

CSC 112 MIDTERM EXAM

You will have until 10:20 a.m. to complete this exam. All your work should be written in the exam booklet. Start with the questions that you know how to do, and try not to spend too long on any one question. Partial credit will be granted where appropriate. Good luck!

1. (10 points) Predict the output generated by each of the following two programs.

a. `#include <iostream.h>`

```
void main() {
    int i;
    int arr[6] = {4, 2, 6, 3, 1, 8};
    int sum = 0;

    for (i = 0; i < 6; i++) {
        sum += arr[i];
        cout << sum << endl;
    }
}
```

b. `#include <iostream.h>`

```
int myfun(int a, int &b, int *c) {
    a++;
    b++;
    (*c)++;
    cout << a << endl;
    cout << b << endl;
    cout << (*c) << endl;
    return a+b+(*c);
}

void main() {
    int x = 1;
    int y = 2;
    int z = 3;
    int w;

    w = myfun(x, y, &z);
    cout << w << endl;
    cout << x << endl;
    cout << y << endl;
    cout << z << endl;
}
```

2. (10 points) Fill in the blanks in this program as indicated.

```
void main() {
    int **arr2d;
    int nrow = 3;
    int ncol = 5;
    int i, j;

    // allocate nrow x ncol 2d array
    // write code here...

    // zero out new array
    // your code should be compatible with this piece
    for (i = 0; i < nrow; i++) {
        for (j = 0; j < ncol; j++) {
            arr2d[i][j] = 0;
        }
    }

    // free array storage
    // write code here...
}
```

3. You are designing a new class, `Pet`, that will keep track of data about pets for a store. The class should be able to record a pet's name, age, price, and sex. It should have methods `Print`, `Read`, and `incrementAge`. In your answers to this question, you should show that you have absorbed the principles of good class design taught in class.

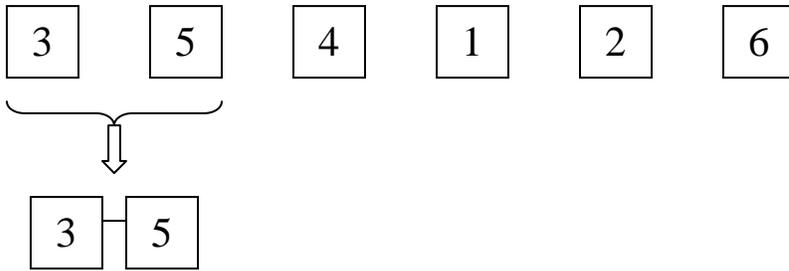
a. (10 points) Write a class definition for `Pet`. Make sure that you use appropriate types for each of the properties, and use the `public` and `private` keywords appropriately. You do not need to include definitions for any of the methods yet – just the declaration within the class definition is sufficient. Your class should include all applicable accessors and manipulators, plus at least one constructor and destructor.

b. (5 points) Write definitions for the manipulator and accessor functions that work with the `price` property.

c. (5 points) Write the definitions of two constructors for `Pet`: one that takes no arguments and fills in the properties with default values, and a second that takes arguments for each of the properties and uses them to initialize the object.

d. (5 points) Write the definition of the `incrementAge` method, which increases the recorded age of the pet by one.

4. (10 points) Show the steps in the merge sort algorithm, starting with the set of lists below. (You don't need to draw all the pointers for this question – just show clearly which lists merge, and the order of the merged lists. The first step is done for you.)



5. (15 points) Find the bugs in the following implementation of insertion sort, which is intended to sort the array in ascending order. Please mark clearly the changes you are making. (Note: assume that the comments correctly show the intended behavior.)

```

void insertion_sort(int *arr, int arr_len) {
    int i, j;
    int tmp;

    // loop through entire list
    for (i = 0; i <= arr_len; i++) {
        // insert item in proper sorted location
        for (j = i; j > 0; j++) {
            // see if we need to move current item
            // to a lower position
            if (arr[i] > arr[j-1]) {
                // swap items
                tmp = arr[j];
                arr[j] = arr[j-1];
                arr[j-1] = tmp;
            }
        }
    }
}

```

6. (10 points) Draw the memory structures that would be created by the following program. Make sure that you show the data values in the list nodes, as well as all the pointers, including head, tail, and item.

```
#include "IntLinkItem.h"

void main() {
    IntLinkItem *head, *tail, *item;

    head = new IntLinkItem;
    tail = new IntLinkItem;
    head->insertBefore(tail);

    item = new IntLinkItem(1);
    item->insertBetween(head,tail);
    item = new IntLinkItem(2);
    item->insertBefore(tail);
    item = new IntLinkItem(3);
    item->insertAfter(head->getNext());
    item = item->getPrev();
}
```

7. (10 points) Write code to find the minimum item in a linked list. (You should return a pointer to the node containing the item.) The function header is given; you need to fill in the body.

```
IntLinkItem* findMinNode(IntList list) {
    // insert code here
}
```

8. (10 points) Write code to take a list and cycle it. (A single cycle takes the first node in the list and moves it to the end. Your function should do this ncycle times.)

```
void cycleList(IntList list, int ncycle) {
    // insert code here
}
```