

Exam #101 Page: 1 Name: _____

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 f2

b. (2 pts) Convert 1100 0001 0011 0101 from binary to hexadecimal c135

c. (2 pts) Convert 101 000 011 from base 2 to base 8 503

d. (2 pts) Convert 64 from base 8 to binary 110 100

e. (2 pts) Convert 178 from decimal to binary 1011 0010

f. (2 pts) Convert 0100 0100 0011 from base 2 to base 16 443

g. (2 pts) Convert 192 from decimal to base 2 1100 0000

h. (2 pts) Convert 100 010 101 from base 2 to octal 425

i. (2 pts) Convert 1111 1011 from base 2 to base 10 251

j. (2 pts) Convert 1000 0111 0100 1100 from base 2 to base 16 874c

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? 4

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

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

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

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]` **char**

b. (2 pts) `d` **char ***

c. (2 pts) `argc` **int**

d. (2 pts) `a->data` **int**

e. (2 pts) `&y` **double ***

f. (2 pts) `w` **Node**

g. (2 pts) `a->next` **Node ***

h. (2 pts) `argv[0]` **char ***

i. (2 pts) `&d` **char ****

j. (2 pts) `*d` **char**

k. (2 pts) `a->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #101 Page: 8 Name: _____

End of Exam

total points=100

Exam #102 Page: 1 Name: _____

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 3254 from hexadecimal to base 2 **0011 0010 0101 0100**

b. (2 pts) Convert 92ba from base 16 to base 2 **1001 0010 1011 1010**

c. (2 pts) Convert d7f7 from base 16 to base 2 **1101 0111 1111 0111**

d. (2 pts) Convert 44 from decimal to binary **0010 1100**

e. (2 pts) Convert 8c1 from hexadecimal to binary **1000 1100 0001**

f. (2 pts) Convert 0011 0110 from binary to decimal **54**

g. (2 pts) Convert 60 from base 8 to binary **110 000**

h. (2 pts) Convert 1111 0011 from binary to decimal **243**

i. (2 pts) Convert 1001 1101 from binary to decimal **157**

j. (2 pts) Convert 54 from base 8 to base 2 **101 100**

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 kiwi apple
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[1][2]`? w

c. (2 pts) What is the value of `argv[2][4]`? e

d. (2 pts) What is the value of `argv[0][6]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node x;  
    double y;  
    int z;  
    char a;  
    Node *b;  
    double *c;  
    int *d;  
    char *e;  
  
    return 0;  
}
```

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

- a. (2 pts) b->next **Node ***
- b. (2 pts) &b **Node ****
- c. (2 pts) b->data **int**
- d. (2 pts) &x **Node ***
- e. (2 pts) argv[0] **char ***
- f. (2 pts) *e **char**
- g. (2 pts) argc **int**
- h. (2 pts) b->next->next **Node ***
- i. (2 pts) argv[1][2] **char**
- j. (2 pts) d **int ***
- k. (2 pts) z **int**

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 parameters 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 parameters 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;
}
```

Exam #102 Page: 8 Name: _____

End of Exam

total points=100

Exam #103 Page: 1 Name: _____

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 35 from base 8 to binary **011 101**

b. (2 pts) Convert 010 011 001 from binary to octal **231**

c. (2 pts) Convert c5d8 from hexadecimal to base 2 **1100 0101 1101 1000**

d. (2 pts) Convert 101 110 001 from binary to octal **561**

e. (2 pts) Convert 1110 1100 0000 1101 from binary to hexadecimal **ec0d**

f. (2 pts) Convert 110 100 001 from binary to octal **641**

g. (2 pts) Convert 33 from octal to base 2 **011 011**

h. (2 pts) Convert 0111 1011 from binary to decimal **123**

i. (2 pts) Convert 0110 0111 from base 2 to base 10 **103**

j. (2 pts) Convert 23 from octal to base 2 **010 011**

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 banana kiwi apple fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][3]`? u

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

d. (2 pts) What is the value of `argv[2][2]`? w

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double c;  
    int d;  
    Node e;  
    char f;  
    double *g;  
    int *h;  
    Node *p;  
    char *q;  
  
    return 0;  
}
```

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

- a. (2 pts) `p->next` **Node ***
- b. (2 pts) `p->data` **int**
- c. (2 pts) `c` **double**
- d. (2 pts) `&f` **char ***
- e. (2 pts) `*g` **double**
- f. (2 pts) `argv[1][2]` **char**
- g. (2 pts) `p->next->next` **Node ***
- h. (2 pts) `argc` **int**
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `h` **int ***
- k. (2 pts) `&q` **char ****

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 parameters 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 parameters 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;
}
```

Exam #103 Page: 8 Name: _____

End of Exam

total points=100

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 63 from octal to binary **110 011**

b. (2 pts) Convert 239 from decimal to base 2 **1110 1111**

c. (2 pts) Convert 0110 1011 0011 1010 from base 2 to hexadecimal **6b3a**

d. (2 pts) Convert 0111 0100 0001 0100 from base 2 to base 16 **7414**

e. (2 pts) Convert 0101 1100 from binary to decimal **92**

f. (2 pts) Convert 1101 0001 from binary to decimal **209**

g. (2 pts) Convert 190 from base 10 to binary **1011 1110**

h. (2 pts) Convert 0010 0010 0000 0010 from binary to hexadecimal **2202**

i. (2 pts) Convert 26 from octal to binary **010 110**

j. (2 pts) Convert 0101 1011 0000 1100 from binary to hexadecimal **5b0c**

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 apple cherry mango
```

a. (2 pts) What is the value of `argc` in this case? 4

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

c. (2 pts) What is the value of `argv[0][1]`? /

d. (2 pts) What is the value of `argv[1][3]`? l

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int t;  
    Node w;  
    double x;  
    char y;  
    int *z;  
    Node *a;  
    double *b;  
    char *c;  
  
    return 0;  
}
```

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

a. (2 pts) a->next->next **Node ***

b. (2 pts) argv[0] **char ***

c. (2 pts) a->next **Node ***

d. (2 pts) &a **Node ****

e. (2 pts) y **char**

f. (2 pts) &y **char ***

g. (2 pts) argc **int**

h. (2 pts) *z **int**

i. (2 pts) c **char ***

j. (2 pts) argv[1][2] **char**

k. (2 pts) a->data **int**

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 parameters 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 parameters 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;
}
```

Exam #104 Page: 8 Name: _____

End of Exam

total points=100

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 20 from base 10 to binary **0001 0100**

b. (2 pts) Convert 52 from octal to base 2 **101 010**

c. (2 pts) Convert 89 from decimal to binary **0101 1001**

d. (2 pts) Convert 1111 1111 1111 1110 from base 2 to hexadecimal **ffffe**

e. (2 pts) Convert 0011 1111 1101 0001 from binary to hexadecimal **3fd1**

f. (2 pts) Convert 6bb6 from base 16 to base 2 **0110 1011 1011 0110**

g. (2 pts) Convert 104 from decimal to binary **0110 1000**

h. (2 pts) Convert 1010 1001 from binary to decimal **169**

i. (2 pts) Convert 001 001 000 from base 2 to base 8 **110**

j. (2 pts) Convert 246 from decimal to binary **1111 0110**

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 apple kiwi lemon
```

a. (2 pts) What is the value of `argc` in this case? **4**

b. (2 pts) What is the value of `argv[0][3]`? **u**

c. (2 pts) What is the value of `argv[2][3]`? **i**

d. (2 pts) What is the value of `argv[1][4]`? **e**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node q;  
    int r;  
    double s;  
    char t;  
    Node *w;  
    int *x;  
    double *y;  
    char *z;  
  
    return 0;  
}
```

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

- a. (2 pts) `*x` **int**
- b. (2 pts) `argv[1][2]` **char**
- c. (2 pts) `w->data` **int**
- d. (2 pts) `w->next` **Node ***
- e. (2 pts) `argc` **int**
- f. (2 pts) `&w` **Node ****
- g. (2 pts) `&s` **double ***
- h. (2 pts) `w->next->next` **Node ***
- i. (2 pts) `s` **double**
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `y` **double ***

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 parameters 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 parameters 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;
}
```

Exam #105 Page: 8 Name: _____

End of Exam

total points=100

Exam #106 Page: 1 Name: _____

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 010 001 011 from binary to base 8 **213**

b. (2 pts) Convert 36 from base 8 to base 2 **011 110**

c. (2 pts) Convert 100 011 110 from binary to base 8 **436**

d. (2 pts) Convert 27 from base 8 to base 2 **010 111**

e. (2 pts) Convert 95ef from hexadecimal to binary **1001 0101 1110 1111**

f. (2 pts) Convert 9e20 from base 16 to base 2 **1001 1110 0010 0000**

g. (2 pts) Convert 106 from base 10 to base 2 **0110 1010**

h. (2 pts) Convert 0001 0010 from base 2 to decimal **18**

i. (2 pts) Convert 110 001 011 from binary to octal **613**

j. (2 pts) Convert 0010 0011 0011 0111 from binary to base 16 **2337**

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 kiwi lemon
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][1]`? **i**

c. (2 pts) What is the value of `argv[2][2]`? **m**

d. (2 pts) What is the value of `argv[0][3]`? **u**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node r;  
    double s;  
    int t;  
    char w;  
    Node *x;  
    double *y;  
    int *z;  
    char *a;  
  
    return 0;  
}
```

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

- a. (2 pts) `w` **char**
- b. (2 pts) `x->data` **int**
- c. (2 pts) `argv[0]` **char ***
- d. (2 pts) `argv[1][2]` **char**
- e. (2 pts) `x->next` **Node ***
- f. (2 pts) `y` **double ***
- g. (2 pts) `argc` **int**
- h. (2 pts) `&y` **double ****
- i. (2 pts) `x->next->next` **Node ***
- j. (2 pts) `*y` **double**
- k. (2 pts) `&s` **double ***

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 parameters 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 parameters 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;
}
```

Exam #106 Page: 8 Name: _____

End of Exam

total points=100

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 8ab0 from hexadecimal to base 2 **1000 1010 1011 0000**

b. (2 pts) Convert 0011 0101 from base 2 to decimal **53**

c. (2 pts) Convert 0111 1101 0101 1001 from binary to hexadecimal **7d59**

d. (2 pts) Convert e82a from base 16 to binary **1110 1000 0010 1010**

e. (2 pts) Convert 36 from octal to base 2 **011 110**

f. (2 pts) Convert 55 from decimal to binary **0011 0111**

g. (2 pts) Convert 20 from decimal to binary **0001 0100**

h. (2 pts) Convert 9a51 from hexadecimal to binary **1001 1010 0101 0001**

i. (2 pts) Convert 1001 0000 from base 2 to base 10 **144**

j. (2 pts) Convert 1011 1111 0010 0001 from base 2 to hexadecimal **bf21**

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 apple date cherry kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][0]`? .

c. (2 pts) What is the value of `argv[1][0]`? a

d. (2 pts) What is the value of `argv[2][3]`? e

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int h;  
    double p;  
    Node q;  
    char r;  
    int *s;  
    double *t;  
    Node *w;  
    char *x;  
  
    return 0;  
}
```

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

a. (2 pts) `*w` **Node**

b. (2 pts) `argv[0]` **char ***

c. (2 pts) `w->data` **int**

d. (2 pts) `&x` **char ****

e. (2 pts) `p` **double**

f. (2 pts) `s` **int ***

g. (2 pts) `w->next` **Node ***

h. (2 pts) `argv[1][2]` **char**

i. (2 pts) `&h` **int ***

j. (2 pts) `w->next->next` **Node ***

k. (2 pts) `argc` **int**

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 parameters 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 parameters 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;
}
```

Exam #107 Page: 8 Name: _____

End of Exam

total points=100

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 e30d from hexadecimal to base 2 **1110 0011 0000 1101**

b. (2 pts) Convert 110 110 000 from base 2 to octal **660**

c. (2 pts) Convert 10 from base 8 to base 2 **001 000**

d. (2 pts) Convert 1010 0011 from base 2 to decimal **163**

e. (2 pts) Convert e9b3 from base 16 to binary **1110 1001 1011 0011**

f. (2 pts) Convert 001 110 010 from binary to octal **162**

g. (2 pts) Convert 0110 0110 0000 1001 from base 2 to base 16 **6609**

h. (2 pts) Convert 010 000 001 from base 2 to octal **201**

i. (2 pts) Convert 1000 0010 from base 2 to decimal **130**

j. (2 pts) Convert 1100 1010 from base 2 to base 10 **202**

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 banana lemon lime
```

a. (2 pts) What is the value of `argc` in this case? **4**

b. (2 pts) What is the value of `argv[0][3]`? **u**

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

d. (2 pts) What is the value of `argv[1][5]`? **a**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int a;  
    Node b;  
    double c;  
    char d;  
    int *e;  
    Node *f;  
    double *g;  
    char *h;  
  
    return 0;  
}
```

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

- a. (2 pts) `&a` **int ***
- b. (2 pts) `&g` **double ****
- c. (2 pts) `argv[1][2]` **char**
- d. (2 pts) `a` **int**
- e. (2 pts) `f->data` **int**
- f. (2 pts) `argc` **int**
- g. (2 pts) `f->next` **Node ***
- h. (2 pts) `argv[0]` **char ***
- i. (2 pts) `*g` **double**
- j. (2 pts) `h` **char ***
- k. (2 pts) `f->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #108 Page: 8 Name: _____

End of Exam

total points=100

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 001 001 111 from binary to octal **117**

b. (2 pts) Convert 1001 0010 from binary to base 10 **146**

c. (2 pts) Convert 109b from hexadecimal to base 2 **0001 0000 1001 1011**

d. (2 pts) Convert 0010 1111 from base 2 to base 10 **47**

e. (2 pts) Convert 63 from base 8 to base 2 **110 011**

f. (2 pts) Convert 64 from octal to binary **110 100**

g. (2 pts) Convert 3 from octal to base 2 **011**

h. (2 pts) Convert 1100 1000 from binary to decimal **200**

i. (2 pts) Convert 010 011 001 from binary to base 8 **231**

j. (2 pts) Convert 0110 0110 from base 2 to base 10 **102**

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 banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[2][1]`? r

c. (2 pts) What is the value of `argv[0][1]`? /

d. (2 pts) What is the value of `argv[1][2]`? m

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int g;  
    Node h;  
    double p;  
    char q;  
    int *r;  
    Node *s;  
    double *t;  
    char *w;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[1][2]` **char**
- b. (2 pts) `q` **char**
- c. (2 pts) `&q` **char ***
- d. (2 pts) `*w` **char**
- e. (2 pts) `s->data` **int**
- f. (2 pts) `argc` **int**
- g. (2 pts) `argv[0]` **char ***
- h. (2 pts) `s->next` **Node ***
- i. (2 pts) `s->next->next` **Node ***
- j. (2 pts) `t` **double ***
- k. (2 pts) `&r` **int ****

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 parameters 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 parameters 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;
}
```

Exam #109 Page: 8 Name: _____

End of Exam

total points=100

Exam #110 Page: 1 Name: _____

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 0101 1001 0100 1110 from base 2 to hexadecimal **594e**

b. (2 pts) Convert 100 from base 10 to binary **0110 0100**

c. (2 pts) Convert 0100 0110 from binary to decimal **70**

d. (2 pts) Convert 139 from base 10 to binary **1000 1011**

e. (2 pts) Convert 001 000 110 from base 2 to base 8 **106**

f. (2 pts) Convert 5 from base 10 to base 2 **0101**

g. (2 pts) Convert 18 from base 10 to base 2 **0001 0010**

h. (2 pts) Convert 0011 0001 from base 2 to decimal **49**

i. (2 pts) Convert 73 from octal to binary **111 011**

j. (2 pts) Convert 1001 0010 from base 2 to decimal **146**

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 cherry grape
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[2][1]`? **r**

c. (2 pts) What is the value of `argv[1][3]`? **r**

d. (2 pts) What is the value of `argv[0][4]`? **n**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double g;  
    Node h;  
    int p;  
    char q;  
    double *r;  
    Node *s;  
    int *t;  
    char *w;  
  
    return 0;  
}
```

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

a. (2 pts) `s` **Node ***

b. (2 pts) `argv[0]` **char ***

c. (2 pts) `&h` **Node ***

d. (2 pts) `g` **double**

e. (2 pts) `s->data` **int**

f. (2 pts) `&t` **int ****

g. (2 pts) `s->next->next` **Node ***

h. (2 pts) `argc` **int**

i. (2 pts) `argv[1][2]` **char**

j. (2 pts) `s->next` **Node ***

k. (2 pts) `*w` **char**

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 parameters 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 parameters 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;
}
```

Exam #110 Page: 8 Name: _____

End of Exam

total points=100

Exam #111 Page: 1 Name: _____

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 1001 1110 from base 2 to base 10 **158**

b. (2 pts) Convert 1e29 from base 16 to base 2 **0001 1110 0010 1001**

c. (2 pts) Convert 0011 0100 from base 2 to decimal **52**

d. (2 pts) Convert 17d2 from base 16 to base 2 **0001 0111 1101 0010**

e. (2 pts) Convert 6 from base 10 to base 2 **0110**

f. (2 pts) Convert 1001 1111 from binary to base 10 **159**

g. (2 pts) Convert 188 from decimal to base 2 **1011 1100**

h. (2 pts) Convert 1011 1001 0001 0011 from base 2 to hexadecimal **b913**

i. (2 pts) Convert 101 110 000 from binary to base 8 **560**

j. (2 pts) Convert 0010 1110 from binary to decimal **46**

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 date banana guava lime
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][0]`? d

c. (2 pts) What is the value of `argv[0][2]`? r

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double d;  
    int e;  
    Node f;  
    char g;  
    double *h;  
    int *p;  
    Node *q;  
    char *r;  
  
    return 0;  
}
```

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

a. (2 pts) `h` **double ***

b. (2 pts) `argc` **int**

c. (2 pts) `q->next->next` **Node ***

d. (2 pts) `q->next` **Node ***

e. (2 pts) `&h` **double ****

f. (2 pts) `*h` **double**

g. (2 pts) `q->data` **int**

h. (2 pts) `&e` **int ***

i. (2 pts) `argv[0]` **char ***

j. (2 pts) `f` **Node**

k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #111 Page: 8 Name: _____

End of Exam

total points=100

Exam #112 Page: 1 Name: _____

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 010 110 101 from binary to octal **265**

b. (2 pts) Convert 7bdc from base 16 to base 2 **0111 1011 1101 1100**

c. (2 pts) Convert 7 from base 8 to base 2 **111**

d. (2 pts) Convert 0101 0110 0001 1111 from base 2 to hexadecimal **561f**

e. (2 pts) Convert 011 010 from binary to base 8 **32**

f. (2 pts) Convert 011 001 010 from binary to octal **312**

g. (2 pts) Convert cfb7 from hexadecimal to base 2 **1100 1111 1011 0111**

h. (2 pts) Convert 0110 1001 from base 2 to base 10 **105**

i. (2 pts) Convert 103 from base 10 to base 2 **0110 0111**

j. (2 pts) Convert 189 from decimal to base 2 **1011 1101**

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 banana cherry fig
```

a. (2 pts) What is the value of `argc` in this case? 4

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

c. (2 pts) What is the value of `argv[0][3]`? u

d. (2 pts) What is the value of `argv[1][2]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int r;  
    Node s;  
    double t;  
    char w;  
    int *x;  
    Node *y;  
    double *z;  
    char *a;  
  
    return 0;  
}
```

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

a. (2 pts) `y->data` **int**

b. (2 pts) `y->next` **Node ***

c. (2 pts) `argv[1][2]` **char**

d. (2 pts) `y` **Node ***

e. (2 pts) `s` **Node**

f. (2 pts) `argv[0]` **char ***

g. (2 pts) `&y` **Node ****

h. (2 pts) `argc` **int**

i. (2 pts) `&r` **int ***

j. (2 pts) `y->next->next` **Node ***

k. (2 pts) `*x` **int**

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 parameters 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 parameters 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;
}
```

Exam #112 Page: 8 Name: _____

End of Exam

total points=100

Exam #113 Page: 1 Name: _____

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 9f80 from hexadecimal to base 2 **1001 1111 1000 0000**

b. (2 pts) Convert 0011 0110 0000 0101 from binary to base 16 **3605**

c. (2 pts) Convert 1100 0001 0001 from binary to base 16 **c11**

d. (2 pts) Convert 1110 0010 0000 1001 from base 2 to base 16 **e209**

e. (2 pts) Convert 240 from decimal to base 2 **1111 0000**

f. (2 pts) Convert 77 from octal to base 2 **111 111**

g. (2 pts) Convert 36 from base 8 to binary **011 110**

h. (2 pts) Convert 1111 0001 from base 2 to base 10 **241**

i. (2 pts) Convert 320d from hexadecimal to binary **0011 0010 0000 1101**

j. (2 pts) Convert 88 from base 10 to base 2 **0101 1000**

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 fig date
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][1]`? **/**

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

d. (2 pts) What is the value of `argv[1][0]`? **f**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node h;  
    double p;  
    int q;  
    char r;  
    Node *s;  
    double *t;  
    int *w;  
    char *x;  
  
    return 0;  
}
```

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

a. (2 pts) `&r` **char ***

b. (2 pts) `argc` **int**

c. (2 pts) `s->data` **int**

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `s->next` **Node ***

f. (2 pts) `&s` **Node ****

g. (2 pts) `s->next->next` **Node ***

h. (2 pts) `s` **Node ***

i. (2 pts) `r` **char**

j. (2 pts) `*w` **int**

k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #113 Page: 8 Name: _____

End of Exam

total points=100

Exam #114 Page: 1 Name: _____

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 208 from base 10 to base 2 **1101 0000**

b. (2 pts) Convert 001 111 from binary to base 8 **17**

c. (2 pts) Convert 0100 0010 from binary to base 10 **66**

d. (2 pts) Convert 17 from base 8 to binary **001 111**

e. (2 pts) Convert 46de from hexadecimal to base 2 **0100 0110 1101 1110**

f. (2 pts) Convert 001 100 010 from base 2 to base 8 **142**

g. (2 pts) Convert 123 from decimal to base 2 **0111 1011**

h. (2 pts) Convert 90 from decimal to base 2 **0101 1010**

i. (2 pts) Convert 211 from base 10 to base 2 **1101 0011**

j. (2 pts) Convert 853d from base 16 to base 2 **1000 0101 0011 1101**

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 date mango guava fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][1]`? a

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

d. (2 pts) What is the value of `argv[0][6]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double h;  
    int p;  
    Node q;  
    char r;  
    double *s;  
    int *t;  
    Node *w;  
    char *x;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[0]` **char ***
- b. (2 pts) `w->next` **Node ***
- c. (2 pts) `&t` **int ****
- d. (2 pts) `h` **double**
- e. (2 pts) `&q` **Node ***
- f. (2 pts) `w->next->next` **Node ***
- g. (2 pts) `argv[1][2]` **char**
- h. (2 pts) `*t` **int**
- i. (2 pts) `argc` **int**
- j. (2 pts) `s` **double ***
- k. (2 pts) `w->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #114 Page: 8 Name: _____

