Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.
    a. (2 pts) Convert 1111 0010 from base 2 to hexadecimal
    b. (2 pts) Convert 1100 0001 0011 0101 from binary to hexadecimal
    c. (2 pts) Convert 101 000 011 from base 2 to base 8
    d. (2 pts) Convert 64 from base 8 to binary
    e. (2 pts) Convert 178 from decimal to binary
    f. (2 pts) Convert 0100 0100 0011 from base 2 to base 16
    g. (2 pts) Convert 192 from decimal to base 2
    h. (2 pts) Convert 100 010 101 from base 2 to octal
    i. (2 pts) Convert 1111 1011 from base 2 to base 10
    j. (2 pts) Convert 1000 0111 0100 1100 from base 2 to base 16

2. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon grape apple
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[2][3]?

c. (2 pts) What is the value of argv[1][2]?

d. (2 pts) What is the value of argv[0][5]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node w;
  int x;
  double y;
  char z;
  Node *a;
  int *b;
  double *c;
  char *d;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2]

b. (2 pts) d

c. (2 pts) argc

d. (2 pts) a->data

e. (2 pts) &y

f. (2 pts) w

g. (2 pts) a->next

h. (2 pts) argv[0]

i. (2 pts) &d

j. (2 pts) *d

k. (2 pts) a->next->next

4. (20 pts) Given the following struct definition:

```
struct Precip {
  int day;
  double inches;
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5









  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100