CSE 20211 – Final – Fall 2011 NAME:
NETID:

```
int a[10];      int *p, *q;     int **d;
```

State the **type** of the following C expressions:

```
a
```

```
p[3] + 1
```

```
q + 3
```

```
**(d + 1)
```

```
p < q ? &a[3] : *d
```

State the **value** of the following C expressions:

```
251 ^ 165
```

```
( 1 << 6 ) - 1
```

```
39 | 203
```

```
127 & 85
```

```
~( 128 >> 3 )
```

Find All The Bugs. (10 points)

The following C program below is meant to read in 100 doubles from a file, sort them, and print them out. Find all the problems with this code, including compile-time errors, run-time errors, logic errors, and errors of omission. For each error, circle the problem in the code, and briefly explain the problem on the right side of the page.

```
#include <stdio.h>
#include <stdlib.h>


void main( int )
{
	int count=100;
	double *arr = malloc( sizeof(int)*count );

	FILE *file = fopen("input.data","w");

	for(i=1;i<=count;i++) {
		fscanf(file,"%lf",arr[i]);
	}

	fclose(file);

	bubble_sort(&arr,count+1);

	free(arr);

	for(i=0;i<count;i++) {
		printf("%d\n",arr[i]);
	}

	return 0;
}


void bubble_sort( double *arr, int count )
{
	int i;

	for(i=0;i<count;i++) {
		if(arr[i]>arr[i+1]) {
			double temp = arr[i+1];
			arr[i+1] = arr[i];
			arr[i] = temp;
		}
	}
}
```

Linked Lists (20 points)

Suppose that you would like a program to keep track of a list of friends. To allow for an arbitrary number of friends, you will use a linked list with the following definition:

```
struct list_node {
      char name[100];
      struct list_node *next;
};
struct list_node *head = 0;
```

Write a C **function** that prints out all of the names in the list. (5 points)

Write a C **function** that searches for a name in the list and returns true if present. (5 points)

Write a C **function** that adds a new name to the head of the list. (5 points)

Write a C **function** that removes a name from the end of the list. (5 points)

Strings (10 points)

Write a C **program** that obtains two strings from the user, and determines whether one line is an anagram of the other, ignoring case and non-letters. Two phrases are considered anagrams if they both contain the same set of letters. For example “Tim Marvolo Riddle” is an anagram of “I am Lord Voldemort”. Continue on the back of this page if needed. (Hint: How many times does each letter occur?)

Just a Little C++ (10 points)

In the box to the right, write out what this C++ program displays.

```
#include <iostream>

using namespace std;

class Mystery {
private:
      int a, b;
public:
      Mystery( int x, int y );
      ~Mystery();
      flob();
};

Mystery::Mystery( int x, int y ) {
      a = x+y;
      b = x-y;
      cout << "hello " << a << endl;
}

Mystery::~Mystery() {
      cout << "goodbye " << b << endl;
}

void Mystery::flob()
{
      cout << "flob " << (a+b) << endl;
}



Mystery g(10,20);

int main() {
      Mystery a(60,20);
      a.flob();
      Mystery b(60,50);
      g.flob();
      b.flob();
      return 0;
}
```