End of Exam

total points=100

Exam #115 Page: 1 Name: _____

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 0001 0101 1100 0001 from binary to hexadecimal **15c1**

b. (2 pts) Convert 110 000 011 from base 2 to octal **603**

c. (2 pts) Convert 14 from base 8 to base 2 **001 100**

d. (2 pts) Convert 62 from base 8 to base 2 **110 010**

e. (2 pts) Convert 12 from octal to binary **001 010**

f. (2 pts) Convert 62 from base 8 to binary **110 010**

g. (2 pts) Convert 25c7 from base 16 to base 2 **0010 0101 1100 0111**

h. (2 pts) Convert 1110 0001 1100 1001 from binary to base 16 **e1c9**

i. (2 pts) Convert 1001 1101 from binary to decimal **157**

j. (2 pts) Convert 33 from base 10 to base 2 **0010 0001**

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 kiwi date mango apple
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? a

c. (2 pts) What is the value of `argv[1][0]`? k

d. (2 pts) What is the value of `argv[0][3]`? u

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int w;  
    double x;  
    Node y;  
    char z;  
    int *a;  
    double *b;  
    Node *c;  
    char *d;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[1][2]` **char**

b. (2 pts) `*d` **char**

c. (2 pts) `c->next` **Node ***

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `argc` **int**

f. (2 pts) `c->data` **int**

g. (2 pts) `c->next->next` **Node ***

h. (2 pts) `y` **Node**

i. (2 pts) `&d` **char ****

j. (2 pts) `&x` **double ***

k. (2 pts) `d` **char ***

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 parameters 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 parameters 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;
}
```

Exam #115 Page: 8 Name: _____

End of Exam

total points=100

Exam #116 Page: 1 Name: _____

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 0110 1110 0001 1110 from base 2 to hexadecimal **6e1e**

b. (2 pts) Convert 31 from base 8 to binary **011 001**

c. (2 pts) Convert 110 101 011 from binary to octal **653**

d. (2 pts) Convert 1000 0101 from base 2 to decimal **133**

e. (2 pts) Convert 1001 1010 from binary to base 10 **154**

f. (2 pts) Convert cc98 from base 16 to binary **1100 1100 1001 1000**

g. (2 pts) Convert 35 from octal to base 2 **011 101**

h. (2 pts) Convert 1000 1000 from binary to base 10 **136**

i. (2 pts) Convert 1001 0000 from base 2 to base 10 **144**

j. (2 pts) Convert 44 from base 10 to binary **0010 1100**

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 cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][1]`? /

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

d. (2 pts) What is the value of `argv[1][0]`? l

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node g;  
    int h;  
    double p;  
    char q;  
    Node *r;  
    int *s;  
    double *t;  
    char *w;  
  
    return 0;  
}
```

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

- a. (2 pts) `p` **double**
- b. (2 pts) `argv[0]` **char ***
- c. (2 pts) `*t` **double**
- d. (2 pts) `&h` **int ***
- e. (2 pts) `s` **int ***
- f. (2 pts) `argc` **int**
- g. (2 pts) `&t` **double ****
- h. (2 pts) `r->next->next` **Node ***
- i. (2 pts) `r->data` **int**
- j. (2 pts) `argv[1][2]` **char**
- k. (2 pts) `r->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #116 Page: 8 Name: _____

End of Exam

total points=100

Exam #117 Page: 1 Name: _____

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 1011 0010 from binary to decimal **178**

b. (2 pts) Convert 30 from base 10 to base 2 **0001 1110**

c. (2 pts) Convert 110 000 111 from binary to octal **607**

d. (2 pts) Convert 0001 0001 from base 2 to decimal **17**

e. (2 pts) Convert 7dee from hexadecimal to binary **0111 1101 1110 1110**

f. (2 pts) Convert 102 from decimal to binary **0110 0110**

g. (2 pts) Convert 10 from base 8 to base 2 **001 000**

h. (2 pts) Convert 1111 from base 2 to base 10 **15**

i. (2 pts) Convert 90 from decimal to binary **0101 1010**

j. (2 pts) Convert 1100 1000 from binary to decimal **200**

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 cherry fig
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[2][0]`? **f**

c. (2 pts) What is the value of `argv[0][3]`? **u**

d. (2 pts) What is the value of `argv[1][5]`? **y**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node d;  
    double e;  
    int f;  
    char g;  
    Node *h;  
    double *p;  
    int *q;  
    char *r;  
  
    return 0;  
}
```

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

- a. (2 pts) `h->data` **int**
- b. (2 pts) `&h` **Node ****
- c. (2 pts) `argv[1][2]` **char**
- d. (2 pts) `&d` **Node ***
- e. (2 pts) `h` **Node ***
- f. (2 pts) `h->next->next` **Node ***
- g. (2 pts) `argc` **int**
- h. (2 pts) `*h` **Node**
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `h->next` **Node ***
- k. (2 pts) `d` **Node**

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 parameters 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 parameters 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;
}
```

Exam #117 Page: 8 Name: _____

End of Exam

total points=100

Exam #118 Page: 1 Name: _____

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 71 from base 8 to base 2 **111 001**

b. (2 pts) Convert 1111 0000 from binary to decimal **240**

c. (2 pts) Convert 249 from decimal to base 2 **1111 1001**

d. (2 pts) Convert 109 from base 10 to binary **0110 1101**

e. (2 pts) Convert 1101 0100 0000 1100 from binary to hexadecimal **d40c**

f. (2 pts) Convert 1001 1000 1101 1110 from base 2 to base 16 **98de**

g. (2 pts) Convert 10 from octal to binary **001 000**

h. (2 pts) Convert 0111 1000 1101 1010 from base 2 to base 16 **78da**

i. (2 pts) Convert 252 from base 10 to base 2 **1111 1100**

j. (2 pts) Convert 244 from decimal to binary **1111 0100**

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 guava apple fig date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][4]`? a

c. (2 pts) What is the value of `argv[2][3]`? l

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double e;  
    int f;  
    Node g;  
    char h;  
    double *p;  
    int *q;  
    Node *r;  
    char *s;  
  
    return 0;  
}
```

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

a. (2 pts) r->next->next **Node ***

b. (2 pts) r->data **int**

c. (2 pts) f **int**

d. (2 pts) s **char ***

e. (2 pts) argv[1][2] **char**

f. (2 pts) *s **char**

g. (2 pts) &r **Node ****

h. (2 pts) argv[0] **char ***

i. (2 pts) r->next **Node ***

j. (2 pts) argc **int**

k. (2 pts) &g **Node ***

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 parameters 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 parameters 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;
}
```

Exam #118 Page: 8 Name: _____

End of Exam

total points=100

Exam #119 Page: 1 Name: _____

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 41 from base 10 to base 2 **0010 1001**

b. (2 pts) Convert 170 from base 10 to base 2 **1010 1010**

c. (2 pts) Convert 231 from base 10 to binary **1110 0111**

d. (2 pts) Convert f9de from base 16 to base 2 **1111 1001 1101 1110**

e. (2 pts) Convert 55 from octal to binary **101 101**

f. (2 pts) Convert 32bb from base 16 to base 2 **0011 0010 1011 1011**

g. (2 pts) Convert 63 from base 8 to binary **110 011**

h. (2 pts) Convert 1000 1010 from binary to base 16 **8a**

i. (2 pts) Convert 198 from decimal to binary **1100 0110**

j. (2 pts) Convert 144 from decimal to binary **1001 0000**

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 mango cherry banana guava
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][4]`? n

c. (2 pts) What is the value of `argv[1][3]`? g

d. (2 pts) What is the value of `argv[2][3]`? r

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double a;  
    int b;  
    Node c;  
    char d;  
    double *e;  
    int *f;  
    Node *g;  
    char *h;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[0]` **char ***
- b. (2 pts) `*f` **int**
- c. (2 pts) `g` **Node ***
- d. (2 pts) `&b` **int ***
- e. (2 pts) `g->data` **int**
- f. (2 pts) `g->next->next` **Node ***
- g. (2 pts) `d` **char**
- h. (2 pts) `&e` **double ****
- i. (2 pts) `g->next` **Node ***
- j. (2 pts) `argv[1][2]` **char**
- k. (2 pts) `argc` **int**

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 parameters 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 parameters 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;
}
```

Exam #119 Page: 8 Name: _____

End of Exam

total points=100

Exam #120 Page: 1 Name: _____

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 129 from base 10 to base 2 **1000 0001**

b. (2 pts) Convert 010 011 010 from base 2 to octal **232**

c. (2 pts) Convert 8d33 from hexadecimal to base 2 **1000 1101 0011 0011**

d. (2 pts) Convert 101 101 010 from base 2 to base 8 **552**

e. (2 pts) Convert 39 from decimal to binary **0010 0111**

f. (2 pts) Convert 0011 0100 from base 2 to base 10 **52**

g. (2 pts) Convert 31 from decimal to binary **0001 1111**

h. (2 pts) Convert 1010 0110 from binary to base 10 **166**

i. (2 pts) Convert 101 110 001 from base 2 to base 8 **561**

j. (2 pts) Convert 1001 1100 from binary to base 10 **156**

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 date apple guava
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][6]`? t

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

d. (2 pts) What is the value of `argv[2][3]`? l

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node r;  
    int s;  
    double t;  
    char w;  
    Node *x;  
    int *y;  
    double *z;  
    char *a;  
  
    return 0;  
}
```

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

a. (2 pts) `w`

`char`

b. (2 pts) `x->next`

`Node *`

c. (2 pts) `argc`

`int`

d. (2 pts) `&a`

`char **`

e. (2 pts) `*x`

`Node`

f. (2 pts) `argv[0]`

`char *`

g. (2 pts) `argv[1][2]`

`char`

h. (2 pts) `&s`

`int *`

i. (2 pts) `x->data`

`int`

j. (2 pts) `y`

`int *`

k. (2 pts) `x->next->next`

`Node *`

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 parameters 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 parameters 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;
}
```

Exam #120 Page: 8 Name: _____

End of Exam

total points=100

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 61 from base 8 to binary **110 001**

b. (2 pts) Convert 74b from base 16 to base 2 **0111 0100 1011**

c. (2 pts) Convert 0111 1011 from base 2 to base 10 **123**

d. (2 pts) Convert 20 from octal to binary **010 000**

e. (2 pts) Convert 1011 0001 1011 from binary to hexadecimal **b1b**

f. (2 pts) Convert 110 011 011 from binary to octal **633**

g. (2 pts) Convert 201 from base 10 to binary **1100 1001**

h. (2 pts) Convert 0010 1110 1010 1100 from binary to base 16 **2eac**

i. (2 pts) Convert 1000 0010 from binary to decimal **130**

j. (2 pts) Convert 0011 1000 from base 2 to base 10 **56**

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 date fig
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[1][3]`? e

c. (2 pts) What is the value of `argv[2][0]`? f

d. (2 pts) What is the value of `argv[0][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double z;  
    Node a;  
    int b;  
    char c;  
    double *d;  
    Node *e;  
    int *f;  
    char *g;  
  
    return 0;  
}
```

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

- a. (2 pts) e->next **Node ***
- b. (2 pts) d **double ***
- c. (2 pts) argc **int**
- d. (2 pts) e->data **int**
- e. (2 pts) e->next->next **Node ***
- f. (2 pts) argv[1][2] **char**
- g. (2 pts) *f **int**
- h. (2 pts) argv[0] **char ***
- i. (2 pts) a **Node**
- j. (2 pts) &e **Node ****
- k. (2 pts) &z **double ***

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 parameters 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 parameters 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;
}
```

Exam #121 Page: 8 Name: _____

End of Exam

total points=100

Exam #122 Page: 1 Name: _____

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 f7dc from hexadecimal to binary **1111 0111 1101 1100**

b. (2 pts) Convert d8d1 from hexadecimal to binary **1101 1000 1101 0001**

c. (2 pts) Convert 54 from base 8 to binary **101 100**

d. (2 pts) Convert 157 from base 10 to base 2 **1001 1101**

e. (2 pts) Convert 0110 0001 0011 1001 from binary to hexadecimal **6139**

f. (2 pts) Convert 0000 from binary to base 10 **0**

g. (2 pts) Convert 110 010 110 from binary to base 8 **626**

h. (2 pts) Convert 1001 0111 1001 1011 from base 2 to base 16 **979b**

i. (2 pts) Convert 0010 0100 from base 2 to base 10 **36**

j. (2 pts) Convert 0110 0100 from base 2 to decimal **100**

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 fig grape lime lemon
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? r

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

d. (2 pts) What is the value of `argv[0][6]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int a;  
    double b;  
    Node c;  
    char d;  
    int *e;  
    double *f;  
    Node *g;  
    char *h;  
  
    return 0;  
}
```

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

- a. (2 pts) `g->data` **int**
- b. (2 pts) `&g` **Node ****
- c. (2 pts) `g->next` **Node ***
- d. (2 pts) `argv[0]` **char ***
- e. (2 pts) `argc` **int**
- f. (2 pts) `argv[1][2]` **char**
- g. (2 pts) `&d` **char ***
- h. (2 pts) `h` **char ***
- i. (2 pts) `g->next->next` **Node ***
- j. (2 pts) `c` **Node**
- k. (2 pts) `*f` **double**

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 parameters 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 parameters 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;
}
```

Exam #122 Page: 8 Name: _____

End of Exam

total points=100

Exam #123 Page: 1 Name: _____

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 001 111 001 from binary to base 8 **171**

b. (2 pts) Convert 44 from octal to base 2 **100 100**

c. (2 pts) Convert 9f53 from hexadecimal to binary **1001 1111 0101 0011**

d. (2 pts) Convert 001 010 011 from base 2 to octal **123**

e. (2 pts) Convert 010 001 001 from binary to base 8 **211**

f. (2 pts) Convert 100 110 100 from base 2 to base 8 **464**

g. (2 pts) Convert 011 101 010 from base 2 to base 8 **352**

h. (2 pts) Convert 0001 1111 from base 2 to base 10 **31**

i. (2 pts) Convert 238 from decimal to binary **1110 1110**

j. (2 pts) Convert 0000 from base 2 to base 10 **0**

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 kiwi grape guava cherry
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][0]`? k

c. (2 pts) What is the value of `argv[0][2]`? r

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int x;  
    double y;  
    Node z;  
    char a;  
    int *b;  
    double *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

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

a. (2 pts) `y` **double**

b. (2 pts) `argv[0]` **char ***

c. (2 pts) `&z` **Node ***

d. (2 pts) `d->next` **Node ***

e. (2 pts) `*e` **char**

f. (2 pts) `d` **Node ***

g. (2 pts) `d->next->next` **Node ***

h. (2 pts) `argc` **int**

i. (2 pts) `&c` **double ****

j. (2 pts) `d->data` **int**

k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #123 Page: 8 Name: _____

End of Exam

total points=100

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 100 101 010 from binary to base 8 **452**

b. (2 pts) Convert 0011 0101 1100 0110 from binary to hexadecimal **35c6**

c. (2 pts) Convert 0100 0100 1011 0101 from base 2 to base 16 **44b5**

d. (2 pts) Convert 1110 0101 0001 1001 from binary to hexadecimal **e519**

e. (2 pts) Convert 55 from octal to binary **101 101**

f. (2 pts) Convert 1001 1011 0111 1111 from base 2 to base 16 **9b7f**

g. (2 pts) Convert 198 from base 10 to base 2 **1100 0110**

h. (2 pts) Convert 1100 0101 from base 2 to base 10 **197**

i. (2 pts) Convert 225 from decimal to base 2 **1110 0001**

j. (2 pts) Convert 011 000 from base 2 to base 8 **30**

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 fig lime apple
```

a. (2 pts) What is the value of `argc` in this case? 4

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

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

d. (2 pts) What is the value of `argv[1][0]`? f

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int h;  
    Node p;  
    double q;  
    char r;  
    int *s;  
    Node *t;  
    double *w;  
    char *x;  
  
    return 0;  
}
```

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

a. (2 pts) `&x` **char ****

b. (2 pts) `argv[0]` **char ***

c. (2 pts) `&q` **double ***

d. (2 pts) `argv[1][2]` **char**

e. (2 pts) `t->next` **Node ***

f. (2 pts) `h` **int**

g. (2 pts) `t->next->next` **Node ***

h. (2 pts) `*w` **double**

i. (2 pts) `t->data` **int**

j. (2 pts) `argc` **int**

k. (2 pts) `s` **int ***

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 parameters 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 parameters 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;
}
```

Exam #124 Page: 8 Name: _____

End of Exam

total points=100

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 217 from base 10 to base 2 **1101 1001**

b. (2 pts) Convert 73 from base 8 to binary **111 011**

c. (2 pts) Convert 0011 0010 1001 0101 from binary to hexadecimal **3295**

d. (2 pts) Convert 0111 0001 0000 0100 from binary to base 16 **7104**

e. (2 pts) Convert 100 110 000 from binary to base 8 **460**

f. (2 pts) Convert 355c from base 16 to binary **0011 0101 0101 1100**

g. (2 pts) Convert 112 from decimal to base 2 **0111 0000**

h. (2 pts) Convert 0100 1101 0110 1110 from binary to base 16 **4d6e**

i. (2 pts) Convert 1010 1011 from base 2 to decimal **171**

j. (2 pts) Convert 101 001 111 from binary to octal **517**

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 date grape
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[2][0]`? g

c. (2 pts) What is the value of `argv[0][4]`? n

d. (2 pts) What is the value of `argv[1][0]`? d

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double e;  
    Node f;  
    int g;  
    char h;  
    double *p;  
    Node *q;  
    int *r;  
    char *s;  
  
    return 0;  
}
```

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

a. (2 pts) `&r` **int ****

b. (2 pts) `q->next` **Node ***

c. (2 pts) `&f` **Node ***

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `g` **int**

f. (2 pts) `argv[1][2]` **char**

g. (2 pts) `q->next->next` **Node ***

h. (2 pts) `s` **char ***

i. (2 pts) `q->data` **int**

j. (2 pts) `argc` **int**

k. (2 pts) `*p` **double**

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 parameters 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 parameters 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;
}
```

Exam #125 Page: 8 Name: _____

End of Exam

total points=100

Exam #126 Page: 1 Name: _____

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 1011 from base 2 to decimal **11**

b. (2 pts) Convert 1100 0001 0111 0100 from binary to hexadecimal **c174**

c. (2 pts) Convert 32 from octal to base 2 **011 010**

d. (2 pts) Convert 63 from octal to base 2 **110 011**

e. (2 pts) Convert 1110 1110 from binary to base 10 **238**

f. (2 pts) Convert 31 from octal to binary **011 001**

g. (2 pts) Convert 115 from decimal to binary **0111 0011**

h. (2 pts) Convert 1011 0110 0101 1101 from binary to base 16 **b65d**

i. (2 pts) Convert 76 from base 10 to base 2 **0100 1100**

j. (2 pts) Convert 110 101 000 from base 2 to base 8 **650**

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 date lime
```

a. (2 pts) What is the value of `argc` in this case? **3**

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

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double w;  
    Node x;  
    int y;  
    char z;  
    double *a;  
    Node *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

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

- a. (2 pts) argv[1][2] **char**
- b. (2 pts) *d **char**
- c. (2 pts) d **char ***
- d. (2 pts) z **char**
- e. (2 pts) argv[0] **char ***
- f. (2 pts) b->next->next **Node ***
- g. (2 pts) b->data **int**
- h. (2 pts) &d **char ****
- i. (2 pts) b->next **Node ***
- j. (2 pts) &z **char ***
- k. (2 pts) argc **int**

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 parameters 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 parameters 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;
}
```

Exam #126 Page: 8 Name: _____

End of Exam

total points=100

Exam #127 Page: 1 Name: _____

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 5039 from hexadecimal to binary **0101 0000 0011 1001**

b. (2 pts) Convert 0111 1011 from binary to base 10 **123**

c. (2 pts) Convert 010 101 101 from binary to base 8 **255**

d. (2 pts) Convert 26 from octal to binary **010 110**

e. (2 pts) Convert d1b2 from hexadecimal to base 2 **1101 0001 1011 0010**

f. (2 pts) Convert 0 from base 8 to base 2 **000**

g. (2 pts) Convert 29 from decimal to binary **0001 1101**

h. (2 pts) Convert 0011 1110 from base 2 to base 10 **62**

i. (2 pts) Convert 0001 0110 from base 2 to decimal **22**

j. (2 pts) Convert 011 100 000 from binary to base 8 **340**

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 guava lime cherry grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][2]`? a

c. (2 pts) What is the value of `argv[2][3]`? e

d. (2 pts) What is the value of `argv[0][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int r;  
    double s;  
    Node t;  
    char w;  
    int *x;  
    double *y;  
    Node *z;  
    char *a;  
  
    return 0;  
}
```

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

a. (2 pts) `&y` **double ****

b. (2 pts) `y` **double ***

c. (2 pts) `&t` **Node ***

d. (2 pts) `z->next` **Node ***

e. (2 pts) `argc` **int**

f. (2 pts) `argv[1][2]` **char**

g. (2 pts) `z->next->next` **Node ***

h. (2 pts) `t` **Node**

i. (2 pts) `argv[0]` **char ***

j. (2 pts) `*y` **double**

k. (2 pts) `z->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #127 Page: 8 Name: _____

End of Exam

total points=100

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 0110 0101 from binary to base 10 **101**

b. (2 pts) Convert 3ab9 from base 16 to binary **0011 1010 1011 1001**

c. (2 pts) Convert 1101 0111 1110 0010 from base 2 to base 16 **d7e2**

d. (2 pts) Convert 167 from decimal to binary **1010 0111**

e. (2 pts) Convert 0110 0000 from base 2 to base 10 **96**

f. (2 pts) Convert 011 000 011 from base 2 to base 8 **303**

g. (2 pts) Convert 57 from base 8 to binary **101 111**

h. (2 pts) Convert 0010 1010 from binary to decimal **42**

i. (2 pts) Convert 0011 0101 from binary to base 10 **53**

j. (2 pts) Convert 011 101 from base 2 to base 8 **35**

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 kiwi cherry fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[2][1]`? h

c. (2 pts) What is the value of `argv[1][3]`? i

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int g;  
    Node h;  
    double p;  
    char q;  
    int *r;  
    Node *s;  
    double *t;  
    char *w;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[1][2]` **char**
- b. (2 pts) `&w` **char ****
- c. (2 pts) `s->data` **int**
- d. (2 pts) `s->next` **Node ***
- e. (2 pts) `*r` **int**
- f. (2 pts) `r` **int ***
- g. (2 pts) `argv[0]` **char ***
- h. (2 pts) `&h` **Node ***
- i. (2 pts) `p` **double**
- j. (2 pts) `argc` **int**
- k. (2 pts) `s->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #128 Page: 8 Name: _____

End of Exam

total points=100

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 aa12 from hexadecimal to binary **1010 1010 0001 0010**

b. (2 pts) Convert 111 101 001 from base 2 to base 8 **751**

c. (2 pts) Convert 197 from decimal to binary **1100 0101**

d. (2 pts) Convert 0011 0011 from binary to base 10 **51**

e. (2 pts) Convert 4386 from base 16 to binary **0100 0011 1000 0110**

f. (2 pts) Convert 76 from base 8 to binary **111 110**

g. (2 pts) Convert 31 from octal to base 2 **011 001**

h. (2 pts) Convert 1011 0001 1101 1110 from binary to hexadecimal **b1de**

i. (2 pts) Convert 77 from base 8 to binary **111 111**

j. (2 pts) Convert 101 010 101 from binary to base 8 **525**

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 guava banana
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[2][2]`? **n**

c. (2 pts) What is the value of `argv[0][0]`? **.**

d. (2 pts) What is the value of `argv[1][1]`? **u**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node s;  
    double t;  
    int w;  
    char x;  
    Node *y;  
    double *z;  
    int *a;  
    char *b;  
  
    return 0;  
}
```

