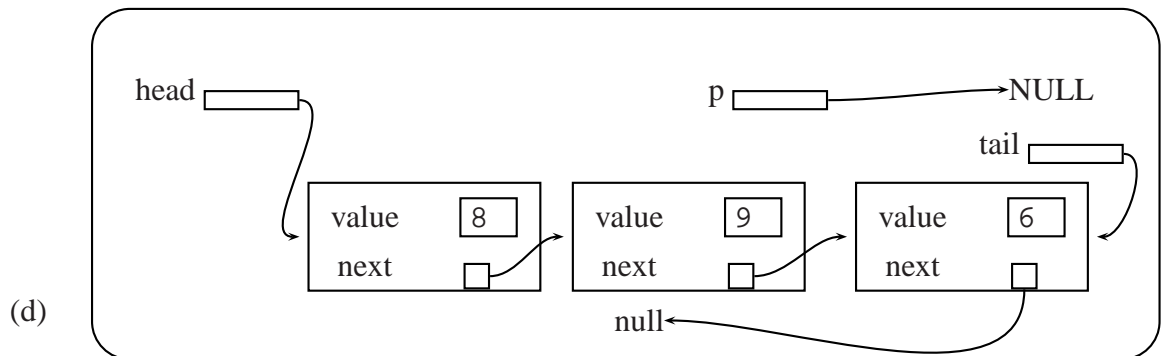
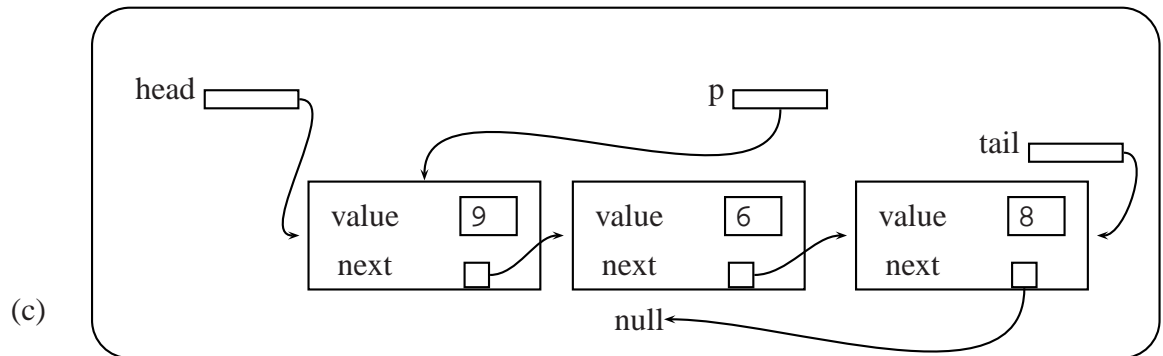
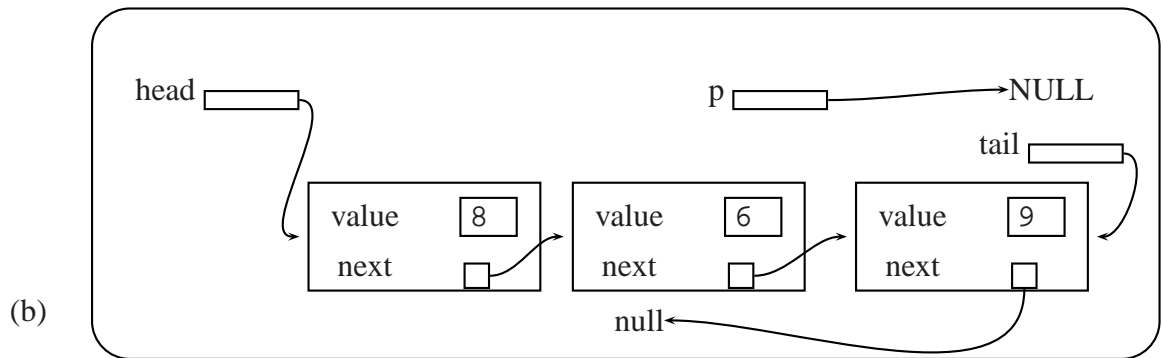
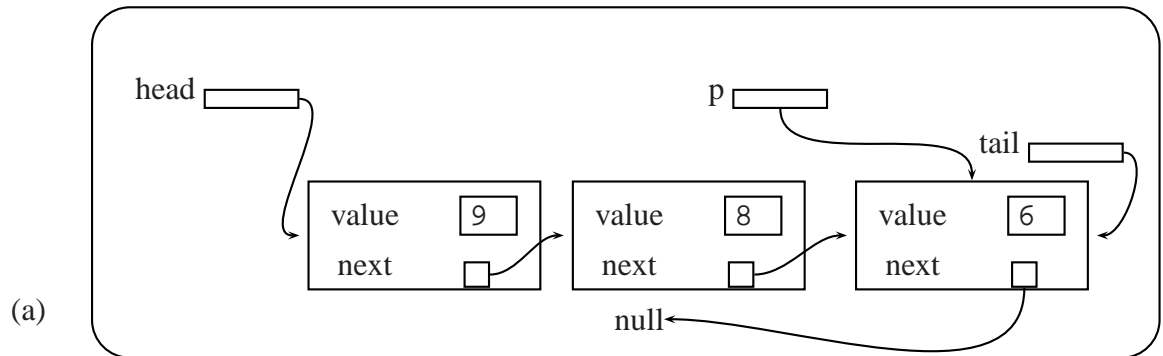
Which of the following would be a reasonable function call to the function referred to in problem 1?

