

Name: _____

SSN: _____

CS 12 - Final Exam March 19, 1997

Be sure to read each problem carefully and follow the directions. Points may be marked off if you do not follow the directions. For example, if the problem asks you to write a function, do not write an entire program. Please feel free to ask if you have any questions.

Problem 1	10	
Problem 2	10	
Problem 3	10	
Problem 4	10	
Problem 5	12	
Problem 6	28	
Problem 7	10	
Problem 8	5	
Problem 9	10	
Problem 10	10	
Problem 11	6	
Problem 12	10	
Problem 13	9	
TOTAL	140	

Name: _____

SSN: _____

1. (10 pts) Write a declaration for a two-dimensional array of integers initializing the values of each element. Write code to search the array and print out the largest value in the array and the row and column in which the value is stored.

2. (10 pts) Given the following definitions write code to nicely display the array using *printf*.

```
struct mystruct
{
    char *str;
    int x;
    float y;
};

void main()
{
    mystruct A[10];

    // initialization code
    ...

    // display code

}
```

3. (10 pts) Overload the << operator to display the structure given in the previous problem. This means you will **not** use printf this time.

4. (10 pts) Rewrite the following code to use pointers rather than the subscript operator. Assume *A* has already been declared as an array of 10 integers.

```
A[2] = 9;
A[6] = 4;
for(int i=0; i<10; i++)
    cout << A[i] << endl;
```

5. (12 pts) Write a program that reads in integers from a file and prints them out to the screen, one per line. Read in integers until either 100 integers have been read or you have reached the end of the input file. You should not assume any particular format for the integers to be in the input file (they may be on a line by themselves, blank lines in between, many on one line...). Prompt the user for the name of the input file.

6.
 - (a) (10 pts) Define a linked list class. Each node should be able to hold information for a grocery store product. It should contain the name of the product, the price, and the number of products in stock.
 - (b) (6 pts) Show any necessary constructors and destructors.
 - (c) (6 pts) Write a member function that inserts a new product node at the head of the list.
 - (d) (6 pts) Write a member function that accepts a string for the product name as a parameter. The functions should then search the list for that product and return a pointer to the node of the product if found, otherwise it should return NULL.

7. (10 pts) State the differences between an array, a dynamically allocated array, and a list. What effect do they have on the number of elements allowed? At what time is the size determined?

8. (5 pts) Briefly state what the following function does: (note: restating the function is not an adequate explanation)

```
double F(double x, int n)
{
    if (n == 0)
        return 0;
    else
        return x + F(x, n-1);
}
```

9. (10 pts) Write a recursive function to compute and return the factorial of a non-negative integer.

10. (10 pts) Write a recursive function that computes the gcd of two numbers. The gcd is defined as:

n if m is divisible by n
gcd(n, the remainder of m divided by n) otherwise

11. (6 pts) Fill in the following table:

Inheritance Type	Base Class Member Type	Derived Class Member Type
public	private protected public	
private	private protected public	

12. (10 pts) Write a template function that accepts two parameters and swaps their values. The function should not return anything.

13. Given the following code, state whether the following code segments are legal or illegal. If legal, show the output.

```
class C1
{
    protected:
        int num1;
        int num2;
    public:
        void setinfo()
        {
            num1 = 1;
            num2 = 2;
        };
        virtual void show()
        {
            cout << num1 << "\t" << num2 << endl;
        };
};
class C2 : public C1
{
    private:
        int count;
    public:
        void setinfo()
        {
            num1 = 3;
            num2 = 4;
            count = 5;
        };
        virtual void show()
        {
            for(int i=0; i<count; i++)
                cout << num1 << "\t" << num2 << endl;
        };
};
class C3 : public C1
{
    public:
        void setinfo()
        {
            num1 = 5;
            num2 = 6;
        };
        virtual void show(int num_times)
        {
            for(int i=0; i<num_times; i++)
                cout << num1 << "\t" << num2 << endl;
        };
};
```

Assume the following declarations have been made. Each set of code is separate. Do not assume that another set of code has come before it.

```
C1 A;  
C2 B;  
C3 C;  
C1 *Ptr;
```

(a) (3 pts)

```
A.setinfo();  
B.setinfo();  
C.setinfo();  
  
A.show();  
B.show();  
C.show();
```

(b) (3 pts)

```
A.setinfo();  
B.setinfo();  
C.setinfo();  
  
Ptr = &A;  
Ptr->show();  
Ptr = &B;  
Ptr->show();  
Ptr = &C;  
Ptr->show();
```

(c) (3 pts)

```
Ptr = &A;  
Ptr->setinfo();  
Ptr = &B;  
Ptr->setinfo();  
Ptr = &C;  
Ptr->setinfo();  
  
Ptr = &A;  
Ptr->show();  
Ptr = &B;  
Ptr->show();  
Ptr = &C;  
Ptr->show();
```