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

a. (2 pts) `argc`

`int`

b. (2 pts) `&z`

`double **`

c. (2 pts) `argv[0]`

`char *`

d. (2 pts) `s`

`Node`

e. (2 pts) `y->data`

`int`

f. (2 pts) `y->next->next`

`Node *`

g. (2 pts) `y->next`

`Node *`

h. (2 pts) `argv[1][2]`

`char`

i. (2 pts) `b`

`char *`

j. (2 pts) `*a`

`int`

k. (2 pts) `&s`

`Node *`

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 parameters 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 parameters 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;
}
```

Exam #129 Page: 8 Name: _____

End of Exam

total points=100

Exam #130 Page: 1 Name: _____

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 219 from base 10 to base 2 **1101 1011**

b. (2 pts) Convert 61 from octal to binary **110 001**

c. (2 pts) Convert 77 from octal to base 2 **111 111**

d. (2 pts) Convert 1001 0000 0000 0010 from binary to hexadecimal **9002**

e. (2 pts) Convert 1001 1001 1010 0011 from base 2 to hexadecimal **99a3**

f. (2 pts) Convert 45 from decimal to base 2 **0010 1101**

g. (2 pts) Convert 6855 from base 16 to binary **0110 1000 0101 0101**

h. (2 pts) Convert 0001 1010 1100 1101 from base 2 to hexadecimal **1acd**

i. (2 pts) Convert 50 from octal to base 2 **101 000**

j. (2 pts) Convert 110 101 101 from binary to octal **655**

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 banana apple cherry mango
```

a. (2 pts) What is the value of `argc` in this case? 5

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

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

d. (2 pts) What is the value of `argv[1][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double t;  
    int w;  
    Node x;  
    char y;  
    double *z;  
    int *a;  
    Node *b;  
    char *c;  
  
    return 0;  
}
```

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

a. (2 pts) b->next->next **Node ***

b. (2 pts) &y **char ***

c. (2 pts) b->next **Node ***

d. (2 pts) argc **int**

e. (2 pts) argv[1][2] **char**

f. (2 pts) w **int**

g. (2 pts) argv[0] **char ***

h. (2 pts) b->data **int**

i. (2 pts) &b **Node ****

j. (2 pts) c **char ***

k. (2 pts) *a **int**

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 parameters 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 parameters 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;
}
```

Exam #130 Page: 8 Name: _____

End of Exam

total points=100

Exam #131 Page: 1 Name: _____

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 0010 0000 0101 0100 from base 2 to base 16 **2054**

b. (2 pts) Convert 128 from decimal to base 2 **1000 0000**

c. (2 pts) Convert ea01 from hexadecimal to base 2 **1110 1010 0000 0001**

d. (2 pts) Convert 27 from decimal to base 2 **0001 1011**

e. (2 pts) Convert 37 from base 8 to base 2 **011 111**

f. (2 pts) Convert 1100 0111 from binary to base 10 **199**

g. (2 pts) Convert 4 from octal to binary **100**

h. (2 pts) Convert 1010 0010 from binary to decimal **162**

i. (2 pts) Convert 011 010 110 from base 2 to octal **326**

j. (2 pts) Convert 011 100 101 from binary to octal **345**

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 mango apple banana date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][0]`? m

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

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double q;  
    int r;  
    Node s;  
    char t;  
    double *w;  
    int *x;  
    Node *y;  
    char *z;  
  
    return 0;  
}
```

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

a. (2 pts) `y->next->next` **Node ***

b. (2 pts) `y->data` **int**

c. (2 pts) `*z` **char**

d. (2 pts) `argc` **int**

e. (2 pts) `y` **Node ***

f. (2 pts) `q` **double**

g. (2 pts) `y->next` **Node ***

h. (2 pts) `argv[0]` **char ***

i. (2 pts) `&s` **Node ***

j. (2 pts) `&w` **double ****

k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #131 Page: 8 Name: _____

End of Exam

total points=100

Exam #132 Page: 1 Name: _____

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 36 from base 8 to base 2 **011 110**

b. (2 pts) Convert 10 from octal to binary **001 000**

c. (2 pts) Convert 1000 1111 0110 0011 from binary to hexadecimal **8f63**

d. (2 pts) Convert 110 101 110 from binary to base 8 **656**

e. (2 pts) Convert 237 from base 10 to binary **1110 1101**

f. (2 pts) Convert 62 from base 8 to base 2 **110 010**

g. (2 pts) Convert 011 000 111 from base 2 to octal **307**

h. (2 pts) Convert 72 from base 10 to base 2 **0100 1000**

i. (2 pts) Convert 010 111 011 from binary to base 8 **273**

j. (2 pts) Convert 37 from octal to base 2 **011 111**

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 apple lemon lime
```

a. (2 pts) What is the value of `argc` in this case? **4**

b. (2 pts) What is the value of `argv[0][4]`? **n**

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

d. (2 pts) What is the value of `argv[2][1]`? **e**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node c;  
    int d;  
    double e;  
    char f;  
    Node *g;  
    int *h;  
    double *p;  
    char *q;  
  
    return 0;  
}
```

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

- a. (2 pts) `*p` **double**
- b. (2 pts) `f` **char**
- c. (2 pts) `g->next` **Node ***
- d. (2 pts) `g->next->next` **Node ***
- e. (2 pts) `&e` **double ***
- f. (2 pts) `&q` **char ****
- g. (2 pts) `g->data` **int**
- h. (2 pts) `argc` **int**
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `argv[1][2]` **char**
- k. (2 pts) `g` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #132 Page: 8 Name: _____

End of Exam

total points=100

Exam #133 Page: 1 Name: _____

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 1011 1101 from base 2 to base 10 **189**

b. (2 pts) Convert 221 from base 10 to base 2 **1101 1101**

c. (2 pts) Convert 37 from octal to binary **011 111**

d. (2 pts) Convert 011 000 110 from binary to octal **306**

e. (2 pts) Convert 1101 0000 1011 0011 from base 2 to hexadecimal **d0b3**

f. (2 pts) Convert 0110 0010 1111 1001 from base 2 to base 16 **62f9**

g. (2 pts) Convert 13 from decimal to binary **1101**

h. (2 pts) Convert 1101 0000 1010 0000 from binary to base 16 **d0a0**

i. (2 pts) Convert 11 from base 8 to binary **001 001**

j. (2 pts) Convert 6 from base 8 to binary **110**

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 lime fig
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[1][2]`? m

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

d. (2 pts) What is the value of `argv[2][1]`? i

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double z;  
    Node a;  
    int b;  
    char c;  
    double *d;  
    Node *e;  
    int *f;  
    char *g;  
  
    return 0;  
}
```

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

- a. (2 pts) `&e` **Node ****
- b. (2 pts) `argc` **int**
- c. (2 pts) `argv[0]` **char ***
- d. (2 pts) `&a` **Node ***
- e. (2 pts) `a` **Node**
- f. (2 pts) `e->next` **Node ***
- g. (2 pts) `argv[1][2]` **char**
- h. (2 pts) `g` **char ***
- i. (2 pts) `e->next->next` **Node ***
- j. (2 pts) `e->data` **int**
- k. (2 pts) `*d` **double**

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 parameters 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 parameters 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;
}
```

Exam #133 Page: 8 Name: _____

End of Exam

total points=100

Exam #134 Page: 1 Name: _____

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 111 011 101 from binary to octal **735**

b. (2 pts) Convert 1010 1111 from binary to decimal **175**

c. (2 pts) Convert 101 100 111 from binary to octal **547**

d. (2 pts) Convert 1011 1111 from binary to base 10 **191**

e. (2 pts) Convert 11 from base 8 to binary **001 001**

f. (2 pts) Convert 1001 0101 0110 0011 from binary to hexadecimal **9563**

g. (2 pts) Convert 4 from base 8 to base 2 **100**

h. (2 pts) Convert 0011 1001 1000 1111 from base 2 to hexadecimal **398f**

i. (2 pts) Convert 110 010 010 from binary to base 8 **622**

j. (2 pts) Convert 010 001 100 from binary to base 8 **214**

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 lime lemon guava cherry
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][2]`? r

c. (2 pts) What is the value of `argv[2][4]`? n

d. (2 pts) What is the value of `argv[1][1]`? i

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int p;  
    double q;  
    Node r;  
    char s;  
    int *t;  
    double *w;  
    Node *x;  
    char *y;  
  
    return 0;  
}
```

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

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

b. (2 pts) s **char**

c. (2 pts) argc **int**

d. (2 pts) x **Node ***

e. (2 pts) *y **char**

f. (2 pts) &s **char ***

g. (2 pts) x->data **int**

h. (2 pts) &y **char ****

i. (2 pts) x->next **Node ***

j. (2 pts) x->next->next **Node ***

k. (2 pts) argv[0] **char ***

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 parameters 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 parameters 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;
}
```

Exam #134 Page: 8 Name: _____

End of Exam

total points=100

Exam #135 Page: 1 Name: _____

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 51 from base 10 to base 2 **0011 0011**

b. (2 pts) Convert 0110 1001 0011 0011 from binary to base 16 **6933**

c. (2 pts) Convert 101 000 011 from base 2 to octal **503**

d. (2 pts) Convert 0100 1011 from binary to decimal **75**

e. (2 pts) Convert 010 100 from base 2 to base 8 **24**

f. (2 pts) Convert 0010 1111 0100 0000 from binary to base 16 **2f40**

g. (2 pts) Convert 56 from octal to base 2 **101 110**

h. (2 pts) Convert 1100 0001 from base 2 to base 10 **193**

i. (2 pts) Convert 44 from base 8 to binary **100 100**

j. (2 pts) Convert 111 000 100 from base 2 to base 8 **704**

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 grape lemon lime fig
```

a. (2 pts) What is the value of `argc` in this case? 5

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

c. (2 pts) What is the value of `argv[0][6]`? t

d. (2 pts) What is the value of `argv[1][3]`? p

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double f;  
    int g;  
    Node h;  
    char p;  
    double *q;  
    int *r;  
    Node *s;  
    char *t;  
  
    return 0;  
}
```

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

a. (2 pts) `argc`

`int`

b. (2 pts) `s->next`

`Node *`

c. (2 pts) `*r`

`int`

d. (2 pts) `argv[0]`

`char *`

e. (2 pts) `&p`

`char *`

f. (2 pts) `&r`

`int **`

g. (2 pts) `r`

`int *`

h. (2 pts) `g`

`int`

i. (2 pts) `s->next->next`

`Node *`

j. (2 pts) `argv[1][2]`

`char`

k. (2 pts) `s->data`

`int`

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 parameters 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 parameters 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;
}
```

Exam #135 Page: 8 Name: _____

End of Exam

total points=100

Exam #136 Page: 1 Name: _____

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 140 from decimal to base 2 **1000 1100**

b. (2 pts) Convert 11 from decimal to base 2 **1011**

c. (2 pts) Convert 0100 0110 from binary to base 10 **70**

d. (2 pts) Convert 728 from base 16 to base 2 **0111 0010 1000**

e. (2 pts) Convert 7a95 from base 16 to binary **0111 1010 1001 0101**

f. (2 pts) Convert 0011 0000 from binary to base 10 **48**

g. (2 pts) Convert 11 from base 10 to base 2 **1011**

h. (2 pts) Convert 0110 0111 1011 0001 from base 2 to base 16 **67b1**

i. (2 pts) Convert 41 from octal to base 2 **100 001**

j. (2 pts) Convert 73 from base 8 to binary **111 011**

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 grape fig date
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][0]`? g

c. (2 pts) What is the value of `argv[0][6]`? t

d. (2 pts) What is the value of `argv[2][2]`? g

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int y;  
    Node z;  
    double a;  
    char b;  
    int *c;  
    Node *d;  
    double *e;  
    char *f;  
  
    return 0;  
}
```

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

a. (2 pts) `c` **int ***

b. (2 pts) `argc` **int**

c. (2 pts) `*c` **int**

d. (2 pts) `d->data` **int**

e. (2 pts) `argv[0]` **char ***

f. (2 pts) `d->next` **Node ***

g. (2 pts) `d->next->next` **Node ***

h. (2 pts) `argv[1][2]` **char**

i. (2 pts) `y` **int**

j. (2 pts) `&a` **double ***

k. (2 pts) `&c` **int ****

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 parameters 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 parameters 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;
}
```

Exam #136 Page: 8 Name: _____

End of Exam

total points=100

Exam #137 Page: 1 Name: _____

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 64 from base 8 to base 2 **110 100**

b. (2 pts) Convert c628 from base 16 to binary **1100 0110 0010 1000**

c. (2 pts) Convert 15 from base 8 to base 2 **001 101**

d. (2 pts) Convert 9312 from base 16 to base 2 **1001 0011 0001 0010**

e. (2 pts) Convert 0101 1101 1110 0000 from base 2 to base 16 **5de0**

f. (2 pts) Convert 110 010 100 from base 2 to base 8 **624**

g. (2 pts) Convert 181 from decimal to base 2 **1011 0101**

h. (2 pts) Convert 1110 1111 0110 0001 from base 2 to base 16 **ef61**

i. (2 pts) Convert 010 011 111 from base 2 to octal **237**

j. (2 pts) Convert 1000 1001 1101 1100 from binary to hexadecimal **89dc**

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 banana lime
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][0]`? **.**

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

d. (2 pts) What is the value of `argv[2][3]`? **e**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double t;  
    Node w;  
    int x;  
    char y;  
    double *z;  
    Node *a;  
    int *b;  
    char *c;  
  
    return 0;  
}
```

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

a. (2 pts) argc **int**

b. (2 pts) a->next->next **Node ***

c. (2 pts) &w **Node ***

d. (2 pts) argv[1][2] **char**

e. (2 pts) a->data **int**

f. (2 pts) c **char ***

g. (2 pts) y **char**

h. (2 pts) *b **int**

i. (2 pts) a->next **Node ***

j. (2 pts) &b **int ****

k. (2 pts) argv[0] **char ***

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 parameters 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 parameters 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;
}
```

Exam #137 Page: 8 Name: _____

End of Exam

total points=100

Exam #138 Page: 1 Name: _____

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 0010 0110 1111 from binary to hexadecimal **26f**

b. (2 pts) Convert 1001 0111 1010 1110 from base 2 to base 16 **97ae**

c. (2 pts) Convert 6b24 from base 16 to base 2 **0110 1011 0010 0100**

d. (2 pts) Convert 1110 1111 0101 0011 from base 2 to base 16 **ef53**

e. (2 pts) Convert 54 from base 8 to base 2 **101 100**

f. (2 pts) Convert fcd6 from base 16 to binary **1111 1100 1101 0110**

g. (2 pts) Convert 183 from base 10 to binary **1011 0111**

h. (2 pts) Convert 0101 1000 from binary to base 10 **88**

i. (2 pts) Convert 74 from octal to base 2 **111 100**

j. (2 pts) Convert 55 from base 8 to binary **101 101**

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 lime apple
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][0]`? .

c. (2 pts) What is the value of `argv[2][0]`? a

d. (2 pts) What is the value of `argv[1][2]`? m

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node e;  
    double f;  
    int g;  
    char h;  
    Node *p;  
    double *q;  
    int *r;  
    char *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `r` `int *`
- b. (2 pts) `*q` `double`
- c. (2 pts) `p->next` `Node *`
- d. (2 pts) `&e` `Node *`
- e. (2 pts) `p->data` `int`
- f. (2 pts) `p->next->next` `Node *`
- g. (2 pts) `argc` `int`
- h. (2 pts) `argv[0]` `char *`
- i. (2 pts) `&p` `Node **`
- j. (2 pts) `e` `Node`
- k. (2 pts) `argv[1][2]` `char`

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 parameters 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 parameters 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;
}
```

Exam #138 Page: 8 Name: _____

End of Exam

total points=100

Exam #139 Page: 1 Name: _____

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 010 001 110 from binary to octal **216**

b. (2 pts) Convert 24 from octal to binary **010 100**

c. (2 pts) Convert 0101 1001 from binary to base 10 **89**

d. (2 pts) Convert 36 from base 8 to binary **011 110**

e. (2 pts) Convert 1001 0111 from base 2 to decimal **151**

f. (2 pts) Convert 150 from decimal to binary **1001 0110**

g. (2 pts) Convert 97 from decimal to binary **0110 0001**

h. (2 pts) Convert 1110 0000 from base 2 to base 10 **224**

i. (2 pts) Convert 1011 1011 1100 1111 from base 2 to hexadecimal **bpcf**

j. (2 pts) Convert 24 from octal to binary **010 100**

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 fig apple kiwi lime
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][2]`? g

c. (2 pts) What is the value of `argv[2][3]`? l

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int b;  
    double c;  
    Node d;  
    char e;  
    int *f;  
    double *g;  
    Node *h;  
    char *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `h->next` **Node ***
- b. (2 pts) `argv[0]` **char ***
- c. (2 pts) `*p` **char**
- d. (2 pts) `&e` **char ***
- e. (2 pts) `argv[1][2]` **char**
- f. (2 pts) `argc` **int**
- g. (2 pts) `h->data` **int**
- h. (2 pts) `&h` **Node ****
- i. (2 pts) `g` **double ***
- j. (2 pts) `h->next->next` **Node ***
- k. (2 pts) `d` **Node**

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 parameters 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 parameters 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;
}
```

Exam #139 Page: 8 Name: _____

End of Exam

total points=100

Exam #140 Page: 1 Name: _____

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 100 111 111 from binary to octal **477**

b. (2 pts) Convert 1111 0100 1010 0010 from binary to base 16 **f4a2**

c. (2 pts) Convert 254 from base 10 to binary **1111 1110**

d. (2 pts) Convert 54 from base 10 to binary **0011 0110**

e. (2 pts) Convert 7c3 from base 16 to base 2 **0111 1100 0011**

f. (2 pts) Convert 100 110 000 from binary to octal **460**

g. (2 pts) Convert 54 from octal to binary **101 100**

h. (2 pts) Convert 1000 0110 0111 0010 from base 2 to base 16 **8672**

i. (2 pts) Convert 1010 1110 0100 0011 from binary to base 16 **ae43**

j. (2 pts) Convert 0101 1101 1001 1011 from binary to base 16 **5d9b**

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 guava fig apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][2]`? a

c. (2 pts) What is the value of `argv[2][0]`? f

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int s;  
    Node t;  
    double w;  
    char x;  
    int *y;  
    Node *z;  
    double *a;  
    char *b;  
  
    return 0;  
}
```

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

a. (2 pts) `b` **char ***

b. (2 pts) `z->data` **int**

c. (2 pts) `t` **Node**

d. (2 pts) `z->next` **Node ***

e. (2 pts) `*a` **double**

f. (2 pts) `&z` **Node ****

g. (2 pts) `z->next->next` **Node ***

h. (2 pts) `&x` **char ***

i. (2 pts) `argc` **int**

j. (2 pts) `argv[0]` **char ***

k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #140 Page: 8 Name: _____

End of Exam

total points=100

Exam #141 Page: 1 Name: _____

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 e48a from hexadecimal to base 2 **1110 0100 1000 1010**

b. (2 pts) Convert 1010 1110 from binary to decimal **174**

c. (2 pts) Convert 236 from decimal to binary **1110 1100**

d. (2 pts) Convert 1100 0010 from base 2 to base 10 **194**

e. (2 pts) Convert 1110 1011 from base 2 to base 10 **235**

f. (2 pts) Convert 14 from octal to base 2 **001 100**

g. (2 pts) Convert 27 from base 8 to binary **010 111**

h. (2 pts) Convert 1110 from base 2 to base 10 **14**

i. (2 pts) Convert 36 from base 8 to binary **011 110**

j. (2 pts) Convert 1111 1001 1000 0101 from binary to base 16 **f985**

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 guava cherry
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][4]`? **n**

c. (2 pts) What is the value of `argv[2][2]`? **e**

d. (2 pts) What is the value of `argv[1][3]`? **v**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node p;  
    double q;  
    int r;  
    char s;  
    Node *t;  
    double *w;  
    int *x;  
    char *y;  
  
    return 0;  
}
```

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

- a. (2 pts) `x` **int ***
- b. (2 pts) `&r` **int ***
- c. (2 pts) `argc` **int**
- d. (2 pts) `p` **Node**
- e. (2 pts) `t->next` **Node ***
- f. (2 pts) `&y` **char ****
- g. (2 pts) `argv[1][2]` **char**
- h. (2 pts) `t->data` **int**
- i. (2 pts) `*t` **Node**
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `t->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #141 Page: 8 Name: _____

End of Exam

total points=100

Exam #142 Page: 1 Name: _____

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 0001 0101 from base 2 to base 10 **21**
- b. (2 pts) Convert 100 000 000 from base 2 to base 8 **400**
- c. (2 pts) Convert 0010 0010 1010 0101 from binary to base 16 **22a5**
- d. (2 pts) Convert 0001 1110 1111 1100 from base 2 to hexadecimal **1efc**
- e. (2 pts) Convert 010 000 010 from base 2 to base 8 **202**
- f. (2 pts) Convert 011 001 000 from base 2 to octal **310**
- g. (2 pts) Convert 95 from base 10 to base 2 **0101 1111**
- h. (2 pts) Convert 0111 0111 from binary to base 10 **119**
- i. (2 pts) Convert 6 from octal to base 2 **110**
- j. (2 pts) Convert 0010 0101 1100 0110 from binary to base 16 **25c6**

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 banana cherry
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[2][1]`? **h**

c. (2 pts) What is the value of `argv[0][5]`? **I**

d. (2 pts) What is the value of `argv[1][2]`? **n**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double q;  
    Node r;  
    int s;  
    char t;  
    double *w;  
    Node *x;  
    int *y;  
    char *z;  
  
    return 0;  
}
```

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

- a. (2 pts) `*z` **char**
- b. (2 pts) `x->next` **Node ***
- c. (2 pts) `argv[1][2]` **char**
- d. (2 pts) `x->next->next` **Node ***
- e. (2 pts) `argc` **int**
- f. (2 pts) `y` **int ***
- g. (2 pts) `&q` **double ***
- h. (2 pts) `r` **Node**
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `&w` **double ****
- k. (2 pts) `x->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #142 Page: 8 Name: _____

End of Exam

total points=100

Exam #143 Page: 1 Name: _____

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 0101 1010 1100 1011 from base 2 to hexadecimal **5acb**

b. (2 pts) Convert 0011 1010 from base 2 to decimal **58**

c. (2 pts) Convert 0001 0000 1000 0110 from binary to hexadecimal **1086**

d. (2 pts) Convert 170 from base 10 to base 2 **1010 1010**

e. (2 pts) Convert 36 from base 10 to binary **0010 0100**

f. (2 pts) Convert 77 from octal to binary **111 111**

g. (2 pts) Convert 9 from base 10 to base 2 **1001**

h. (2 pts) Convert 1111 1110 from binary to base 10 **254**

i. (2 pts) Convert 111 001 000 from base 2 to octal **710**

j. (2 pts) Convert 1100 0001 1011 0001 from base 2 to base 16 **c1b1**

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 mango guava grape apple
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][0]`? g

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

d. (2 pts) What is the value of `argv[0][3]`? u

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double y;  
    int z;  
    Node a;  
    char b;  
    double *c;  
    int *d;  
    Node *e;  
    char *f;  
  
    return 0;  
}
```

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

- a. (2 pts) `e->data` **int**
- b. (2 pts) `e->next` **Node ***
- c. (2 pts) `argv[0]` **char ***
- d. (2 pts) `&e` **Node ****
- e. (2 pts) `c` **double ***
- f. (2 pts) `e->next->next` **Node ***
- g. (2 pts) `b` **char**
- h. (2 pts) `argc` **int**
- i. (2 pts) `argv[1][2]` **char**
- j. (2 pts) `*d` **int**
- k. (2 pts) `&y` **double ***

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 parameters 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 parameters 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;
}
```

Exam #143 Page: 8 Name: _____

End of Exam

total points=100

Exam #144 Page: 1 Name: _____

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 0001 0111 0011 1110 from binary to base 16 **173e**

b. (2 pts) Convert 1001 1000 from binary to base 10 **152**

c. (2 pts) Convert 249 from base 10 to binary **1111 1001**

d. (2 pts) Convert 72 from base 8 to binary **111 010**

e. (2 pts) Convert 43 from decimal to base 2 **0010 1011**

f. (2 pts) Convert 1100 0011 1101 1011 from binary to base 16 **c3db**

g. (2 pts) Convert 1d2d from base 16 to binary **0001 1101 0010 1101**

h. (2 pts) Convert 1010 1111 0010 1000 from binary to hexadecimal **af28**

i. (2 pts) Convert 147 from base 10 to binary **1001 0011**

j. (2 pts) Convert 23 from base 8 to base 2 **010 011**

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 kiwi lime grape date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? i

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

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int w;  
    double x;  
    Node y;  
    char z;  
    int *a;  
    double *b;  
    Node *c;  
    char *d;  
  
    return 0;  
}
```