- (a) `sortAscending(myArray, 4);`
- (b) `sortAscending(myArray[], 4);`
- (c) `sortAscending(myArray[4], 4);`
- (d) `sortAscending(&myArray, 4);`
- (e) `sortAscending(*myArray, 4);`

Questions 3 through 4 deal with the code for `p04.cc` on page 15.

3. (4 pts) The output of `p04.cc` will be
 - (a) 9 6 8
 - (b) 9 8 6
 - (c) 8 9 6
 - (d) 8 6 9
 - (e) 6 8 9

4. (4 pts) Which of the following is an accurate picture of what is in memory at the time the `return 0;` statement on line 39 of program `p04.c` (page 15) is reached? Look carefully at where `head`, `tail` and `p` are pointing.



5. (2 pts) Which of the following is a correct function prototype for a copy constructor for class `Foo_C`?
- (a) `Foo_C(const Foo_C orig);`
 - (b) `Foo_C(const Foo_C &orig);`
 - (c) `void Foo_C(const Foo_C &orig);`
 - (d) `void Foo_C(const Foo_C &orig) const;`
 - (e) `Foo_C copy(const Foo_C &orig);`
6. (3 pts) Which of the following should be the last line executed inside a correctly written overloaded assignment operator?
- (a) `return this;`
 - (b) `return *this;`
 - (c) `return right;`
 - (d) `return left;`
 - (e) `return 0;`
7. (3 pts) Which of the following should be the last line executed inside a correctly written overloaded stream insertion operator, assuming that this operator is NOT written as a member function?
- (a) `return this;`
 - (b) `return *this;`
 - (c) `return right;`
 - (d) `return left;`
 - (e) `return 0;`

8. (10 pts) A correctly written recursive function typically has two parts.

Below is an incomplete definition for a recursive function to compute two the power of n , given that n is a non-negative integer. Finish the definition, clearly identifying (with comments) the two “parts” that a recursive function must have in order to be correct.

```
1 // twoToTheN.cc 181 Final Exam
2 // Calculate 2 to the nth power
3
4 #include <iostream>
5 using namespace std;
6
7 int twoToTheN(int n)
8 {
9     // handle error case
10    if (n<0)
11    {
12        cerr << "Fatal error: passed negative value to twoToTheN " << endl;
13        exit(1);
14    }
15
16    // Insert your code here for the two parts that a
17    // correct recursive function typically needs:
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37 }
38
39 int main(void)
40 {
41     int n;
42     cout << "Enter n: ";
43     cin >> n;
44     cout << twoToTheN(n) << endl;
45     return 0;
46 }
```

9. In the context of talking about pointers and the heap, we sometimes speak of “garbage” and “memory leaks”. What is garbage (5 pts) and what is a memory leak (5 pts)?

10. (10 pts) Below is a shell of a main program. Add a few lines of C++ code that create at least one piece of “garbage” (i.e. a memory leak.) Mark the *specific* line of code that causes the memory leak with a comment `// memory leak here`.