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

- a. (2 pts) `*b` **double**
- b. (2 pts) `argc` **int**
- c. (2 pts) `&z` **char ***
- d. (2 pts) `argv[1][2]` **char**
- e. (2 pts) `argv[0]` **char ***
- f. (2 pts) `c->next->next` **Node ***
- g. (2 pts) `c->next` **Node ***
- h. (2 pts) `y` **Node**
- i. (2 pts) `&a` **int ****
- j. (2 pts) `c->data` **int**
- k. (2 pts) `c` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #144 Page: 8 Name: _____

End of Exam

total points=100

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 010 111 000 from binary to base 8 **270**

b. (2 pts) Convert 0101 0010 0101 0101 from binary to hexadecimal **5255**

c. (2 pts) Convert 1110 0111 from base 2 to decimal **231**

d. (2 pts) Convert 751d from base 16 to base 2 **0111 0101 0001 1101**

e. (2 pts) Convert 1110 1101 0000 from binary to hexadecimal **ed0**

f. (2 pts) Convert 5db8 from hexadecimal to base 2 **0101 1101 1011 1000**

g. (2 pts) Convert 61 from octal to binary **110 001**

h. (2 pts) Convert 0011 0110 from binary to decimal **54**

i. (2 pts) Convert 010 111 011 from binary to base 8 **273**

j. (2 pts) Convert 72 from octal to binary **111 010**

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 apple kiwi fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][4]`? e

c. (2 pts) What is the value of `argv[0][3]`? u

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int q;  
    Node r;  
    double s;  
    char t;  
    int *w;  
    Node *x;  
    double *y;  
    char *z;  
  
    return 0;  
}
```

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

a. (2 pts) `x->next->next` **Node ***

b. (2 pts) `argv[1][2]` **char**

c. (2 pts) `&s` **double ***

d. (2 pts) `x->data` **int**

e. (2 pts) `&y` **double ****

f. (2 pts) `x->next` **Node ***

g. (2 pts) `argv[0]` **char ***

h. (2 pts) `argc` **int**

i. (2 pts) `x` **Node ***

j. (2 pts) `q` **int**

k. (2 pts) `*y` **double**

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 parameters 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 parameters 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;
}
```

Exam #145 Page: 8 Name: _____

End of Exam

total points=100

Exam #201 Page: 1 Name: _____

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 010 001 from binary to octal **21**

b. (2 pts) Convert 0101 0000 from binary to base 10 **80**

c. (2 pts) Convert 1010 1100 1101 1101 from base 2 to hexadecimal **acdd**

d. (2 pts) Convert 1000 1101 from base 2 to decimal **141**

e. (2 pts) Convert 48 from base 10 to base 2 **0011 0000**

f. (2 pts) Convert 0100 0001 1010 0111 from base 2 to hexadecimal **41a7**

g. (2 pts) Convert 0011 0001 from base 2 to decimal **49**

h. (2 pts) Convert 110 110 110 from binary to base 8 **666**

i. (2 pts) Convert 110 101 011 from base 2 to base 8 **653**

j. (2 pts) Convert 1000 0100 from binary to base 10 **132**

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 date banana mango
```

a. (2 pts) What is the value of `argc` in this case? **4**

b. (2 pts) What is the value of `argv[0][2]`? **r**

c. (2 pts) What is the value of `argv[2][0]`? **b**

d. (2 pts) What is the value of `argv[1][0]`? **d**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int r;  
    Node s;  
    double t;  
    char w;  
    int *x;  
    Node *y;  
    double *z;  
    char *a;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[1][2]` **char**

b. (2 pts) `&y` **Node ****

c. (2 pts) `y` **Node ***

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `y->next->next` **Node ***

f. (2 pts) `s` **Node**

g. (2 pts) `argc` **int**

h. (2 pts) `y->next` **Node ***

i. (2 pts) `y->data` **int**

j. (2 pts) `*x` **int**

k. (2 pts) `&t` **double ***

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 parameters 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 parameters 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;
}
```

Exam #201 Page: 8 Name: _____

End of Exam

total points=100

Exam #202 Page: 1 Name: _____

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 0011 1010 0100 0010 from base 2 to base 16 **3a42**

b. (2 pts) Convert 001 000 011 from base 2 to octal **103**

c. (2 pts) Convert 227 from base 10 to base 2 **1110 0011**

d. (2 pts) Convert 233 from decimal to binary **1110 1001**

e. (2 pts) Convert 1000 0110 from base 2 to base 10 **134**

f. (2 pts) Convert 011 101 000 from base 2 to base 8 **350**

g. (2 pts) Convert 0011 0100 0000 0010 from binary to hexadecimal **3402**

h. (2 pts) Convert 010 001 000 from base 2 to base 8 **210**

i. (2 pts) Convert 011 101 111 from binary to octal **357**

j. (2 pts) Convert 1011 0000 from binary to decimal **176**

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 guava cherry
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[2][4]`? **r**

c. (2 pts) What is the value of `argv[1][4]`? **a**

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double s;  
    Node t;  
    int w;  
    char x;  
    double *y;  
    Node *z;  
    int *a;  
    char *b;  
  
    return 0;  
}
```

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

- a. (2 pts) `y` **double ***
- b. (2 pts) `z->data` **int**
- c. (2 pts) `w` **int**
- d. (2 pts) `argv[1][2]` **char**
- e. (2 pts) `argc` **int**
- f. (2 pts) `z->next->next` **Node ***
- g. (2 pts) `&a` **int ****
- h. (2 pts) `z->next` **Node ***
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `&s` **double ***
- k. (2 pts) `*y` **double**

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 parameters 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 parameters 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;
}
```

Exam #202 Page: 8 Name: _____

End of Exam

total points=100

Exam #203 Page: 1 Name: _____

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 0111 1111 from binary to decimal **127**

b. (2 pts) Convert dc09 from hexadecimal to base 2 **1101 1100 0000 1001**

c. (2 pts) Convert 110 100 010 from binary to octal **642**

d. (2 pts) Convert 117 from decimal to base 2 **0111 0101**

e. (2 pts) Convert 0110 1010 0010 0001 from base 2 to base 16 **6a21**

f. (2 pts) Convert 3 from octal to base 2 **011**

g. (2 pts) Convert 1101 1101 1111 0011 from base 2 to hexadecimal **ddf3**

h. (2 pts) Convert cc17 from base 16 to base 2 **1100 1100 0001 0111**

i. (2 pts) Convert 010 000 011 from base 2 to base 8 **203**

j. (2 pts) Convert 0100 1100 from base 2 to decimal **76**

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 cherry lime apple mango
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][5]`? y

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

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int z;  
    double a;  
    Node b;  
    char c;  
    int *d;  
    double *e;  
    Node *f;  
    char *g;  
  
    return 0;  
}
```

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

- a. (2 pts) `f->data` **int**
- b. (2 pts) `&c` **char ***
- c. (2 pts) `f->next->next` **Node ***
- d. (2 pts) `z` **int**
- e. (2 pts) `*e` **double**
- f. (2 pts) `argc` **int**
- g. (2 pts) `f->next` **Node ***
- h. (2 pts) `&d` **int ****
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `g` **char ***
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #203 Page: 8 Name: _____

End of Exam

total points=100

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 1101 0111 from base 2 to decimal **215**

b. (2 pts) Convert 011 111 101 from binary to octal **375**

c. (2 pts) Convert 0111 0110 from base 2 to base 10 **118**

d. (2 pts) Convert 001 100 010 from base 2 to base 8 **142**

e. (2 pts) Convert 1101 1010 from base 2 to decimal **218**

f. (2 pts) Convert 1111 from binary to base 10 **15**

g. (2 pts) Convert 0010 1111 from binary to base 10 **47**

h. (2 pts) Convert 7289 from base 16 to binary **0111 0010 1000 1001**

i. (2 pts) Convert 001 101 000 from binary to octal **150**

j. (2 pts) Convert 010 110 000 from binary to base 8 **260**

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 apple kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][1]`? e

c. (2 pts) What is the value of `argv[0][4]`? n

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node q;  
    int r;  
    double s;  
    char t;  
    Node *w;  
    int *x;  
    double *y;  
    char *z;  
  
    return 0;  
}
```

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

- a. (2 pts) `&z` `char **`
- b. (2 pts) `argc` `int`
- c. (2 pts) `*x` `int`
- d. (2 pts) `w->next->next` `Node *`
- e. (2 pts) `argv[0]` `char *`
- f. (2 pts) `w->data` `int`
- g. (2 pts) `t` `char`
- h. (2 pts) `&t` `char *`
- i. (2 pts) `w->next` `Node *`
- j. (2 pts) `argv[1][2]` `char`
- k. (2 pts) `x` `int *`

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 parameters 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 parameters 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;
}
```

Exam #204 Page: 8 Name: _____

End of Exam

total points=100

Exam #205 Page: 1 Name: _____

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 0001 1100 0101 1101 from binary to hexadecimal **1c5d**

b. (2 pts) Convert 38fd from hexadecimal to base 2 **0011 1000 1111 1101**

c. (2 pts) Convert 31 from base 8 to base 2 **011 001**

d. (2 pts) Convert 1011 1101 0100 0111 from base 2 to hexadecimal **bd47**

e. (2 pts) Convert 189 from base 10 to base 2 **1011 1101**

f. (2 pts) Convert 169 from decimal to base 2 **1010 1001**

g. (2 pts) Convert 1101 1001 from base 2 to decimal **217**

h. (2 pts) Convert 111 110 100 from binary to octal **764**

i. (2 pts) Convert 77 from octal to base 2 **111 111**

j. (2 pts) Convert 111 101 000 from base 2 to octal **750**

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 date lemon lime
```

- a. (2 pts) What is the value of `argc` in this case? **4**
- b. (2 pts) What is the value of `argv[0][6]`? **t**
- c. (2 pts) What is the value of `argv[2][3]`? **o**
- d. (2 pts) What is the value of `argv[1][1]`? **a**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int g;  
    Node h;  
    double p;  
    char q;  
    int *r;  
    Node *s;  
    double *t;  
    char *w;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[0]` **char ***
- b. (2 pts) `p` **double**
- c. (2 pts) `s->next->next` **Node ***
- d. (2 pts) `r` **int ***
- e. (2 pts) `&s` **Node ****
- f. (2 pts) `&p` **double ***
- g. (2 pts) `argc` **int**
- h. (2 pts) `s->next` **Node ***
- i. (2 pts) `argv[1][2]` **char**
- j. (2 pts) `s->data` **int**
- k. (2 pts) `*t` **double**

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 parameters 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 parameters 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;
}
```

Exam #205 Page: 8 Name: _____

End of Exam

total points=100

Exam #206 Page: 1 Name: _____

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 77 from base 10 to base 2 **0100 1101**

b. (2 pts) Convert 10 from base 10 to binary **1010**

c. (2 pts) Convert 154 from decimal to binary **1001 1010**

d. (2 pts) Convert 0001 1001 from base 2 to base 10 **25**

e. (2 pts) Convert 1403 from base 16 to base 2 **0001 0100 0000 0011**

f. (2 pts) Convert db84 from base 16 to binary **1101 1011 1000 0100**

g. (2 pts) Convert 1101 1011 from binary to base 10 **219**

h. (2 pts) Convert 011 000 110 from base 2 to base 8 **306**

i. (2 pts) Convert 1001 1111 from binary to decimal **159**

j. (2 pts) Convert 0010 0000 from base 2 to base 10 **32**

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 mango date
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[2][1]`? **a**

c. (2 pts) What is the value of `argv[0][5]`? **I**

d. (2 pts) What is the value of `argv[1][4]`? **o**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double h;  
    Node p;  
    int q;  
    char r;  
    double *s;  
    Node *t;  
    int *w;  
    char *x;  
  
    return 0;  
}
```

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

- a. (2 pts) `t->data` **int**
- b. (2 pts) `&x` **char ****
- c. (2 pts) `t->next->next` **Node ***
- d. (2 pts) `argc` **int**
- e. (2 pts) `t->next` **Node ***
- f. (2 pts) `r` **char**
- g. (2 pts) `argv[1][2]` **char**
- h. (2 pts) `s` **double ***
- i. (2 pts) `&p` **Node ***
- j. (2 pts) `*w` **int**
- k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #206 Page: 8 Name: _____

End of Exam

total points=100

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 44 from base 8 to base 2 **100 100**

b. (2 pts) Convert 61 from base 8 to base 2 **110 001**

c. (2 pts) Convert 1000 1000 from binary to decimal **136**

d. (2 pts) Convert 51 from base 8 to binary **101 001**

e. (2 pts) Convert 75 from base 8 to binary **111 101**

f. (2 pts) Convert 117 from decimal to base 2 **0111 0101**

g. (2 pts) Convert 1000 0101 from base 2 to base 10 **133**

h. (2 pts) Convert ead9 from base 16 to base 2 **1110 1010 1101 1001**

i. (2 pts) Convert 011 010 011 from base 2 to base 8 **323**

j. (2 pts) Convert 101 111 000 from binary to base 8 **570**

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 date lime kiwi cherry
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? i

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

d. (2 pts) What is the value of `argv[0][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double e;  
    int f;  
    Node g;  
    char h;  
    double *p;  
    int *q;  
    Node *r;  
    char *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `&q` `int **`
- b. (2 pts) `&e` `double *`
- c. (2 pts) `*s` `char`
- d. (2 pts) `r->next->next` `Node *`
- e. (2 pts) `f` `int`
- f. (2 pts) `r->next` `Node *`
- g. (2 pts) `argv[1][2]` `char`
- h. (2 pts) `argc` `int`
- i. (2 pts) `s` `char *`
- j. (2 pts) `argv[0]` `char *`
- k. (2 pts) `r->data` `int`

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 parameters 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 parameters 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;
}
```

Exam #207 Page: 8 Name: _____

End of Exam

total points=100

Exam #208 Page: 1 Name: _____

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 0100 1111 0001 0001 from binary to hexadecimal **4f11**

b. (2 pts) Convert 10 from octal to binary **001 000**

c. (2 pts) Convert 34 from octal to base 2 **011 100**

d. (2 pts) Convert 227 from decimal to base 2 **1110 0011**

e. (2 pts) Convert 1111 1110 0101 1011 from base 2 to base 16 **fe5b**

f. (2 pts) Convert 0011 1011 from binary to base 10 **59**

g. (2 pts) Convert 1001 1001 from base 2 to decimal **153**

h. (2 pts) Convert 100 110 110 from base 2 to octal **466**

i. (2 pts) Convert 25 from base 10 to base 2 **0001 1001**

j. (2 pts) Convert 0100 1010 from binary to decimal **74**

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 guava mango date
```

a. (2 pts) What is the value of `argc` in this case? 4

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

c. (2 pts) What is the value of `argv[1][4]`? a

d. (2 pts) What is the value of `argv[0][3]`? u

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int s;  
    Node t;  
    double w;  
    char x;  
    int *y;  
    Node *z;  
    double *a;  
    char *b;  
  
    return 0;  
}
```

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

- a. (2 pts) `&b` `char **`
- b. (2 pts) `argc` `int`
- c. (2 pts) `z->next` `Node *`
- d. (2 pts) `y` `int *`
- e. (2 pts) `*b` `char`
- f. (2 pts) `argv[1][2]` `char`
- g. (2 pts) `&x` `char *`
- h. (2 pts) `z->data` `int`
- i. (2 pts) `z->next->next` `Node *`
- j. (2 pts) `s` `int`
- k. (2 pts) `argv[0]` `char *`

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 parameters 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 parameters 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;
}
```

Exam #208 Page: 8 Name: _____

End of Exam

total points=100

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 1001 0011 from binary to base 10 **147**

b. (2 pts) Convert 110 111 001 from binary to base 8 **671**

c. (2 pts) Convert 27 from base 8 to base 2 **010 111**

d. (2 pts) Convert 0110 1111 from binary to decimal **111**

e. (2 pts) Convert 70 from base 8 to binary **111 000**

f. (2 pts) Convert 110 101 001 from binary to octal **651**

g. (2 pts) Convert 67 from decimal to binary **0100 0011**

h. (2 pts) Convert 001 000 101 from base 2 to octal **105**

i. (2 pts) Convert 227 from decimal to binary **1110 0011**

j. (2 pts) Convert 71 from base 8 to binary **111 001**

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 kiwi fig
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[2][0]`? f

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

d. (2 pts) What is the value of `argv[1][2]`? w

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node h;  
    double p;  
    int q;  
    char r;  
    Node *s;  
    double *t;  
    int *w;  
    char *x;  
  
    return 0;  
}
```

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

- a. (2 pts) `x` **char ***
- b. (2 pts) `argv[1][2]` **char**
- c. (2 pts) `&t` **double ****
- d. (2 pts) `argv[0]` **char ***
- e. (2 pts) `s->next->next` **Node ***
- f. (2 pts) `q` **int**
- g. (2 pts) `s->next` **Node ***
- h. (2 pts) `argc` **int**
- i. (2 pts) `*t` **double**
- j. (2 pts) `&r` **char ***
- k. (2 pts) `s->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #209 Page: 8 Name: _____

End of Exam

total points=100

Exam #210 Page: 1 Name: _____

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 110 001 010 from base 2 to octal **612**

b. (2 pts) Convert 53 from base 8 to binary **101 011**

c. (2 pts) Convert 9633 from base 16 to base 2 **1001 0110 0011 0011**

d. (2 pts) Convert 203 from decimal to base 2 **1100 1011**

e. (2 pts) Convert 001 101 111 from base 2 to octal **157**

f. (2 pts) Convert 75a from base 16 to base 2 **0111 0101 1010**

g. (2 pts) Convert 0100 0101 from base 2 to base 10 **69**

h. (2 pts) Convert 1000 1011 from binary to decimal **139**

i. (2 pts) Convert 133 from base 10 to base 2 **1000 0101**

j. (2 pts) Convert 18 from decimal to binary **0001 0010**

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 banana fig
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[2][1]`? **i**

c. (2 pts) What is the value of `argv[1][0]`? **b**

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double a;  
    Node b;  
    int c;  
    char d;  
    double *e;  
    Node *f;  
    int *g;  
    char *h;  
  
    return 0;  
}
```

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

- a. (2 pts) `&g` `int **`
- b. (2 pts) `f->next` `Node *`
- c. (2 pts) `argv[0]` `char *`
- d. (2 pts) `argc` `int`
- e. (2 pts) `g` `int *`
- f. (2 pts) `*e` `double`
- g. (2 pts) `&b` `Node *`
- h. (2 pts) `argv[1][2]` `char`
- i. (2 pts) `d` `char`
- j. (2 pts) `f->data` `int`
- k. (2 pts) `f->next->next` `Node *`

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 parameters 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 parameters 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;
}
```

Exam #210 Page: 8 Name: _____

End of Exam

total points=100

Exam #211 Page: 1 Name: _____

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 10 from decimal to binary **1010**

b. (2 pts) Convert 104 from base 10 to base 2 **0110 1000**

c. (2 pts) Convert 1000 0100 from binary to base 10 **132**

d. (2 pts) Convert 87 from decimal to binary **0101 0111**

e. (2 pts) Convert 27 from base 10 to binary **0001 1011**

f. (2 pts) Convert 161 from base 10 to base 2 **1010 0001**

g. (2 pts) Convert 1110 1111 from binary to base 10 **239**

h. (2 pts) Convert 138f from hexadecimal to binary **0001 0011 1000 1111**

i. (2 pts) Convert 4f6c from hexadecimal to binary **0100 1111 0110 1100**

j. (2 pts) Convert 174 from base 10 to binary **1010 1110**

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 guava fig kiwi cherry
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][0]`? g

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

d. (2 pts) What is the value of `argv[0][6]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int x;  
    double y;  
    Node z;  
    char a;  
    int *b;  
    double *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

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

a. (2 pts) `d->next->next` **Node ***

b. (2 pts) `&x` **int ***

c. (2 pts) `y` **double**

d. (2 pts) `c` **double ***

e. (2 pts) `argv[0]` **char ***

f. (2 pts) `argc` **int**

g. (2 pts) `d->data` **int**

h. (2 pts) `*d` **Node**

i. (2 pts) `argv[1][2]` **char**

j. (2 pts) `d->next` **Node ***

k. (2 pts) `&b` **int ****

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 parameters 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 parameters 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;
}
```

Exam #211 Page: 8 Name: _____

End of Exam

total points=100

Exam #212 Page: 1 Name: _____

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 98 from base 10 to binary **0110 0010**

b. (2 pts) Convert 1011 from base 2 to decimal **11**

c. (2 pts) Convert 41 from base 10 to binary **0010 1001**

d. (2 pts) Convert 100 110 from base 2 to octal **46**

e. (2 pts) Convert 1000 1011 1000 1001 from base 2 to hexadecimal **8b89**

f. (2 pts) Convert 1010 0010 1000 1000 from binary to base 16 **a288**

g. (2 pts) Convert 65 from base 10 to base 2 **0100 0001**

h. (2 pts) Convert 101 110 100 from binary to base 8 **564**

i. (2 pts) Convert 0100 0001 from base 2 to base 10 **65**

j. (2 pts) Convert 56 from base 8 to base 2 **101 110**

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 cherry banana lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][2]`? e

c. (2 pts) What is the value of `argv[0][6]`? t

d. (2 pts) What is the value of `argv[2][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node h;  
    int p;  
    double q;  
    char r;  
    Node *s;  
    int *t;  
    double *w;  
    char *x;  
  
    return 0;  
}
```

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

- a. (2 pts) `s` **Node ***
- b. (2 pts) `argv[1][2]` **char**
- c. (2 pts) `s->data` **int**
- d. (2 pts) `argc` **int**
- e. (2 pts) `p` **int**
- f. (2 pts) `s->next->next` **Node ***
- g. (2 pts) `argv[0]` **char ***
- h. (2 pts) `&x` **char ****
- i. (2 pts) `&h` **Node ***
- j. (2 pts) `*t` **int**
- k. (2 pts) `s->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #212 Page: 8 Name: _____

End of Exam

total points=100

Exam #213 Page: 1 Name: _____

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 51 from octal to binary **101 001**

b. (2 pts) Convert 197 from decimal to base 2 **1100 0101**

c. (2 pts) Convert 101 110 from binary to octal **56**

d. (2 pts) Convert 1001 1111 0101 0011 from base 2 to base 16 **9f53**

e. (2 pts) Convert 011 011 101 from base 2 to octal **335**

f. (2 pts) Convert 17 from octal to base 2 **001 111**

g. (2 pts) Convert 1110 1011 0000 0111 from binary to base 16 **eb07**

h. (2 pts) Convert 0100 0001 from binary to base 10 **65**

i. (2 pts) Convert 11 from decimal to binary **1011**

j. (2 pts) Convert 0101 0110 0010 0110 from binary to base 16 **5626**

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 fig grape
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][5]`? **I**

c. (2 pts) What is the value of `argv[2][0]`? **g**

d. (2 pts) What is the value of `argv[1][2]`? **g**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node e;  
    double f;  
    int g;  
    char h;  
    Node *p;  
    double *q;  
    int *r;  
    char *s;  
  
    return 0;  
}
```

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

- a. (2 pts) p->next **Node ***
- b. (2 pts) &h **char ***
- c. (2 pts) p->data **int**
- d. (2 pts) argc **int**
- e. (2 pts) p->next->next **Node ***
- f. (2 pts) argv[0] **char ***
- g. (2 pts) &q **double ****
- h. (2 pts) argv[1][2] **char**
- i. (2 pts) s **char ***
- j. (2 pts) *s **char**
- k. (2 pts) h **char**

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 parameters 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 parameters 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;
}
```

Exam #213 Page: 8 Name: _____

End of Exam

total points=100

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 d8d0 from base 16 to base 2 **1101 1000 1101 0000**

b. (2 pts) Convert 96b0 from hexadecimal to binary **1001 0110 1011 0000**

c. (2 pts) Convert 0100 1101 1011 0101 from binary to base 16 **4db5**

d. (2 pts) Convert 1111 1011 from base 2 to base 10 **251**

e. (2 pts) Convert 1100 0100 from base 2 to decimal **196**

f. (2 pts) Convert 33 from base 8 to binary **011 011**

g. (2 pts) Convert 237 from base 10 to binary **1110 1101**

h. (2 pts) Convert aaa0 from base 16 to binary **1010 1010 1010 0000**

i. (2 pts) Convert 1010 1101 from base 2 to base 10 **173**

j. (2 pts) Convert 40 from base 8 to binary **100 000**

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 date lime guava kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

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

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

d. (2 pts) What is the value of `argv[1][2]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double e;  
    int f;  
    Node g;  
    char h;  
    double *p;  
    int *q;  
    Node *r;  
    char *s;  
  
    return 0;  
}
```

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

a. (2 pts) `argc`

int

b. (2 pts) `e`

double

c. (2 pts) `r->next`

Node *

d. (2 pts) `*r`

Node

e. (2 pts) `argv[1][2]`

char

f. (2 pts) `argv[0]`

char *

g. (2 pts) `r->data`

int

h. (2 pts) `&g`

Node *

i. (2 pts) `&s`

char **

j. (2 pts) `r->next->next`

Node *

k. (2 pts) `r`

Node *

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 parameters 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 parameters 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;
}
```

Exam #214 Page: 8 Name: _____

End of Exam

total points=100

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 111 011 from base 2 to octal **73**

b. (2 pts) Convert 0101 0000 1101 1001 from base 2 to base 16 **50d9**

c. (2 pts) Convert 0011 1011 1001 0101 from binary to hexadecimal **3b95**

d. (2 pts) Convert 41 from base 8 to binary **100 001**

e. (2 pts) Convert a83e from base 16 to base 2 **1010 1000 0011 1110**

f. (2 pts) Convert 1000 1010 1011 from base 2 to base 16 **8ab**

g. (2 pts) Convert 151 from decimal to binary **1001 0111**

h. (2 pts) Convert 0011 0010 0101 0000 from binary to hexadecimal **3250**

i. (2 pts) Convert 119 from decimal to binary **0111 0111**

j. (2 pts) Convert 7 from octal to base 2 **111**

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 mango banana guava apple
```

a. (2 pts) What is the value of `argc` in this case? 5

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

c. (2 pts) What is the value of `argv[1][0]`? m

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double r;  
    int s;  
    Node t;  
    char w;  
    double *x;  
    int *y;  
    Node *z;  
    char *a;  
  
    return 0;  
}
```

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

a. (2 pts) `y`

`int *`

b. (2 pts) `z->data`

`int`

c. (2 pts) `argv[1][2]`

`char`

d. (2 pts) `argc`

`int`

e. (2 pts) `t`

`Node`

f. (2 pts) `&s`

`int *`

g. (2 pts) `z->next`

`Node *`

h. (2 pts) `argv[0]`

`char *`

i. (2 pts) `z->next->next`

`Node *`

j. (2 pts) `&y`

`int **`

k. (2 pts) `*x`

`double`

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 parameters 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 parameters 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;
}
```

Exam #215 Page: 8 Name: _____

End of Exam

total points=100

Exam #216 Page: 1 Name: _____

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 011 101 100 from base 2 to octal **354**

b. (2 pts) Convert 243 from base 10 to base 2 **1111 0011**

c. (2 pts) Convert e0f7 from hexadecimal to base 2 **1110 0000 1111 0111**

d. (2 pts) Convert 4311 from base 16 to binary **0100 0011 0001 0001**

e. (2 pts) Convert 6 from base 8 to binary **110**

f. (2 pts) Convert 9 from base 10 to base 2 **1001**

g. (2 pts) Convert 1110 1000 1101 1001 from base 2 to hexadecimal **e8d9**

h. (2 pts) Convert 110 110 001 from binary to base 8 **661**

i. (2 pts) Convert 106 from base 10 to base 2 **0110 1010**

j. (2 pts) Convert 0010 1001 1110 0101 from binary to hexadecimal **29e5**

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 banana lemon grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][5]`? a

c. (2 pts) What is the value of `argv[0][4]`? n

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

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

- a. (2 pts) `&d` **Node ***
- b. (2 pts) `f` **double**
- c. (2 pts) `*r` **char**
- d. (2 pts) `&h` **Node ****
- e. (2 pts) `h->next->next` **Node ***
- f. (2 pts) `h->next` **Node ***
- g. (2 pts) `h->data` **int**
- h. (2 pts) `argc` **int**
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `argv[1][2]` **char**
- k. (2 pts) `r` **char ***

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 parameters 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 parameters 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;
}
```

Exam #216 Page: 8 Name: _____

End of Exam

total points=100

Exam #217 Page: 1 Name: _____

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 baeb from base 16 to binary **1011 1010 1110 1011**

b. (2 pts) Convert 53 from base 8 to binary **101 011**

c. (2 pts) Convert ced7 from hexadecimal to base 2 **1100 1110 1101 0111**

d. (2 pts) Convert cefb from base 16 to base 2 **1100 1110 1111 1011**

e. (2 pts) Convert fc02 from base 16 to base 2 **1111 1100 0000 0010**

f. (2 pts) Convert 1010 0011 from binary to base 10 **163**

g. (2 pts) Convert 44 from octal to binary **100 100**

h. (2 pts) Convert 6072 from hexadecimal to base 2 **0110 0000 0111 0010**

i. (2 pts) Convert 0011 0100 from binary to decimal **52**

j. (2 pts) Convert 1100 0101 1100 1111 from binary to hexadecimal **c5cf**

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 guava lemon
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][3]`? **u**

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double a;  
    Node b;  
    int c;  
    char d;  
    double *e;  
    Node *f;  
    int *g;  
    char *h;  
  
    return 0;  
}
```

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

- a. (2 pts) `f->next` **Node ***
- b. (2 pts) `argv[1][2]` **char**
- c. (2 pts) `argv[0]` **char ***
- d. (2 pts) `argc` **int**
- e. (2 pts) `g` **int ***
- f. (2 pts) `a` **double**
- g. (2 pts) `&g` **int ****
- h. (2 pts) `&a` **double ***
- i. (2 pts) `f->next->next` **Node ***
- j. (2 pts) `f->data` **int**
- k. (2 pts) `*f` **Node**

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 parameters 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 parameters 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;
}
```

Exam #217 Page: 8 Name: _____

End of Exam

total points=100

Exam #218 Page: 1 Name: _____

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 1110 1100 from binary to base 10 **236**

b. (2 pts) Convert 0111 1111 0101 0011 from binary to base 16 **7f53**

c. (2 pts) Convert 001 010 from binary to octal **12**

d. (2 pts) Convert 0010 1011 0011 1101 from base 2 to hexadecimal **2b3d**

e. (2 pts) Convert 82 from base 10 to base 2 **0101 0010**

f. (2 pts) Convert 1101 0110 0100 0010 from binary to hexadecimal **d642**

g. (2 pts) Convert 1001 0100 1111 1000 from base 2 to base 16 **94f8**

h. (2 pts) Convert c961 from base 16 to base 2 **1100 1001 0110 0001**

i. (2 pts) Convert d60e from base 16 to base 2 **1101 0110 0000 1110**

j. (2 pts) Convert 1111 0010 0001 0000 from binary to base 16 **f210**

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 cherry grape lemon fig
```

a. (2 pts) What is the value of `argc` in this case? 5

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

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

d. (2 pts) What is the value of `argv[1][0]`? c

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int b;  
    double c;  
    Node d;  
    char e;  
    int *f;  
    double *g;  
    Node *h;  
    char *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `g` **double ***
- b. (2 pts) `&d` **Node ***
- c. (2 pts) `c` **double**
- d. (2 pts) `argc` **int**
- e. (2 pts) `h->next->next` **Node ***
- f. (2 pts) `argv[1][2]` **char**
- g. (2 pts) `h->data` **int**
- h. (2 pts) `*f` **int**
- i. (2 pts) `h->next` **Node ***
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `&f` **int ****

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 parameters 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 parameters 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;
}
```

Exam #218 Page: 8 Name: _____

End of Exam

total points=100

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 0011 0001 0010 1100 from binary to base 16 **312c**

b. (2 pts) Convert 0011 1001 from base 2 to decimal **57**

c. (2 pts) Convert 74 from base 8 to binary **111 100**

d. (2 pts) Convert 55 from base 8 to base 2 **101 101**

e. (2 pts) Convert 356b from base 16 to base 2 **0011 0101 0110 1011**

f. (2 pts) Convert 701e from base 16 to binary **0111 0000 0001 1110**

g. (2 pts) Convert 0011 1110 1110 1001 from binary to hexadecimal **3ee9**

h. (2 pts) Convert 0101 0001 0001 0010 from base 2 to base 16 **5112**

i. (2 pts) Convert 1010 0000 from binary to decimal **160**

j. (2 pts) Convert 1000 1101 1111 1010 from base 2 to base 16 **8dfa**

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 mango grape lemon date
```

a. (2 pts) What is the value of `argc` in this case? 5

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

c. (2 pts) What is the value of `argv[1][3]`? g

d. (2 pts) What is the value of `argv[0][6]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int x;  
    double y;  
    Node z;  
    char a;  
    int *b;  
    double *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

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

- a. (2 pts) `&y` **double ***
- b. (2 pts) `x` **int**
- c. (2 pts) `d->next->next` **Node ***
- d. (2 pts) `argc` **int**
- e. (2 pts) `argv[0]` **char ***
- f. (2 pts) `*d` **Node**
- g. (2 pts) `&d` **Node ****
- h. (2 pts) `b` **int ***
- i. (2 pts) `d->data` **int**
- j. (2 pts) `argv[1][2]` **char**
- k. (2 pts) `d->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #219 Page: 8 Name: _____

End of Exam

total points=100

Exam #220 Page: 1 Name: _____

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 1000 1001 1000 1001 from binary to base 16 **8989**

b. (2 pts) Convert 1101 1100 0100 1000 from base 2 to base 16 **dc48**

c. (2 pts) Convert 100 110 000 from binary to base 8 **460**

d. (2 pts) Convert 114 from base 10 to base 2 **0111 0010**

e. (2 pts) Convert 165 from base 10 to base 2 **1010 0101**

f. (2 pts) Convert 34 from base 8 to binary **011 100**

g. (2 pts) Convert 1001 0000 from base 2 to base 10 **144**

h. (2 pts) Convert 1111 0111 from base 2 to base 10 **247**

i. (2 pts) Convert 1001 0010 from base 2 to decimal **146**

j. (2 pts) Convert 998e from base 16 to binary **1001 1001 1000 1110**

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 date apple mango
```

a. (2 pts) What is the value of `argc` in this case? **4**

b. (2 pts) What is the value of `argv[1][0]`? **d**

c. (2 pts) What is the value of `argv[2][4]`? **e**

d. (2 pts) What is the value of `argv[0][4]`? **n**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int h;  
    Node p;  
    double q;  
    char r;  
    int *s;  
    Node *t;  
    double *w;  
    char *x;  
  
    return 0;  
}
```

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

a. (2 pts) t->next->next **Node ***

b. (2 pts) argv[1][2] **char**

c. (2 pts) &t **Node ****

d. (2 pts) argv[0] **char ***

e. (2 pts) *t **Node**

f. (2 pts) t->data **int**

g. (2 pts) argc **int**

h. (2 pts) t->next **Node ***

i. (2 pts) r **char**

j. (2 pts) x **char ***

k. (2 pts) &p **Node ***

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 parameters 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 parameters 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;
}
```

Exam #220 Page: 8 Name: _____

End of Exam

total points=100

Exam #221 Page: 1 Name: _____

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 110 011 100 from binary to octal **634**

b. (2 pts) Convert 1001 0110 from base 2 to decimal **150**

c. (2 pts) Convert 1000 0110 0101 1001 from base 2 to hexadecimal **8659**

d. (2 pts) Convert 1111 1110 from binary to decimal **254**

e. (2 pts) Convert 1000 1001 0010 1111 from base 2 to base 16 **892f**

f. (2 pts) Convert 010 110 from base 2 to base 8 **26**

g. (2 pts) Convert 0011 1010 from binary to base 10 **58**

h. (2 pts) Convert 7f34 from hexadecimal to base 2 **0111 1111 0011 0100**

i. (2 pts) Convert 010 111 001 from base 2 to base 8 **271**

j. (2 pts) Convert 3579 from base 16 to binary **0011 0101 0111 1001**

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 grape banana
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][2]`? r

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

d. (2 pts) What is the value of `argv[1][3]`? p

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node w;  
    double x;  
    int y;  
    char z;  
    Node *a;  
    double *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

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

a. (2 pts) `argc`

`int`

b. (2 pts) `a->next`

`Node *`

c. (2 pts) `a->data`

`int`

d. (2 pts) `c`

`int *`

e. (2 pts) `a->next->next`

`Node *`

f. (2 pts) `argv[1][2]`

`char`

g. (2 pts) `&w`

`Node *`

h. (2 pts) `*d`

`char`

i. (2 pts) `x`

`double`

j. (2 pts) `&d`

`char **`

k. (2 pts) `argv[0]`

`char *`

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 parameters 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 parameters 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;
}
```

Exam #221 Page: 8 Name: _____

End of Exam

total points=100

Exam #222 Page: 1 Name: _____

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 77 from base 8 to binary **111 111**

b. (2 pts) Convert 011 001 111 from binary to base 8 **317**

c. (2 pts) Convert 1011 1100 from base 2 to base 10 **188**

d. (2 pts) Convert 90 from base 10 to binary **0101 1010**

e. (2 pts) Convert 67 from base 8 to base 2 **110 111**

f. (2 pts) Convert 001 111 011 from binary to octal **173**

g. (2 pts) Convert 17 from base 8 to binary **001 111**

h. (2 pts) Convert e823 from hexadecimal to base 2 **1110 1000 0010 0011**

i. (2 pts) Convert 254 from base 10 to binary **1111 1110**

j. (2 pts) Convert 0110 0001 1011 1010 from base 2 to hexadecimal **61ba**

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 fig mango kiwi grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][0]`? f

c. (2 pts) What is the value of `argv[2][4]`? o

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double x;  
    int y;  
    Node z;  
    char a;  
    double *b;  
    int *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

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

a. (2 pts) `&a` **char ***

b. (2 pts) `d->next` **Node ***

c. (2 pts) `c` **int ***

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `*d` **Node**

f. (2 pts) `d->data` **int**

g. (2 pts) `argc` **int**

h. (2 pts) `d->next->next` **Node ***

i. (2 pts) `&b` **double ****

j. (2 pts) `z` **Node**

k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #222 Page: 8 Name: _____

End of Exam

total points=100

Exam #223 Page: 1 Name: _____

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 0100 0100 from binary to base 10 **68**
- b. (2 pts) Convert 221f from hexadecimal to binary **0010 0010 0001 1111**
- c. (2 pts) Convert 1010 1010 from binary to base 10 **170**
- d. (2 pts) Convert 230 from base 10 to binary **1110 0110**
- e. (2 pts) Convert 110 000 101 from base 2 to octal **605**
- f. (2 pts) Convert 110 101 111 from base 2 to octal **657**
- g. (2 pts) Convert 71 from octal to base 2 **111 001**
- h. (2 pts) Convert 011 011 111 from base 2 to base 8 **337**
- i. (2 pts) Convert 1100 1000 from binary to base 10 **200**
- j. (2 pts) Convert fda4 from hexadecimal to binary **1111 1101 1010 0100**

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 kiwi lime mango lemon
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][0]`? l

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int s;  
    double t;  
    Node w;  
    char x;  
    int *y;  
    double *z;  
    Node *a;  
    char *b;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[0]` **char ***

b. (2 pts) `argv[1][2]` **char**

c. (2 pts) `argc` **int**

d. (2 pts) `a->data` **int**

e. (2 pts) `*y` **int**

f. (2 pts) `a->next` **Node ***

g. (2 pts) `&w` **Node ***

h. (2 pts) `&a` **Node ****

i. (2 pts) `y` **int ***

j. (2 pts) `t` **double**

k. (2 pts) `a->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #223 Page: 8 Name: _____

End of Exam

total points=100

Exam #224 Page: 1 Name: _____

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 56 from decimal to binary **0011 1000**

b. (2 pts) Convert 2 from octal to binary **010**

c. (2 pts) Convert c06 from hexadecimal to base 2 **1100 0000 0110**

d. (2 pts) Convert 111 111 from base 2 to octal **77**

e. (2 pts) Convert 100 111 000 from base 2 to base 8 **470**

f. (2 pts) Convert 20 from base 10 to base 2 **0001 0100**

g. (2 pts) Convert 0111 0110 0111 0010 from binary to base 16 **7672**

h. (2 pts) Convert c51 from base 16 to binary **1100 0101 0001**

i. (2 pts) Convert 77 from octal to base 2 **111 111**

j. (2 pts) Convert 1000 0110 from base 2 to base 10 **134**

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 fig lemon cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][2]`? g

c. (2 pts) What is the value of `argv[0][6]`? t

d. (2 pts) What is the value of `argv[2][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int z;  
    Node a;  
    double b;  
    char c;  
    int *d;  
    Node *e;  
    double *f;  
    char *g;  
  
    return 0;  
}
```

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

- a. (2 pts) `&g` **char ****
- b. (2 pts) `a` **Node**
- c. (2 pts) `e->next->next` **Node ***
- d. (2 pts) `*g` **char**
- e. (2 pts) `argv[1][2]` **char**
- f. (2 pts) `e->data` **int**
- g. (2 pts) `e->next` **Node ***
- h. (2 pts) `argv[0]` **char ***
- i. (2 pts) `&b` **double ***
- j. (2 pts) `argc` **int**
- k. (2 pts) `d` **int ***

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 parameters 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 parameters 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;
}
```

Exam #224 Page: 8 Name: _____

End of Exam

total points=100

Exam #225 Page: 1 Name: _____

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 37 from octal to binary **011 111**

b. (2 pts) Convert 110 001 000 from base 2 to base 8 **610**

c. (2 pts) Convert f9e6 from base 16 to base 2 **1111 1001 1110 0110**

d. (2 pts) Convert 1010 1011 from base 2 to base 10 **171**

e. (2 pts) Convert 127 from base 10 to binary **0111 1111**

f. (2 pts) Convert 1010 1110 from base 2 to decimal **174**

g. (2 pts) Convert 0010 0000 0110 0011 from base 2 to base 16 **2063**

h. (2 pts) Convert 9401 from hexadecimal to binary **1001 0100 0000 0001**

i. (2 pts) Convert 1100 0111 1111 0000 from base 2 to hexadecimal **c7f0**

j. (2 pts) Convert 001 000 100 from binary to octal **104**

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 cherry kiwi
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][3]`? **u**

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

d. (2 pts) What is the value of `argv[1][0]`? **c**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node w;  
    double x;  
    int y;  
    char z;  
    Node *a;  
    double *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

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

- a. (2 pts) `*a` **Node**
- b. (2 pts) `argv[0]` **char ***
- c. (2 pts) `a->next` **Node ***
- d. (2 pts) `a->data` **int**
- e. (2 pts) `&x` **double ***
- f. (2 pts) `argc` **int**
- g. (2 pts) `z` **char**
- h. (2 pts) `&b` **double ****
- i. (2 pts) `d` **char ***
- j. (2 pts) `a->next->next` **Node ***
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #225 Page: 8 Name: _____

End of Exam

total points=100

Exam #226 Page: 1 Name: _____

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 af31 from base 16 to base 2 **1010 1111 0011 0001**

b. (2 pts) Convert 1001 0101 from binary to decimal **149**

c. (2 pts) Convert 0011 0000 from base 2 to decimal **48**

d. (2 pts) Convert 7 from decimal to binary **0111**

e. (2 pts) Convert d5e6 from base 16 to binary **1101 0101 1110 0110**

f. (2 pts) Convert 1110 0000 1100 0110 from binary to base 16 **e0c6**

g. (2 pts) Convert 0010 0010 1001 0001 from binary to hexadecimal **2291**

h. (2 pts) Convert 77 from octal to binary **111 111**

i. (2 pts) Convert 32 from octal to base 2 **011 010**

j. (2 pts) Convert 0100 1110 from base 2 to base 10 **78**

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 cherry guava
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[1][3]`? r

c. (2 pts) What is the value of `argv[2][4]`? a

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double w;  
    Node x;  
    int y;  
    char z;  
    double *a;  
    Node *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

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

- a. (2 pts) `c` **int ***
- b. (2 pts) `argv[1][2]` **char**
- c. (2 pts) `w` **double**
- d. (2 pts) `&z` **char ***
- e. (2 pts) `b->data` **int**
- f. (2 pts) `b->next->next` **Node ***
- g. (2 pts) `&c` **int ****
- h. (2 pts) `b->next` **Node ***
- i. (2 pts) `argc` **int**
- j. (2 pts) `*a` **double**
- k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #226 Page: 8 Name: _____

End of Exam

total points=100

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 111 101 000 from binary to base 8 **750**

b. (2 pts) Convert 79 from decimal to base 2 **0100 1111**

c. (2 pts) Convert 7 from base 8 to binary **111**

d. (2 pts) Convert 147 from base 10 to base 2 **1001 0011**

e. (2 pts) Convert 56 from base 8 to binary **101 110**

f. (2 pts) Convert 7aa2 from hexadecimal to base 2 **0111 1010 1010 0010**

g. (2 pts) Convert 1100 1100 1000 0001 from base 2 to hexadecimal **cc81**

h. (2 pts) Convert 100 001 001 from base 2 to octal **411**

i. (2 pts) Convert 0011 0011 1100 0010 from base 2 to hexadecimal **33c2**

j. (2 pts) Convert 1110 1010 from binary to base 10 **234**

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 fig guava grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? i

c. (2 pts) What is the value of `argv[1][4]`? n

d. (2 pts) What is the value of `argv[0][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int r;  
    double s;  
    Node t;  
    char w;  
    int *x;  
    double *y;  
    Node *z;  
    char *a;  
  
    return 0;  
}
```

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

- a. (2 pts) `y` **double ***
- b. (2 pts) `&x` **int ****
- c. (2 pts) `z->next->next` **Node ***
- d. (2 pts) `argv[1][2]` **char**
- e. (2 pts) `argc` **int**
- f. (2 pts) `z->next` **Node ***
- g. (2 pts) `w` **char**
- h. (2 pts) `&t` **Node ***
- i. (2 pts) `*y` **double**
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `z->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #227 Page: 8 Name: _____

End of Exam

total points=100

Exam #228 Page: 1 Name: _____

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 010 011 000 from base 2 to base 8 **230**

b. (2 pts) Convert 1111 0010 from binary to decimal **242**

c. (2 pts) Convert 110 000 111 from base 2 to base 8 **607**

d. (2 pts) Convert 010 011 110 from base 2 to octal **236**

e. (2 pts) Convert 0010 1001 from base 2 to base 10 **41**

f. (2 pts) Convert 36 from base 8 to base 2 **011 110**

g. (2 pts) Convert 111 100 from base 2 to octal **74**

h. (2 pts) Convert 2b12 from base 16 to binary **0010 1011 0001 0010**

i. (2 pts) Convert 0010 0110 0011 0101 from binary to hexadecimal **2635**

j. (2 pts) Convert 111 101 100 from binary to octal **754**

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 lime apple grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][0]`? l

c. (2 pts) What is the value of `argv[0][4]`? n

d. (2 pts) What is the value of `argv[2][2]`? p

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

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

a. (2 pts) `h->next->next` **Node ***

b. (2 pts) `h->data` **int**

c. (2 pts) `h->next` **Node ***

d. (2 pts) `*h` **Node**

e. (2 pts) `&r` **char ****

f. (2 pts) `argv[1][2]` **char**

g. (2 pts) `&f` **double ***

h. (2 pts) `r` **char ***

i. (2 pts) `argv[0]` **char ***

j. (2 pts) `f` **double**

k. (2 pts) `argc` **int**

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 parameters 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 parameters 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;
}
```