Note: If you understand the concept, is possible to answer correctly with just two lines of C++ code.

```
1 // memLeak.cc An example memory leak, 181 Final Exam
2
3 int main(void)
4 {
5
6
7
8
9
10
11
12
13
14
15
16
17     return 0;
18 }
```

11. (30 pts) Write the .h files only (the class declaration) for two classes `CourseGrade_C` and `GradeReport_C`.

The full specification of these classes appears on page 16. Refer to that page to be sure that your class definitions are complete!

There is extra space for your answer on the next several pages.

Extra space for your answer to question 11

Extra space for your answer to question 11

Extra space for your answer to question 11

12. (20 pts) Now choose ANY TWO of the following FIVE member functions of `GradeReport_C`, and write the function definitions as they would appear in the file `gradeReport.cc`.

- The member function that adds a new course grade to a grade report. Be sure you allocate a *new* course grade object that is a copy of the one passed in by const reference.
- The member function that computes total credit hours. (This should traverse the linked list—you are not permitted to keep this value as a separate data member.)
- The member function that returns true or false depending on whether a given course is in the schedule.
- The copy constructor for the `GradeReport_C` class.
- The destructor for the `GradeReport_C` class.

Choose ONLY TWO. (You may NOT answer three or four and have me “pick the best ones”. If you answer more than two, I will arbitrarily choose the two that are easiest for me to grade, and these won’t necessarily be your best two!)

Extra space for your answer to question 12

Extra space for your answer to question 12

Extra space for your answer to question 12

End of Exam, seed 3456, version A. Total Points: 100

Code Excerpts and Specifications referred to in Exam Questions

p04.cc

```
1 // p03.cc P. Conrad, Fall 2005, CISC181 Final Exam
2
3 #include <iostream>
4 using namespace std;
5
6 struct Node_S
7 {
8     int data;
9     Node_S *next;
10 };
11
12 int main(int argc, char *argv[])
13 {
14
15     Node_S *head = NULL;
16     Node_S *tail = NULL;
17     Node_S *p = NULL;
18
19     head = new Node_S;
20     head->data = 9;
21     head->next = NULL;
22     tail = head;
23
24     p = new Node_S;
25     p->data = 6;
26     p->next = NULL;
27     tail -> next = p;
28     tail = p;
29
30     p = new Node_S;
31     p->data = 8;
32     p->next = head;
33     head = p;
34
35     for (p=head; p; p=p->next)
36         cout << p->data << " ";
37     cout << endl;
38
39     return 0;
40 }
```

Specifications for Problem 11

A `CourseGrade_C` object represents a single grade in a single course, and should contain the following private data members:

- space for a course number with four letters and three numbers (e.g. MATH205, CISC181, CPEG202, ENGL110) stored as a C-string. This may be a fixed size char array, or a char * pointing to space allocated on the heap.
- space that can hold a grade of A, B, C, D, or F, with plus or minus (e.g. A-, B+) stored as a C-string. This may be a fixed size char array, or a char * pointing to space allocated on the heap.
- a number of credit hours, stored as an integer.

A `GradeReport_C` object represents a list of `CourseGrade_C` objects. The `GradeReport_C` class definition should contain a private struct definition for a `Node_S` that has two fields: a next pointer, and a pointer to a `CourseGrade_C` object allocated from the heap. The `GradeReport_C` class should have only two private data members: pointers to the head and tail of a linked list of nodes. It should also have public member functions to:

- add a course grade to a grade report (pass in a const reference to an existing course grade object.)
- report the total number of credits (as an integer)
- report the GPA (as a number such as 3.76 or 2.54).
- report whether or not a certain course is on the student's schedule. Pass in the course number as `const char * const courseNum`, and return a value of type `bool`.

Also include function prototypes for:

- a constructor that takes no arguments and initializes an empty grade report. Include an inline definition that initializes the head and tail pointers to `NULL`. (You may assume that `NULL` is defined.)
- a copy constructor
- a destructor
- an overloaded assignment operator.

IMPORTANT: For full credit:

- be sure to use `const` on any and all member functions for which it is appropriate.
- Your answer should include the pre-processor directives that prevent multiple inclusion of the .h file.

End of code excerpts for 