Exam #228 Page: 8 Name: _____

End of Exam

total points=100

Exam #229 Page: 1 Name: _____

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 145 from decimal to binary **1001 0001**

b. (2 pts) Convert 1010 1100 1101 0000 from base 2 to base 16 **acd0**

c. (2 pts) Convert 1011 0001 0110 1000 from base 2 to hexadecimal **b168**

d. (2 pts) Convert 1101 1011 0011 1100 from base 2 to hexadecimal **db3c**

e. (2 pts) Convert 3 from base 8 to binary **011**

f. (2 pts) Convert 0001 0101 1101 0000 from base 2 to base 16 **15d0**

g. (2 pts) Convert 110 010 000 from binary to octal **620**

h. (2 pts) Convert 54 from base 8 to base 2 **101 100**

i. (2 pts) Convert 74 from base 8 to base 2 **111 100**

j. (2 pts) Convert 100 100 100 from binary to octal **444**

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 lime date
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][1]`? /

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

d. (2 pts) What is the value of `argv[1][0]`? l

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node q;  
    double r;  
    int s;  
    char t;  
    Node *w;  
    double *x;  
    int *y;  
    char *z;  
  
    return 0;  
}
```

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

a. (2 pts) `y`

`int *`

b. (2 pts) `*y`

`int`

c. (2 pts) `argc`

`int`

d. (2 pts) `q`

`Node`

e. (2 pts) `&x`

`double **`

f. (2 pts) `w->data`

`int`

g. (2 pts) `argv[1][2]`

`char`

h. (2 pts) `w->next->next`

`Node *`

i. (2 pts) `argv[0]`

`char *`

j. (2 pts) `&r`

`double *`

k. (2 pts) `w->next`

`Node *`

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 parameters 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 parameters 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;
}
```

Exam #229 Page: 8 Name: _____

End of Exam

total points=100

Exam #230 Page: 1 Name: _____

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 1100 0010 from base 2 to decimal **194**

b. (2 pts) Convert 7e56 from base 16 to binary **0111 1110 0101 0110**

c. (2 pts) Convert 1110 0111 from base 2 to base 10 **231**

d. (2 pts) Convert 0011 0111 from base 2 to decimal **55**

e. (2 pts) Convert 0110 0011 0001 0011 from base 2 to hexadecimal **6313**

f. (2 pts) Convert 010 010 000 from binary to octal **220**

g. (2 pts) Convert 1100 1010 0101 0011 from binary to base 16 **ca53**

h. (2 pts) Convert 110 111 from binary to octal **67**

i. (2 pts) Convert 1001 0010 0000 0111 from binary to base 16 **9207**

j. (2 pts) Convert 101 111 100 from binary to octal **574**

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 cherry apple lime
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][0]`? c

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

d. (2 pts) What is the value of `argv[1][0]`? l

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double q;  
    int r;  
    Node s;  
    char t;  
    double *w;  
    int *x;  
    Node *y;  
    char *z;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[0]` **char ***

b. (2 pts) `*x` **int**

c. (2 pts) `x` **int ***

d. (2 pts) `&z` **char ****

e. (2 pts) `argc` **int**

f. (2 pts) `argv[1][2]` **char**

g. (2 pts) `&q` **double ***

h. (2 pts) `s` **Node**

i. (2 pts) `y->next` **Node ***

j. (2 pts) `y->data` **int**

k. (2 pts) `y->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #230 Page: 8 Name: _____

End of Exam

total points=100

Exam #231 Page: 1 Name: _____

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 78d from base 16 to base 2 **0111 1000 1101**

b. (2 pts) Convert 001 110 000 from binary to base 8 **160**

c. (2 pts) Convert 110 101 011 from binary to base 8 **653**

d. (2 pts) Convert 60 from base 8 to base 2 **110 000**

e. (2 pts) Convert 21 from base 8 to base 2 **010 001**

f. (2 pts) Convert 70 from octal to base 2 **111 000**

g. (2 pts) Convert 0111 0100 0100 0100 from base 2 to hexadecimal **7444**

h. (2 pts) Convert 163 from decimal to binary **1010 0011**

i. (2 pts) Convert 5c1e from base 16 to base 2 **0101 1100 0001 1110**

j. (2 pts) Convert 010 110 100 from base 2 to octal **264**

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 fig lime guava
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? i

c. (2 pts) What is the value of `argv[1][0]`? l

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double g;  
    int h;  
    Node p;  
    char q;  
    double *r;  
    int *s;  
    Node *t;  
    char *w;  
  
    return 0;  
}
```

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

- a. (2 pts) `argc` **int**
- b. (2 pts) `t->data` **int**
- c. (2 pts) `t->next->next` **Node ***
- d. (2 pts) `&s` **int ****
- e. (2 pts) `g` **double**
- f. (2 pts) `r` **double ***
- g. (2 pts) `argv[0]` **char ***
- h. (2 pts) `*w` **char**
- i. (2 pts) `argv[1][2]` **char**
- j. (2 pts) `t->next` **Node ***
- k. (2 pts) `&q` **char ***

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 parameters 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 parameters 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;
}
```

Exam #231 Page: 8 Name: _____

End of Exam

total points=100

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 0101 1111 1110 1010 from base 2 to base 16 **5fea**

b. (2 pts) Convert 219 from decimal to base 2 **1101 1011**

c. (2 pts) Convert 0111 1011 from base 2 to decimal **123**

d. (2 pts) Convert 7efb from base 16 to base 2 **0111 1110 1111 1011**

e. (2 pts) Convert 1011 0110 1101 1000 from base 2 to hexadecimal **b6d8**

f. (2 pts) Convert e366 from hexadecimal to base 2 **1110 0011 0110 0110**

g. (2 pts) Convert 1100 0101 from base 2 to base 10 **197**

h. (2 pts) Convert 49d3 from hexadecimal to base 2 **0100 1001 1101 0011**

i. (2 pts) Convert 4e92 from hexadecimal to binary **0100 1110 1001 0010**

j. (2 pts) Convert 31 from base 8 to base 2 **011 001**

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 fig cherry lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][1]`? i

c. (2 pts) What is the value of `argv[0][0]`? .

d. (2 pts) What is the value of `argv[2][4]`? r

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node z;  
    int a;  
    double b;  
    char c;  
    Node *d;  
    int *e;  
    double *f;  
    char *g;  
  
    return 0;  
}
```

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

- a. (2 pts) `argc` **int**
- b. (2 pts) `&d` **Node ****
- c. (2 pts) `d->next` **Node ***
- d. (2 pts) `c` **char**
- e. (2 pts) `argv[0]` **char ***
- f. (2 pts) `d->next->next` **Node ***
- g. (2 pts) `g` **char ***
- h. (2 pts) `d->data` **int**
- i. (2 pts) `&c` **char ***
- j. (2 pts) `*f` **double**
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #232 Page: 8 Name: _____

End of Exam

total points=100

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 101 001 001 from base 2 to base 8 **511**

b. (2 pts) Convert 45 from octal to binary **100 101**

c. (2 pts) Convert 0110 1000 from binary to decimal **104**

d. (2 pts) Convert ae5 from base 16 to binary **1010 1110 0101**

e. (2 pts) Convert 46 from base 8 to base 2 **100 110**

f. (2 pts) Convert 125 from base 10 to base 2 **0111 1101**

g. (2 pts) Convert 0110 1111 from binary to base 10 **111**

h. (2 pts) Convert 110 100 011 from base 2 to base 8 **643**

i. (2 pts) Convert 0001 1000 1010 1001 from base 2 to base 16 **18a9**

j. (2 pts) Convert 0 from base 8 to base 2 **000**

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 apple lemon
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][1]`? **p**

c. (2 pts) What is the value of `argv[0][4]`? **n**

d. (2 pts) What is the value of `argv[2][1]`? **e**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double w;  
    Node x;  
    int y;  
    char z;  
    double *a;  
    Node *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[0]` **char ***

b. (2 pts) `argv[1][2]` **char**

c. (2 pts) `&y` **int ***

d. (2 pts) `b->next` **Node ***

e. (2 pts) `argc` **int**

f. (2 pts) `b->data` **int**

g. (2 pts) `&c` **int ****

h. (2 pts) `y` **int**

i. (2 pts) `c` **int ***

j. (2 pts) `*a` **double**

k. (2 pts) `b->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #233 Page: 8 Name: _____

End of Exam

total points=100

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 65 from octal to binary **110 101**

b. (2 pts) Convert 0110 0110 1111 1001 from base 2 to base 16 **66f9**

c. (2 pts) Convert 159 from decimal to binary **1001 1111**

d. (2 pts) Convert 0110 0111 0010 0110 from base 2 to base 16 **6726**

e. (2 pts) Convert 111 100 000 from base 2 to base 8 **740**

f. (2 pts) Convert 53 from base 8 to binary **101 011**

g. (2 pts) Convert 0111 0010 from base 2 to decimal **114**

h. (2 pts) Convert 001 110 100 from base 2 to base 8 **164**

i. (2 pts) Convert 1011 1010 0110 0011 from binary to hexadecimal **ba63**

j. (2 pts) Convert 001 011 100 from base 2 to octal **134**

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 grape lime lemon mango
```

a. (2 pts) What is the value of `argc` in this case? **5**

b. (2 pts) What is the value of `argv[2][0]`? **l**

c. (2 pts) What is the value of `argv[0][1]`? **/**

d. (2 pts) What is the value of `argv[1][1]`? **r**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int g;  
    double h;  
    Node p;  
    char q;  
    int *r;  
    double *s;  
    Node *t;  
    char *w;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[1][2]` **char**

b. (2 pts) `*w` **char**

c. (2 pts) `argv[0]` **char ***

d. (2 pts) `q` **char**

e. (2 pts) `t->next` **Node ***

f. (2 pts) `&r` **int ****

g. (2 pts) `t->data` **int**

h. (2 pts) `&g` **int ***

i. (2 pts) `argc` **int**

j. (2 pts) `t->next->next` **Node ***

k. (2 pts) `s` **double ***

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 parameters 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 parameters 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;
}
```

Exam #234 Page: 8 Name: _____

End of Exam

total points=100

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 0001 1011 from base 2 to decimal **27**
- b. (2 pts) Convert 0010 0001 from binary to decimal **33**
- c. (2 pts) Convert 141 from decimal to binary **1000 1101**
- d. (2 pts) Convert f311 from base 16 to binary **1111 0011 0001 0001**
- e. (2 pts) Convert 211 from base 10 to base 2 **1101 0011**
- f. (2 pts) Convert 0100 1001 1000 1010 from binary to hexadecimal **498a**
- g. (2 pts) Convert 0001 1100 from binary to base 10 **28**
- h. (2 pts) Convert 194 from decimal to base 2 **1100 0010**
- i. (2 pts) Convert 1000 0100 0111 1011 from base 2 to base 16 **847b**
- j. (2 pts) Convert 62 from octal to base 2 **110 010**

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 guava fig mango cherry
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][3]`? u

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double c;  
    int d;  
    Node e;  
    char f;  
    double *g;  
    int *h;  
    Node *p;  
    char *q;  
  
    return 0;  
}
```

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

a. (2 pts) `p->next->next` **Node ***

b. (2 pts) `argc` **int**

c. (2 pts) `&f` **char ***

d. (2 pts) `d` **int**

e. (2 pts) `g` **double ***

f. (2 pts) `p->next` **Node ***

g. (2 pts) `argv[0]` **char ***

h. (2 pts) `argv[1][2]` **char**

i. (2 pts) `p->data` **int**

j. (2 pts) `&p` **Node ****

k. (2 pts) `*h` **int**

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 parameters 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 parameters 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;
}
```

Exam #235 Page: 8 Name: _____

End of Exam

total points=100

Exam #236 Page: 1 Name: _____

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 0111 0011 from binary to decimal **115**

b. (2 pts) Convert 110 000 111 from binary to base 8 **607**

c. (2 pts) Convert 328b from hexadecimal to base 2 **0011 0010 1000 1011**

d. (2 pts) Convert 101 011 101 from binary to octal **535**

e. (2 pts) Convert 0100 0100 0000 0101 from base 2 to hexadecimal **4405**

f. (2 pts) Convert 010 010 101 from binary to octal **225**

g. (2 pts) Convert 6db9 from base 16 to binary **0110 1101 1011 1001**

h. (2 pts) Convert 32 from octal to base 2 **011 010**

i. (2 pts) Convert 118 from decimal to binary **0111 0110**

j. (2 pts) Convert 1101 0101 1000 0010 from binary to hexadecimal **d582**

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 lime kiwi date
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][0]`? .

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

d. (2 pts) What is the value of `argv[2][2]`? w

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int t;  
    Node w;  
    double x;  
    char y;  
    int *z;  
    Node *a;  
    double *b;  
    char *c;  
  
    return 0;  
}
```

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

a. (2 pts) `c` `char *`

b. (2 pts) `&a` `Node **`

c. (2 pts) `a->next->next` `Node *`

d. (2 pts) `*z` `int`

e. (2 pts) `a->next` `Node *`

f. (2 pts) `&y` `char *`

g. (2 pts) `t` `int`

h. (2 pts) `argv[0]` `char *`

i. (2 pts) `argc` `int`

j. (2 pts) `argv[1][2]` `char`

k. (2 pts) `a->data` `int`

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 parameters 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 parameters 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;
}
```

Exam #236 Page: 8 Name: _____

End of Exam

total points=100

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 1011 1000 0100 0110 from base 2 to base 16 **b846**

b. (2 pts) Convert 0111 1110 from binary to decimal **126**

c. (2 pts) Convert 32 from decimal to base 2 **0010 0000**

d. (2 pts) Convert 0011 1010 from binary to base 10 **58**

e. (2 pts) Convert 001 001 110 from base 2 to base 8 **116**

f. (2 pts) Convert 71 from octal to base 2 **111 001**

g. (2 pts) Convert 0001 0111 1010 1010 from binary to hexadecimal **17aa**

h. (2 pts) Convert 111 100 000 from binary to base 8 **740**

i. (2 pts) Convert 4105 from hexadecimal to base 2 **0100 0001 0000 0101**

j. (2 pts) Convert 0111 0001 0110 1100 from binary to hexadecimal **716c**

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 mango lime
```

a. (2 pts) What is the value of `argc` in this case? 3

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

c. (2 pts) What is the value of `argv[0][0]`? .

d. (2 pts) What is the value of `argv[1][4]`? o

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double q;  
    Node r;  
    int s;  
    char t;  
    double *w;  
    Node *x;  
    int *y;  
    char *z;  
  
    return 0;  
}
```

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

a. (2 pts) x->next->next **Node ***

b. (2 pts) argc **int**

c. (2 pts) t **char**

d. (2 pts) x->data **int**

e. (2 pts) &s **int ***

f. (2 pts) argv[0] **char ***

g. (2 pts) argv[1][2] **char**

h. (2 pts) &z **char ****

i. (2 pts) *y **int**

j. (2 pts) x->next **Node ***

k. (2 pts) x **Node ***

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 parameters 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 parameters 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;
}
```

Exam #237 Page: 8 Name: _____

End of Exam

total points=100

Exam #238 Page: 1 Name: _____

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 e9a8 from hexadecimal to binary **1110 1001 1010 1000**

b. (2 pts) Convert 79 from base 10 to base 2 **0100 1111**

c. (2 pts) Convert 25 from octal to base 2 **010 101**

d. (2 pts) Convert 150 from decimal to base 2 **1001 0110**

e. (2 pts) Convert 0111 1101 from base 2 to base 10 **125**

f. (2 pts) Convert 0001 0111 from base 2 to decimal **23**

g. (2 pts) Convert 0001 1001 from base 2 to base 10 **25**

h. (2 pts) Convert 89 from base 10 to base 2 **0101 1001**

i. (2 pts) Convert e2c0 from hexadecimal to binary **1110 0010 1100 0000**

j. (2 pts) Convert 47 from octal to base 2 **100 111**

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 guava apple lemon grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? p

c. (2 pts) What is the value of `argv[1][4]`? a

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int c;  
    double d;  
    Node e;  
    char f;  
    int *g;  
    double *h;  
    Node *p;  
    char *q;  
  
    return 0;  
}
```

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

a. (2 pts) `p->next->next` **Node ***

b. (2 pts) `c` **int**

c. (2 pts) `&d` **double ***

d. (2 pts) `argv[1][2]` **char**

e. (2 pts) `*h` **double**

f. (2 pts) `p->data` **int**

g. (2 pts) `h` **double ***

h. (2 pts) `argc` **int**

i. (2 pts) `p->next` **Node ***

j. (2 pts) `argv[0]` **char ***

k. (2 pts) `&g` **int ****

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 parameters 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 parameters 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;
}
```

Exam #238 Page: 8 Name: _____

End of Exam

total points=100

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 13 from octal to binary 001 011

b. (2 pts) Convert 9c5 from base 16 to binary 1001 1100 0101

c. (2 pts) Convert 44aa from base 16 to base 2 0100 0100 1010 1010

d. (2 pts) Convert 34 from decimal to binary 0010 0010

e. (2 pts) Convert 60ba from base 16 to binary 0110 0000 1011 1010

f. (2 pts) Convert 1011 0000 from binary to decimal 176

g. (2 pts) Convert c3c9 from hexadecimal to binary 1100 0011 1100 1001

h. (2 pts) Convert e0e4 from base 16 to base 2 1110 0000 1110 0100

i. (2 pts) Convert 1010 1100 1101 0111 from base 2 to base 16 acd7

j. (2 pts) Convert 16 from base 8 to base 2 001 110

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 lime kiwi fig banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][3]`? e

c. (2 pts) What is the value of `argv[2][3]`? i

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int z;  
    double a;  
    Node b;  
    char c;  
    int *d;  
    double *e;  
    Node *f;  
    char *g;  
  
    return 0;  
}
```

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

a. (2 pts) `f->next->next` **Node ***

b. (2 pts) `*g` **char**

c. (2 pts) `b` **Node**

d. (2 pts) `f->next` **Node ***

e. (2 pts) `argc` **int**

f. (2 pts) `&z` **int ***

g. (2 pts) `&f` **Node ****

h. (2 pts) `d` **int ***

i. (2 pts) `f->data` **int**

j. (2 pts) `argv[1][2]` **char**

k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #239 Page: 8 Name: _____

End of Exam

total points=100

Exam #240 Page: 1 Name: _____

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 72 from base 8 to base 2 **111 010**

b. (2 pts) Convert 6778 from base 16 to base 2 **0110 0111 0111 1000**

c. (2 pts) Convert 46 from base 10 to base 2 **0010 1110**

d. (2 pts) Convert 6106 from base 16 to binary **0110 0001 0000 0110**

e. (2 pts) Convert 0110 0111 from base 2 to base 10 **103**

f. (2 pts) Convert 011 101 101 from binary to base 8 **355**

g. (2 pts) Convert 1101 0111 0110 1000 from binary to hexadecimal **d768**

h. (2 pts) Convert 100 100 010 from base 2 to octal **442**

i. (2 pts) Convert 010 111 000 from base 2 to base 8 **270**

j. (2 pts) Convert 1100 0111 1101 1110 from base 2 to hexadecimal **c7de**

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 guava kiwi banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][2]`? a

c. (2 pts) What is the value of `argv[0][2]`? r

d. (2 pts) What is the value of `argv[2][2]`? w

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node x;  
    int y;  
    double z;  
    char a;  
    Node *b;  
    int *c;  
    double *d;  
    char *e;  
  
    return 0;  
}
```

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

- a. (2 pts) b->next **Node ***
- b. (2 pts) argv[0] **char ***
- c. (2 pts) argc **int**
- d. (2 pts) b->data **int**
- e. (2 pts) c **int ***
- f. (2 pts) argv[1][2] **char**
- g. (2 pts) y **int**
- h. (2 pts) *e **char**
- i. (2 pts) b->next->next **Node ***
- j. (2 pts) &x **Node ***
- k. (2 pts) &b **Node ****

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 parameters 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 parameters 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;
}
```

Exam #240 Page: 8 Name: _____

End of Exam

total points=100

Exam #241 Page: 1 Name: _____

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 0010 1111 from base 2 to decimal **47**

b. (2 pts) Convert 0010 0001 1010 0001 from binary to hexadecimal **21a1**

c. (2 pts) Convert 27 from decimal to binary **0001 1011**

d. (2 pts) Convert ecf0 from base 16 to base 2 **1110 1100 1111 0000**

e. (2 pts) Convert 4b12 from base 16 to binary **0100 1011 0001 0010**

f. (2 pts) Convert 4 from octal to base 2 **100**

g. (2 pts) Convert 1000 0001 0101 1001 from base 2 to base 16 **8159**

h. (2 pts) Convert 110 001 from base 2 to octal **61**

i. (2 pts) Convert 0010 0110 from binary to decimal **38**

j. (2 pts) Convert 63c9 from hexadecimal to binary **0110 0011 1100 1001**

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 date lime lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][4]`? n

c. (2 pts) What is the value of `argv[1][0]`? d

d. (2 pts) What is the value of `argv[2][2]`? m

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

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

a. (2 pts) h->next->next **Node ***

b. (2 pts) g **char**

c. (2 pts) h->next **Node ***

d. (2 pts) &g **char ***

e. (2 pts) argv[1][2] **char**

f. (2 pts) argc **int**

g. (2 pts) *h **Node**

h. (2 pts) h->data **int**

i. (2 pts) &q **double ****

j. (2 pts) h **Node ***

k. (2 pts) argv[0] **char ***

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 parameters 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 parameters 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;
}
```

Exam #241 Page: 8 Name: _____

End of Exam

total points=100

Exam #242 Page: 1 Name: _____

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 011 000 010 from binary to octal **302**

b. (2 pts) Convert 74 from base 8 to base 2 **111 100**

c. (2 pts) Convert 24 from octal to base 2 **010 100**

d. (2 pts) Convert 0100 1001 0011 0010 from base 2 to base 16 **4932**

e. (2 pts) Convert 1010 0001 0011 0000 from base 2 to hexadecimal **a130**

f. (2 pts) Convert 0100 0010 from base 2 to decimal **66**

g. (2 pts) Convert 1000 0011 from binary to decimal **131**

h. (2 pts) Convert 81ea from hexadecimal to binary **1000 0001 1110 1010**

i. (2 pts) Convert 110 010 000 from base 2 to base 8 **620**

j. (2 pts) Convert 100 100 000 from binary to octal **440**

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 grape lime
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[2][0]`? **l**

c. (2 pts) What is the value of `argv[0][3]`? **u**

d. (2 pts) What is the value of `argv[1][2]`? **a**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node d;  
    double e;  
    int f;  
    char g;  
    Node *h;  
    double *p;  
    int *q;  
    char *r;  
  
    return 0;  
}
```

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

- a. (2 pts) `&e` **double ***
- b. (2 pts) `h->data` **int**
- c. (2 pts) `*h` **Node**
- d. (2 pts) `r` **char ***
- e. (2 pts) `argc` **int**
- f. (2 pts) `h->next->next` **Node ***
- g. (2 pts) `d` **Node**
- h. (2 pts) `&h` **Node ****
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `h->next` **Node ***
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #242 Page: 8 Name: _____

End of Exam

total points=100

Exam #243 Page: 1 Name: _____

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 166 from decimal to base 2 **1010 0110**

b. (2 pts) Convert 101 011 010 from base 2 to octal **532**

c. (2 pts) Convert 4020 from base 16 to base 2 **0100 0000 0010 0000**

d. (2 pts) Convert 65 from base 8 to binary **110 101**

e. (2 pts) Convert 100 001 000 from base 2 to octal **410**

f. (2 pts) Convert 110 111 001 from base 2 to octal **671**

g. (2 pts) Convert 45 from base 10 to binary **0010 1101**

h. (2 pts) Convert 99a from base 16 to binary **1001 1001 1010**

i. (2 pts) Convert 44 from octal to binary **100 100**

j. (2 pts) Convert 001 010 111 from binary to octal **127**

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 fig guava lime banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? u

c. (2 pts) What is the value of `argv[1][0]`? f

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int a;  
    double b;  
    Node c;  
    char d;  
    int *e;  
    double *f;  
    Node *g;  
    char *h;  
  
    return 0;  
}
```

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

- a. (2 pts) `&g` **Node ****
- b. (2 pts) `argv[1][2]` **char**
- c. (2 pts) `argc` **int**
- d. (2 pts) `&a` **int ***
- e. (2 pts) `g` **Node ***
- f. (2 pts) `argv[0]` **char ***
- g. (2 pts) `g->next` **Node ***
- h. (2 pts) `g->data` **int**
- i. (2 pts) `*f` **double**
- j. (2 pts) `g->next->next` **Node ***
- k. (2 pts) `c` **Node**

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 parameters 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 parameters 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;
}
```

Exam #243 Page: 8 Name: _____

End of Exam

total points=100

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 fe78 from hexadecimal to base 2 **1111 1110 0111 1000**

b. (2 pts) Convert 24 from base 8 to base 2 **010 100**

c. (2 pts) Convert 1110 0101 1000 0001 from binary to hexadecimal **e581**

d. (2 pts) Convert 100 100 001 from binary to octal **441**

e. (2 pts) Convert 244 from base 10 to base 2 **1111 0100**

f. (2 pts) Convert 222 from decimal to binary **1101 1110**

g. (2 pts) Convert 0111 1111 0010 1011 from binary to base 16 **7f2b**

h. (2 pts) Convert 101 100 000 from base 2 to octal **540**

i. (2 pts) Convert 41 from base 8 to base 2 **100 001**

j. (2 pts) Convert 3788 from hexadecimal to base 2 **0011 0111 1000 1000**

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 mango fig kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][6]`? t

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int r;  
    Node s;  
    double t;  
    char w;  
    int *x;  
    Node *y;  
    double *z;  
    char *a;  
  
    return 0;  
}
```

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

- a. (2 pts) `y` **Node ***
- b. (2 pts) `t` **double**
- c. (2 pts) `y->data` **int**
- d. (2 pts) `argc` **int**
- e. (2 pts) `&r` **int ***
- f. (2 pts) `&y` **Node ****
- g. (2 pts) `argv[0]` **char ***
- h. (2 pts) `y->next->next` **Node ***
- i. (2 pts) `y->next` **Node ***
- j. (2 pts) `*y` **Node**
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #244 Page: 8 Name: _____

End of Exam

total points=100

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 20 from octal to base 2 **010 000**

b. (2 pts) Convert 10 from decimal to binary **1010**

c. (2 pts) Convert d362 from base 16 to binary **1101 0011 0110 0010**

d. (2 pts) Convert 0001 1100 from binary to base 10 **28**

e. (2 pts) Convert 1101 1000 0100 0000 from base 2 to hexadecimal **d840**

f. (2 pts) Convert 7801 from hexadecimal to base 2 **0111 1000 0000 0001**

g. (2 pts) Convert 0010 1001 0001 1011 from base 2 to hexadecimal **291b**

h. (2 pts) Convert 55 from base 10 to binary **0011 0111**

i. (2 pts) Convert 010 011 101 from binary to octal **235**

j. (2 pts) Convert d372 from base 16 to base 2 **1101 0011 0111 0010**

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 fig cherry banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[2][4]`? r

c. (2 pts) What is the value of `argv[0][2]`? r

d. (2 pts) What is the value of `argv[1][1]`? i

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int h;  
    Node p;  
    double q;  
    char r;  
    int *s;  
    Node *t;  
    double *w;  
    char *x;  
  
    return 0;  
}
```

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

- a. (2 pts) `t->data` **int**
- b. (2 pts) `h` **int**
- c. (2 pts) `argc` **int**
- d. (2 pts) `&x` **char ****
- e. (2 pts) `&r` **char ***
- f. (2 pts) `t->next` **Node ***
- g. (2 pts) `s` **int ***
- h. (2 pts) `*w` **double**
- i. (2 pts) `argv[1][2]` **char**
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `t->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #245 Page: 8 Name: _____

End of Exam

total points=100

Exam #301 Page: 1 Name: _____

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 16 from base 10 to binary **0001 0000**

b. (2 pts) Convert 67 from base 8 to binary **110 111**

c. (2 pts) Convert 1011 1000 from base 2 to base 10 **184**

d. (2 pts) Convert 4ae8 from base 16 to base 2 **0100 1010 1110 1000**

e. (2 pts) Convert 101 011 101 from binary to octal **535**

f. (2 pts) Convert 7f0a from hexadecimal to base 2 **0111 1111 0000 1010**

g. (2 pts) Convert 101 000 110 from base 2 to octal **506**

h. (2 pts) Convert 13 from octal to base 2 **001 011**

i. (2 pts) Convert 53 from base 8 to binary **101 011**

j. (2 pts) Convert 819f from base 16 to binary **1000 0001 1001 1111**

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 date lime banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][1]`? a

c. (2 pts) What is the value of `argv[0][6]`? t

d. (2 pts) What is the value of `argv[2][2]`? m

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node h;  
    int p;  
    double q;  
    char r;  
    Node *s;  
    int *t;  
    double *w;  
    char *x;  
  
    return 0;  
}
```

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

a. (2 pts) &x **char ****

b. (2 pts) s->next->next **Node ***

c. (2 pts) s->data **int**

d. (2 pts) *t **int**

e. (2 pts) argv[1][2] **char**

f. (2 pts) p **int**

g. (2 pts) x **char ***

h. (2 pts) s->next **Node ***

i. (2 pts) &p **int ***

j. (2 pts) argc **int**

k. (2 pts) argv[0] **char ***

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 parameters 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 parameters 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;
}
```

Exam #301 Page: 8 Name: _____

End of Exam

total points=100

Exam #302 Page: 1 Name: _____

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 010 000 100 from binary to base 8 **204**

b. (2 pts) Convert 1011 0001 0000 0110 from base 2 to base 16 **b106**

c. (2 pts) Convert 1110 1110 1000 0001 from base 2 to hexadecimal **ee81**

d. (2 pts) Convert 1010 0111 0010 1001 from binary to hexadecimal **a729**

e. (2 pts) Convert 0100 from binary to decimal **4**

f. (2 pts) Convert 101 100 010 from base 2 to base 8 **542**

g. (2 pts) Convert 1010 0101 0110 0001 from binary to base 16 **a561**

h. (2 pts) Convert 100 101 001 from base 2 to base 8 **451**

i. (2 pts) Convert 0101 0001 from binary to base 10 **81**

j. (2 pts) Convert 1010 1101 1101 1111 from binary to hexadecimal **addf**

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

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

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

```
./thing cherry banana
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][6]`? g

c. (2 pts) What is the value of `argv[2][3]`? a

d. (2 pts) What is the value of `argv[1][1]`? h

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char p;  
    double q;  
    int r;  
    Node s;  
    char *t;  
    double *w;  
    int *x;  
    Node *y;  
  
    return 0;  
}
```

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

- a. (2 pts) `y->data` **int**
- b. (2 pts) `&t` **char ****
- c. (2 pts) `r` **int**
- d. (2 pts) `x` **int ***
- e. (2 pts) `argc` **int**
- f. (2 pts) `y->next->next` **Node ***
- g. (2 pts) `y->next` **Node ***
- h. (2 pts) `*w` **double**
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `&p` **char ***
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #302 Page: 8 Name: _____

End of Exam

total points=100

Exam #303 Page: 1 Name: _____

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 870f from base 16 to binary **1000 0111 0000 1111**

b. (2 pts) Convert 0110 1011 from binary to decimal **107**

c. (2 pts) Convert 220 from base 10 to binary **1101 1100**

d. (2 pts) Convert 0011 0011 0001 0100 from binary to base 16 **3314**

e. (2 pts) Convert 232 from decimal to binary **1110 1000**

f. (2 pts) Convert 22 from octal to binary **010 010**

g. (2 pts) Convert 0100 1111 0101 0010 from base 2 to base 16 **4f52**

h. (2 pts) Convert 28 from base 10 to base 2 **0001 1100**

i. (2 pts) Convert 6 from base 8 to binary **110**

j. (2 pts) Convert 0100 1001 1100 1010 from base 2 to hexadecimal **49ca**

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

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

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

```
./thing grape banana lime kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][4]`? n

c. (2 pts) What is the value of `argv[0][0]`? .

d. (2 pts) What is the value of `argv[1][4]`? e

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double w;  
    int x;  
    char y;  
    Node z;  
    double *a;  
    int *b;  
    char *c;  
    Node *d;  
  
    return 0;  
}
```

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

- a. (2 pts) `w` **double**
- b. (2 pts) `&c` **char ****
- c. (2 pts) `&z` **Node ***
- d. (2 pts) `argv[0]` **char ***
- e. (2 pts) `d->next` **Node ***
- f. (2 pts) `d->data` **int**
- g. (2 pts) `argv[1][2]` **char**
- h. (2 pts) `b` **int ***
- i. (2 pts) `argc` **int**
- j. (2 pts) `*c` **char**
- k. (2 pts) `d->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #303 Page: 8 Name: _____

End of Exam

total points=100

Exam #304 Page: 1 Name: _____

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 67 from decimal to base 2 **0100 0011**

b. (2 pts) Convert 62 from octal to base 2 **110 010**

c. (2 pts) Convert 1100 0101 from binary to base 10 **197**

d. (2 pts) Convert 34 from octal to binary **011 100**

e. (2 pts) Convert 1110 1111 from binary to decimal **239**

f. (2 pts) Convert 17 from decimal to base 2 **0001 0001**

g. (2 pts) Convert 0110 0010 1111 0001 from base 2 to hexadecimal **62f1**

h. (2 pts) Convert cd05 from hexadecimal to binary **1100 1101 0000 0101**

i. (2 pts) Convert 1100 1011 from base 2 to base 10 **203**

j. (2 pts) Convert 66 from octal to base 2 **110 110**

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

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

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

```
./thing banana lime grape date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][3]`? a

c. (2 pts) What is the value of `argv[2][3]`? e

d. (2 pts) What is the value of `argv[0][6]`? g

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int s;  
    double t;  
    char w;  
    Node x;  
    int *y;  
    double *z;  
    char *a;  
    Node *b;  
  
    return 0;  
}
```

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

- a. (2 pts) `x` **Node**
- b. (2 pts) `b->data` **int**
- c. (2 pts) `b->next->next` **Node ***
- d. (2 pts) `&x` **Node ***
- e. (2 pts) `b->next` **Node ***
- f. (2 pts) `&y` **int ****
- g. (2 pts) `argc` **int**
- h. (2 pts) `*a` **char**
- i. (2 pts) `b` **Node ***
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #304 Page: 8 Name: _____

End of Exam

total points=100

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 42 from octal to base 2 **100 010**

b. (2 pts) Convert 131 from decimal to binary **1000 0011**

c. (2 pts) Convert 1011 0011 from binary to base 10 **179**

d. (2 pts) Convert 77 from octal to base 2 **111 111**

e. (2 pts) Convert d28d from hexadecimal to binary **1101 0010 1000 1101**

f. (2 pts) Convert aae1 from hexadecimal to base 2 **1010 1010 1110 0001**

g. (2 pts) Convert 1100 1110 0010 from binary to hexadecimal **ce2**

h. (2 pts) Convert 010 101 001 from base 2 to octal **251**

i. (2 pts) Convert 149 from decimal to binary **1001 0101**

j. (2 pts) Convert 0111 0011 1111 1011 from binary to base 16 **73fb**

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

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

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

```
./thing lemon apple kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[2][4]`? e

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

d. (2 pts) What is the value of `argv[0][3]`? h

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int p;  
    char q;  
    double r;  
    Node s;  
    int *t;  
    char *w;  
    double *x;  
    Node *y;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[1][2]` **char**
- b. (2 pts) `argc` **int**
- c. (2 pts) `*t` **int**
- d. (2 pts) `argv[0]` **char ***
- e. (2 pts) `x` **double ***
- f. (2 pts) `&x` **double ****
- g. (2 pts) `&r` **double ***
- h. (2 pts) `y->next->next` **Node ***
- i. (2 pts) `y->data` **int**
- j. (2 pts) `y->next` **Node ***
- k. (2 pts) `q` **char**

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 parameters 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 parameters 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;
}
```

Exam #305 Page: 8 Name: _____

End of Exam

total points=100

Exam #306 Page: 1 Name: _____

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 1011 1001 1100 0011 from base 2 to hexadecimal **b9c3**

b. (2 pts) Convert 0101 0100 from base 2 to decimal **84**

c. (2 pts) Convert 72 from base 8 to base 2 **111 010**

d. (2 pts) Convert 010 110 011 from binary to octal **263**

e. (2 pts) Convert 0010 1000 1010 1011 from binary to hexadecimal **28ab**

f. (2 pts) Convert 1101 1101 0100 1011 from base 2 to hexadecimal **dd4b**

g. (2 pts) Convert 1111 0001 0000 from base 2 to base 16 **f10**

h. (2 pts) Convert 101 111 011 from binary to octal **573**

i. (2 pts) Convert 54 from decimal to base 2 **0011 0110**

j. (2 pts) Convert 50 from octal to binary **101 000**

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

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

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

```
./thing banana lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[1][2]`? n

c. (2 pts) What is the value of `argv[0][2]`? t

d. (2 pts) What is the value of `argv[2][1]`? i

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double b;  
    char c;  
    int d;  
    Node e;  
    double *f;  
    char *g;  
    int *h;  
    Node *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `d` **int**
- b. (2 pts) `argv[0]` **char ***
- c. (2 pts) `p->next` **Node ***
- d. (2 pts) `p->next->next` **Node ***
- e. (2 pts) `argv[1][2]` **char**
- f. (2 pts) `&b` **double ***
- g. (2 pts) `p->data` **int**
- h. (2 pts) `&p` **Node ****
- i. (2 pts) `*f` **double**
- j. (2 pts) `argc` **int**
- k. (2 pts) `g` **char ***

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 parameters 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 parameters 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;
}
```

Exam #306 Page: 8 Name: _____

End of Exam

total points=100

Exam #307 Page: 1 Name: _____

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 111 111 101 from base 2 to octal **775**

b. (2 pts) Convert 1110 1011 1000 from base 2 to hexadecimal **eb8**

c. (2 pts) Convert d7d7 from base 16 to base 2 **1101 0111 1101 0111**

d. (2 pts) Convert 71 from octal to base 2 **111 001**

e. (2 pts) Convert 2 from octal to base 2 **010**

f. (2 pts) Convert 7727 from base 16 to binary **0111 0111 0010 0111**

g. (2 pts) Convert 1011 1001 0000 0001 from binary to hexadecimal **b901**

h. (2 pts) Convert 4555 from hexadecimal to binary **0100 0101 0101 0101**

i. (2 pts) Convert fa from hexadecimal to binary **1111 1010**

j. (2 pts) Convert 17 from base 8 to base 2 **001 111**

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

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

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

```
./thing lime guava banana grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? u

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double y;  
    int z;  
    char a;  
    Node b;  
    double *c;  
    int *d;  
    char *e;  
    Node *f;  
  
    return 0;  
}
```

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

a. (2 pts) `y` **double**

b. (2 pts) `f->next->next` **Node ***

c. (2 pts) `argc` **int**

d. (2 pts) `c` **double ***

e. (2 pts) `f->data` **int**

f. (2 pts) `f->next` **Node ***

g. (2 pts) `&e` **char ****

h. (2 pts) `argv[1][2]` **char**

i. (2 pts) `&y` **double ***

j. (2 pts) `argv[0]` **char ***

k. (2 pts) `*d` **int**

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 parameters 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 parameters 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;
}
```

Exam #307 Page: 8 Name: _____

End of Exam

total points=100

Exam #308 Page: 1 Name: _____

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 010 101 101 from base 2 to octal **255**

b. (2 pts) Convert b184 from hexadecimal to base 2 **1011 0001 1000 0100**

c. (2 pts) Convert 0111 1101 0011 1001 from binary to hexadecimal **7d39**

d. (2 pts) Convert 161 from decimal to base 2 **1010 0001**

e. (2 pts) Convert 124 from decimal to binary **0111 1100**

f. (2 pts) Convert 0111 1000 from binary to decimal **120**

g. (2 pts) Convert 1010 from binary to base 10 **10**

h. (2 pts) Convert 72 from octal to binary **111 010**

i. (2 pts) Convert f36d from base 16 to base 2 **1111 0011 0110 1101**

j. (2 pts) Convert 0100 0111 1011 1011 from base 2 to hexadecimal **47bb**

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

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

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

```
./thing apple fig date guava
```

a. (2 pts) What is the value of `argc` in this case? 5

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

c. (2 pts) What is the value of `argv[1][4]`? e

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double p;  
    int q;  
    char r;  
    Node s;  
    double *t;  
    int *w;  
    char *x;  
    Node *y;  
  
    return 0;  
}
```

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

a. (2 pts) `y->next->next` **Node ***

b. (2 pts) `x` **char ***

c. (2 pts) `&s` **Node ***

d. (2 pts) `&t` **double ****

e. (2 pts) `argv[1][2]` **char**

f. (2 pts) `argc` **int**

g. (2 pts) `*t` **double**

h. (2 pts) `y->next` **Node ***

i. (2 pts) `argv[0]` **char ***

j. (2 pts) `p` **double**

k. (2 pts) `y->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #308 Page: 8 Name: _____

End of Exam

total points=100

Exam #309 Page: 1 Name: _____

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 9bde from base 16 to base 2 **1001 1011 1101 1110**

b. (2 pts) Convert 6bad from hexadecimal to binary **0110 1011 1010 1101**

c. (2 pts) Convert 107 from base 10 to binary **0110 1011**

d. (2 pts) Convert 2cf3 from hexadecimal to binary **0010 1100 1111 0011**

e. (2 pts) Convert 5fbb from hexadecimal to base 2 **0101 1111 1011 1011**

f. (2 pts) Convert 100 100 from binary to base 8 **44**

g. (2 pts) Convert 1011 0100 from base 2 to base 10 **180**

h. (2 pts) Convert 011 100 110 from binary to octal **346**

i. (2 pts) Convert 1011 1101 from binary to base 10 **189**

j. (2 pts) Convert 1110 0011 1010 0101 from base 2 to hexadecimal **e3a5**

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

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

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

```
./thing grape kiwi apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][1]`? r

c. (2 pts) What is the value of `argv[0][1]`? /

d. (2 pts) What is the value of `argv[2][3]`? i

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char e;  
    int f;  
    double g;  
    Node h;  
    char *p;  
    int *q;  
    double *r;  
    Node *s;  
  
    return 0;  
}
```

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

a. (2 pts) `s->next` **Node ***

b. (2 pts) `s->next->next` **Node ***

c. (2 pts) `*r` **double**

d. (2 pts) `argv[1][2]` **char**

e. (2 pts) `s->data` **int**

f. (2 pts) `argv[0]` **char ***

g. (2 pts) `g` **double**

h. (2 pts) `&s` **Node ****

i. (2 pts) `argc` **int**

j. (2 pts) `&g` **double ***

k. (2 pts) `q` **int ***

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 parameters 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 parameters 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;
}
```

Exam #309 Page: 8 Name: _____

End of Exam

total points=100

Exam #310 Page: 1 Name: _____

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 1100 1101 from binary to decimal **205**

b. (2 pts) Convert 3d33 from base 16 to base 2 **0011 1101 0011 0011**

c. (2 pts) Convert 50 from base 8 to base 2 **101 000**

d. (2 pts) Convert 1000 1001 0011 0101 from binary to base 16 **8935**

e. (2 pts) Convert 1011 0101 1101 1001 from binary to base 16 **b5d9**

f. (2 pts) Convert 68 from base 10 to base 2 **0100 0100**

g. (2 pts) Convert 101 101 101 from base 2 to octal **555**

h. (2 pts) Convert 220 from base 10 to binary **1101 1100**

i. (2 pts) Convert 0101 1111 from base 2 to base 10 **95**

j. (2 pts) Convert 1111 1110 0110 from base 2 to base 16 **fe6**

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

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

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

```
./thing fig apple
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][4]`? **i**

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

d. (2 pts) What is the value of `argv[1][1]`? **i**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char x;  
    double y;  
    int z;  
    Node a;  
    char *b;  
    double *c;  
    int *d;  
    Node *e;  
  
    return 0;  
}
```

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

a. (2 pts) `a` **Node**

b. (2 pts) `argv[0]` **char ***

c. (2 pts) `e->next->next` **Node ***

d. (2 pts) `&y` **double ***

e. (2 pts) `b` **char ***

f. (2 pts) `*c` **double**

g. (2 pts) `e->next` **Node ***

h. (2 pts) `argc` **int**

i. (2 pts) `argv[1][2]` **char**

j. (2 pts) `e->data` **int**

k. (2 pts) `&b` **char ****

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 parameters 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 parameters 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;
}
```

Exam #310 Page: 8 Name: _____

End of Exam

total points=100

Exam #311 Page: 1 Name: _____

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 0001 0010 0010 0000 from base 2 to hexadecimal **1220**

b. (2 pts) Convert 111 101 110 from base 2 to octal **756**

c. (2 pts) Convert 1000 1111 0101 1000 from binary to hexadecimal **8f58**

d. (2 pts) Convert 0001 0101 0001 1111 from binary to hexadecimal **151f**

e. (2 pts) Convert 100 110 010 from binary to base 8 **462**

f. (2 pts) Convert 222 from base 10 to binary **1101 1110**

g. (2 pts) Convert 011 000 001 from base 2 to base 8 **301**

h. (2 pts) Convert 6416 from hexadecimal to binary **0110 0100 0001 0110**

i. (2 pts) Convert 41 from decimal to binary **0010 1001**

j. (2 pts) Convert 1010 1011 1101 0000 from binary to base 16 **abd0**

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

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

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

```
./thing grape guava
```

a. (2 pts) What is the value of `argc` in this case? **3**

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

c. (2 pts) What is the value of `argv[0][2]`? **t**

d. (2 pts) What is the value of `argv[1][0]`? **g**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double s;  
    char t;  
    int w;  
    Node x;  
    double *y;  
    char *z;  
    int *a;  
    Node *b;  
  
    return 0;  
}
```

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

a. (2 pts) b->next->next **Node ***

b. (2 pts) b **Node ***

c. (2 pts) argv[1][2] **char**

d. (2 pts) &a **int ****

e. (2 pts) *b **Node**

f. (2 pts) argc **int**

g. (2 pts) t **char**

h. (2 pts) &s **double ***

i. (2 pts) b->next **Node ***

j. (2 pts) b->data **int**

k. (2 pts) argv[0] **char ***

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 parameters 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 parameters 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;
}
```

Exam #311 Page: 8 Name: _____

End of Exam

total points=100

Exam #312 Page: 1 Name: _____

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 0110 1010 0111 1100 from binary to hexadecimal **6a7c**

b. (2 pts) Convert 1001 1010 0010 1000 from base 2 to base 16 **9a28**

c. (2 pts) Convert 15 from base 8 to base 2 **001 101**

d. (2 pts) Convert 208 from base 10 to base 2 **1101 0000**

e. (2 pts) Convert 2 from octal to binary **010**

f. (2 pts) Convert dfeb from base 16 to binary **1101 1111 1110 1011**

g. (2 pts) Convert 1011 0010 from binary to decimal **178**

h. (2 pts) Convert 010 101 from binary to octal **25**

i. (2 pts) Convert 27 from base 10 to binary **0001 1011**

j. (2 pts) Convert 1011 0111 from base 2 to decimal **183**

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

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

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

```
./thing cherry kiwi banana lime
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? i

c. (2 pts) What is the value of `argv[1][5]`? y

d. (2 pts) What is the value of `argv[0][3]`? h

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int e;  
    double f;  
    char g;  
    Node h;  
    int *p;  
    double *q;  
    char *r;  
    Node *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `s->data` **int**
- b. (2 pts) `argv[1][2]` **char**
- c. (2 pts) `&q` **double ****
- d. (2 pts) `argv[0]` **char ***
- e. (2 pts) `s->next->next` **Node ***
- f. (2 pts) `&e` **int ***
- g. (2 pts) `*r` **char**
- h. (2 pts) `s->next` **Node ***
- i. (2 pts) `r` **char ***
- j. (2 pts) `argc` **int**
- k. (2 pts) `f` **double**

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 parameters 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 parameters 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;
}
```

Exam #312 Page: 8 Name: _____

End of Exam

total points=100

Exam #313 Page: 1 Name: _____

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 1010 1111 from base 2 to decimal **175**

b. (2 pts) Convert 25 from octal to binary **010 101**

c. (2 pts) Convert 10 from octal to base 2 **001 000**

d. (2 pts) Convert 010 111 001 from base 2 to base 8 **271**

e. (2 pts) Convert 111 011 001 from binary to base 8 **731**

f. (2 pts) Convert 0111 1001 1100 1000 from binary to hexadecimal **79c8**

g. (2 pts) Convert 5c67 from base 16 to binary **0101 1100 0110 0111**

h. (2 pts) Convert 146 from decimal to binary **1001 0010**

i. (2 pts) Convert 1110 0101 from binary to base 10 **229**

j. (2 pts) Convert 0101 0011 from binary to base 10 **83**

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

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

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

```
./thing banana date kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][3]`? a

c. (2 pts) What is the value of `argv[0][4]`? i

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char b;  
    int c;  
    double d;  
    Node e;  
    char *f;  
    int *g;  
    double *h;  
    Node *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[1][2]` **char**
- b. (2 pts) `g` **int ***
- c. (2 pts) `&e` **Node ***
- d. (2 pts) `*f` **char**
- e. (2 pts) `e` **Node**
- f. (2 pts) `p->next->next` **Node ***
- g. (2 pts) `argv[0]` **char ***
- h. (2 pts) `p->data` **int**
- i. (2 pts) `p->next` **Node ***
- j. (2 pts) `argc` **int**
- k. (2 pts) `&p` **Node ****

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 parameters 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 parameters 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;
}
```

Exam #313 Page: 8 Name: _____

End of Exam

total points=100

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 70 from octal to base 2 **111 000**

b. (2 pts) Convert 001 001 011 from base 2 to base 8 **113**

c. (2 pts) Convert 010 110 001 from binary to octal **261**

d. (2 pts) Convert b8de from hexadecimal to base 2 **1011 1000 1101 1110**

e. (2 pts) Convert 20 from octal to base 2 **010 000**

f. (2 pts) Convert 53 from octal to base 2 **101 011**

g. (2 pts) Convert 0101 1110 from binary to base 10 **94**

h. (2 pts) Convert 251 from base 10 to binary **1111 1011**

i. (2 pts) Convert 135 from base 10 to binary **1000 0111**

j. (2 pts) Convert 7f8f from hexadecimal to binary **0111 1111 1000 1111**

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

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

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

```
./thing lime fig
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][2]`? **t**

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char b;  
    double c;  
    int d;  
    Node e;  
    char *f;  
    double *g;  
    int *h;  
    Node *p;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[0]` **char ***

b. (2 pts) `p->next->next` **Node ***

c. (2 pts) `argv[1][2]` **char**

d. (2 pts) `p->next` **Node ***

e. (2 pts) `p->data` **int**

f. (2 pts) `f` **char ***

g. (2 pts) `*p` **Node**

h. (2 pts) `b` **char**

i. (2 pts) `&g` **double ****

j. (2 pts) `&c` **double ***

k. (2 pts) `argc` **int**

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 parameters 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 parameters 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;
}
```

Exam #314 Page: 8 Name: _____

End of Exam

total points=100

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 37 from decimal to binary **0010 0101**

b. (2 pts) Convert 1101 1111 from base 2 to decimal **223**

c. (2 pts) Convert 010 001 101 from binary to octal **215**

d. (2 pts) Convert 0100 0100 from base 2 to decimal **68**

e. (2 pts) Convert 2651 from hexadecimal to binary **0010 0110 0101 0001**

f. (2 pts) Convert 0100 0110 0000 1111 from base 2 to hexadecimal **460f**

g. (2 pts) Convert 1000 from base 2 to decimal **8**

h. (2 pts) Convert 82d8 from base 16 to binary **1000 0010 1101 1000**

i. (2 pts) Convert 0101 0001 from binary to base 10 **81**

j. (2 pts) Convert 1b7a from base 16 to binary **0001 1011 0111 1010**

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

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

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

```
./thing apple date
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[2][1]`? a

c. (2 pts) What is the value of `argv[0][0]`? .

d. (2 pts) What is the value of `argv[1][0]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char h;  
    double p;  
    int q;  
    Node r;  
    char *s;  
    double *t;  
    int *w;  
    Node *x;  
  
    return 0;  
}
```

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

a. (2 pts) x->next->next **Node ***

b. (2 pts) argc **int**

c. (2 pts) x->data **int**

d. (2 pts) argv[1][2] **char**

e. (2 pts) r **Node**

f. (2 pts) x **Node ***

g. (2 pts) argv[0] **char ***

h. (2 pts) *t **double**

i. (2 pts) &p **double ***

j. (2 pts) &x **Node ****

k. (2 pts) x->next **Node ***

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 parameters 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 parameters 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;
}
```

Exam #315 Page: 8 Name: _____

End of Exam

total points=100

Exam #316 Page: 1 Name: _____

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 125 from decimal to binary **0111 1101**

b. (2 pts) Convert 100 000 101 from base 2 to octal **405**

c. (2 pts) Convert 1110 1100 from binary to base 10 **236**

d. (2 pts) Convert 0 from base 8 to binary **000**

e. (2 pts) Convert 45 from octal to base 2 **100 101**

f. (2 pts) Convert 71 from base 10 to binary **0100 0111**

g. (2 pts) Convert 0101 1010 0011 1001 from binary to base 16 **5a39**

h. (2 pts) Convert 001 010 010 from base 2 to octal **122**

i. (2 pts) Convert 0100 0100 from base 2 to decimal **68**

j. (2 pts) Convert 001 001 110 from binary to base 8 **116**

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

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

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

```
./thing lime guava grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][0]`? l

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int a;  
    char b;  
    double c;  
    Node d;  
    int *e;  
    char *f;  
    double *g;  
    Node *h;  
  
    return 0;  
}
```

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

- a. (2 pts) `&a` **int ***
- b. (2 pts) `h->data` **int**
- c. (2 pts) `c` **double**
- d. (2 pts) `h->next` **Node ***
- e. (2 pts) `argv[0]` **char ***
- f. (2 pts) `argc` **int**
- g. (2 pts) `g` **double ***
- h. (2 pts) `&g` **double ****
- i. (2 pts) `h->next->next` **Node ***
- j. (2 pts) `argv[1][2]` **char**
- k. (2 pts) `*e` **int**

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 parameters 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 parameters 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;
}
```

Exam #316 Page: 8 Name: _____

End of Exam

total points=100

Exam #317 Page: 1 Name: _____

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 1100 0010 1101 1001 from binary to hexadecimal **c2d9**

b. (2 pts) Convert 60 from decimal to base 2 **0011 1100**

c. (2 pts) Convert 110 110 100 from binary to octal **664**

d. (2 pts) Convert 43 from base 8 to base 2 **100 011**

e. (2 pts) Convert 0111 1010 from binary to decimal **122**

f. (2 pts) Convert 1110 0001 from binary to decimal **225**

g. (2 pts) Convert 0100 0010 1001 from base 2 to base 16 **429**

h. (2 pts) Convert 176 from decimal to binary **1011 0000**

i. (2 pts) Convert 14 from decimal to base 2 **1110**

j. (2 pts) Convert 110 000 101 from base 2 to base 8 **605**

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

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

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

```
./thing guava cherry
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[2][5]`? y

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

d. (2 pts) What is the value of `argv[0][4]`? i

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char x;  
    double y;  
    int z;  
    Node a;  
    char *b;  
    double *c;  
    int *d;  
    Node *e;  
  
    return 0;  
}
```

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

- a. (2 pts) `&a` **Node ***
- b. (2 pts) `&b` **char ****
- c. (2 pts) `e->next->next` **Node ***
- d. (2 pts) `*d` **int**
- e. (2 pts) `b` **char ***
- f. (2 pts) `e->next` **Node ***
- g. (2 pts) `argv[0]` **char ***
- h. (2 pts) `x` **char**
- i. (2 pts) `argc` **int**
- j. (2 pts) `e->data` **int**
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #317 Page: 8 Name: _____

End of Exam

total points=100

Exam #318 Page: 1 Name: _____

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 f43b from base 16 to base 2 **1111 0100 0011 1011**

b. (2 pts) Convert 14 from decimal to base 2 **1110**

c. (2 pts) Convert 107b from hexadecimal to base 2 **0001 0000 0111 1011**

d. (2 pts) Convert 111 010 001 from base 2 to octal **721**

e. (2 pts) Convert 208 from decimal to binary **1101 0000**

f. (2 pts) Convert 0001 0011 1010 0101 from binary to base 16 **13a5**

g. (2 pts) Convert 658 from hexadecimal to base 2 **0110 0101 1000**

h. (2 pts) Convert 19e9 from hexadecimal to binary **0001 1001 1110 1001**

i. (2 pts) Convert 1010 1111 from base 2 to decimal **175**

j. (2 pts) Convert 1110 1111 from base 2 to base 10 **239**

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

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

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

```
./thing lime cherry
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][6]`? g

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double x;  
    char y;  
    int z;  
    Node a;  
    double *b;  
    char *c;  
    int *d;  
    Node *e;  
  
    return 0;  
}
```

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

- a. (2 pts) `e->next` **Node ***
- b. (2 pts) `b` **double ***
- c. (2 pts) `*c` **char**
- d. (2 pts) `y` **char**
- e. (2 pts) `argc` **int**
- f. (2 pts) `e->data` **int**
- g. (2 pts) `&c` **char ****
- h. (2 pts) `&z` **int ***
- i. (2 pts) `argv[1][2]` **char**
- j. (2 pts) `e->next->next` **Node ***
- k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #318 Page: 8 Name: _____

End of Exam

total points=100

Exam #319 Page: 1 Name: _____

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 16 from base 8 to base 2 **001 110**

b. (2 pts) Convert 62 from base 8 to base 2 **110 010**

c. (2 pts) Convert 1111 1110 from base 2 to base 10 **254**

d. (2 pts) Convert 35 from base 8 to base 2 **011 101**

e. (2 pts) Convert 1011 0011 0111 1111 from binary to hexadecimal **b37f**

f. (2 pts) Convert ad82 from base 16 to base 2 **1010 1101 1000 0010**

g. (2 pts) Convert 1011 0000 0100 1000 from base 2 to base 16 **b048**

h. (2 pts) Convert 50 from base 8 to binary **101 000**

i. (2 pts) Convert 011 110 100 from binary to base 8 **364**

j. (2 pts) Convert 1000 1011 from binary to base 10 **139**

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

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

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

```
./thing banana fig lime grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][4]`? n

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double s;  
    int t;  
    char w;  
    Node x;  
    double *y;  
    int *z;  
    char *a;  
    Node *b;  
  
    return 0;  
}
```

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

a. (2 pts) `&t` **int ***

b. (2 pts) `b->data` **int**

c. (2 pts) `s` **double**

d. (2 pts) `b->next->next` **Node ***

e. (2 pts) `&y` **double ****

f. (2 pts) `*b` **Node**

g. (2 pts) `argv[0]` **char ***

h. (2 pts) `argv[1][2]` **char**

i. (2 pts) `argc` **int**

j. (2 pts) `b->next` **Node ***

k. (2 pts) `b` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #319 Page: 8 Name: _____

End of Exam

total points=100

Exam #320 Page: 1 Name: _____

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 001 011 010 from binary to octal **132**

b. (2 pts) Convert 1011 0000 from base 2 to base 10 **176**

c. (2 pts) Convert 27 from base 8 to base 2 **010 111**

d. (2 pts) Convert 53 from octal to base 2 **101 011**

e. (2 pts) Convert 1000 1101 0110 0011 from binary to base 16 **8d63**

f. (2 pts) Convert ea6f from base 16 to base 2 **1110 1010 0110 1111**

g. (2 pts) Convert 010 000 000 from base 2 to base 8 **200**

h. (2 pts) Convert 62 from base 10 to binary **0011 1110**

i. (2 pts) Convert 1010 1111 0110 0110 from binary to hexadecimal **af66**

j. (2 pts) Convert 0001 0100 0000 0100 from binary to hexadecimal **1404**

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 cherry fig lime lemon
```

a. (2 pts) What is the value of `argc` in this case? 5

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

c. (2 pts) What is the value of `argv[1][5]`? y

d. (2 pts) What is the value of `argv[0][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int z;  
    double a;  
    Node b;  
    char c;  
    int *d;  
    double *e;  
    Node *f;  
    char *g;  
  
    return 0;  
}
```

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

a. (2 pts) `argc`

int

b. (2 pts) `*f`

Node

c. (2 pts) `z`

int

d. (2 pts) `f->data`

int

e. (2 pts) `f->next`

Node *

f. (2 pts) `argv[1][2]`

char

g. (2 pts) `f`

Node *

h. (2 pts) `&a`

double *

i. (2 pts) `argv[0]`

char *

j. (2 pts) `f->next->next`

Node *

k. (2 pts) `&d`

int **

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 parameters 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 parameters 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;
}
```

Exam #320 Page: 8 Name: _____

End of Exam

total points=100

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 723f from hexadecimal to binary 0111 0010 0011 1111

b. (2 pts) Convert 106 from base 10 to binary 0110 1010

c. (2 pts) Convert 4daa from hexadecimal to base 2 0100 1101 1010 1010

d. (2 pts) Convert 3934 from hexadecimal to binary 0011 1001 0011 0100

e. (2 pts) Convert 34 from octal to binary 011 100

f. (2 pts) Convert 132 from base 10 to base 2 1000 0100

g. (2 pts) Convert 1110 1010 from binary to base 10 234

h. (2 pts) Convert 197 from base 10 to binary 1100 0101

i. (2 pts) Convert 36 from base 8 to base 2 011 110

j. (2 pts) Convert 1010 1111 1110 1111 from base 2 to hexadecimal afef

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 kiwi fig grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][2]`? w

c. (2 pts) What is the value of `argv[0][1]`? /

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int w;  
    Node x;  
    double y;  
    char z;  
    int *a;  
    Node *b;  
    double *c;  
    char *d;  
  
    return 0;  
}
```

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

- a. (2 pts) `b->next` **Node ***
- b. (2 pts) `b` **Node ***
- c. (2 pts) `y` **double**
- d. (2 pts) `b->next->next` **Node ***
- e. (2 pts) `*d` **char**
- f. (2 pts) `&w` **int ***
- g. (2 pts) `b->data` **int**
- h. (2 pts) `argv[1][2]` **char**
- i. (2 pts) `argc` **int**
- j. (2 pts) `&c` **double ****
- k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #321 Page: 8 Name: _____

End of Exam

total points=100

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 1010 0011 from base 2 to base 10 **163**

b. (2 pts) Convert 0011 1100 from binary to decimal **60**

c. (2 pts) Convert 100 001 000 from binary to octal **410**

d. (2 pts) Convert 100 101 010 from binary to base 8 **452**

e. (2 pts) Convert 110 001 101 from binary to octal **615**

f. (2 pts) Convert 55 from octal to binary **101 101**

g. (2 pts) Convert 111 011 000 from base 2 to octal **730**

h. (2 pts) Convert 13 from octal to binary **001 011**

i. (2 pts) Convert 6 from base 8 to binary **110**

j. (2 pts) Convert 1101 1100 0011 0000 from base 2 to hexadecimal **dc30**

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 guava date
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][2]`? r

c. (2 pts) What is the value of `argv[1][3]`? v

d. (2 pts) What is the value of `argv[2][2]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double x;  
    Node y;  
    int z;  
    char a;  
    double *b;  
    Node *c;  
    int *d;  
    char *e;  
  
    return 0;  
}
```

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

a. (2 pts) `argc` **int**

b. (2 pts) `argv[0]` **char ***

c. (2 pts) `*e` **char**

d. (2 pts) `&a` **char ***

e. (2 pts) `&e` **char ****

f. (2 pts) `c->next` **Node ***

g. (2 pts) `c->next->next` **Node ***

h. (2 pts) `b` **double ***

i. (2 pts) `c->data` **int**

j. (2 pts) `a` **char**

k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #322 Page: 8 Name: _____

End of Exam

total points=100

Exam #323 Page: 1 Name: _____

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 1110 1000 1000 0001 from binary to hexadecimal **e881**

b. (2 pts) Convert 1111 0110 0101 1110 from binary to hexadecimal **f65e**

c. (2 pts) Convert 0111 0001 from base 2 to decimal **113**

d. (2 pts) Convert 10 from octal to binary **001 000**

e. (2 pts) Convert 170 from decimal to base 2 **1010 1010**

f. (2 pts) Convert 0101 0000 1001 0011 from binary to hexadecimal **5093**

g. (2 pts) Convert 100 101 100 from binary to base 8 **454**

h. (2 pts) Convert 55 from base 8 to binary **101 101**

i. (2 pts) Convert 1110 0101 0100 1111 from base 2 to base 16 **e54f**

j. (2 pts) Convert 0111 1000 0001 1010 from binary to hexadecimal **781a**

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 guava date fig lime
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][1]`? u

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int d;  
    double e;  
    Node f;  
    char g;  
    int *h;  
    double *p;  
    Node *q;  
    char *r;  
  
    return 0;  
}
```

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

- a. (2 pts) q->data **int**
- b. (2 pts) *h **int**
- c. (2 pts) q->next->next **Node ***
- d. (2 pts) argv[0] **char ***
- e. (2 pts) argv[1][2] **char**
- f. (2 pts) r **char ***
- g. (2 pts) &q **Node ****
- h. (2 pts) argc **int**
- i. (2 pts) q->next **Node ***
- j. (2 pts) &f **Node ***
- k. (2 pts) f **Node**

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 parameters 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 parameters 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;
}
```

Exam #323 Page: 8 Name: _____

End of Exam

total points=100

Exam #324 Page: 1 Name: _____

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 0100 0000 1101 1101 from base 2 to base 16 **40dd**

b. (2 pts) Convert 992a from base 16 to base 2 **1001 1001 0010 1010**

c. (2 pts) Convert 101 110 from binary to base 8 **56**

d. (2 pts) Convert 220 from base 10 to binary **1101 1100**

e. (2 pts) Convert 110 101 from binary to octal **65**

f. (2 pts) Convert 81 from base 10 to binary **0101 0001**

g. (2 pts) Convert e7d2 from base 16 to binary **1110 0111 1101 0010**

h. (2 pts) Convert 92 from base 10 to binary **0101 1100**

i. (2 pts) Convert 1101 0111 1100 0011 from binary to hexadecimal **d7c3**

j. (2 pts) Convert 83ae from base 16 to base 2 **1000 0011 1010 1110**

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 kiwi grape guava lime
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][4]`? e

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

d. (2 pts) What is the value of `argv[0][6]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double e;  
    int f;  
    Node g;  
    char h;  
    double *p;  
    int *q;  
    Node *r;  
    char *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `&p` **double ****
- b. (2 pts) `r->next->next` **Node ***
- c. (2 pts) `r->next` **Node ***
- d. (2 pts) `r->data` **int**
- e. (2 pts) `&f` **int ***
- f. (2 pts) `argv[1][2]` **char**
- g. (2 pts) `f` **int**
- h. (2 pts) `*s` **char**
- i. (2 pts) `r` **Node ***
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `argc` **int**

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 parameters 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 parameters 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;
}
```

Exam #324 Page: 8 Name: _____

End of Exam

total points=100

Exam #325 Page: 1 Name: _____

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 100 001 011 from binary to octal **413**

b. (2 pts) Convert 0101 0011 0101 0011 from base 2 to hexadecimal **5353**

c. (2 pts) Convert 001 010 from binary to octal **12**

d. (2 pts) Convert 68dd from base 16 to base 2 **0110 1000 1101 1101**

e. (2 pts) Convert 253 from decimal to binary **1111 1101**

f. (2 pts) Convert 1110 1011 from base 2 to base 10 **235**

g. (2 pts) Convert 91c2 from hexadecimal to binary **1001 0001 1100 0010**

h. (2 pts) Convert 71 from base 8 to binary **111 001**

i. (2 pts) Convert 1010 0001 1101 1010 from base 2 to base 16 **a1da**

j. (2 pts) Convert 1f98 from hexadecimal to base 2 **0001 1111 1001 1000**

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 kiwi guava
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][1]`? e

c. (2 pts) What is the value of `argv[0][4]`? n

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int q;  
    Node r;  
    double s;  
    char t;  
    int *w;  
    Node *x;  
    double *y;  
    char *z;  
  
    return 0;  
}
```

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

a. (2 pts) *x

Node

b. (2 pts) x->data

int

c. (2 pts) &z

char **

d. (2 pts) t

char

e. (2 pts) argv[0]

char *

f. (2 pts) argc

int

g. (2 pts) &r

Node *

h. (2 pts) x->next->next

Node *

i. (2 pts) x

Node *

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

char

k. (2 pts) x->next

Node *

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 parameters 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 parameters 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;
}
```

Exam #325 Page: 8 Name: _____

End of Exam

total points=100

Exam #326 Page: 1 Name: _____

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 55 from base 8 to base 2 **101 101**

b. (2 pts) Convert 11 from octal to base 2 **001 001**

c. (2 pts) Convert 3b8a from base 16 to base 2 **0011 1011 1000 1010**

d. (2 pts) Convert 1100 0101 0001 1110 from binary to base 16 **c51e**

e. (2 pts) Convert 0101 0011 from binary to base 10 **83**

f. (2 pts) Convert 1e29 from hexadecimal to base 2 **0001 1110 0010 1001**

g. (2 pts) Convert 93f0 from base 16 to base 2 **1001 0011 1111 0000**

h. (2 pts) Convert 23 from octal to binary **010 011**

i. (2 pts) Convert 0100 0011 1001 0100 from binary to hexadecimal **4394**

j. (2 pts) Convert 4bd9 from hexadecimal to base 2 **0100 1011 1101 1001**

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 fig date
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][0]`? .

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

d. (2 pts) What is the value of `argv[1][1]`? i

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double r;  
    Node s;  
    int t;  
    char w;  
    double *x;  
    Node *y;  
    int *z;  
    char *a;  
  
    return 0;  
}
```

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

- a. (2 pts) `&w` **char ***
- b. (2 pts) `x` **double ***
- c. (2 pts) `argv[0]` **char ***
- d. (2 pts) `y->next->next` **Node ***
- e. (2 pts) `*y` **Node**
- f. (2 pts) `argv[1][2]` **char**
- g. (2 pts) `s` **Node**
- h. (2 pts) `y->next` **Node ***
- i. (2 pts) `&x` **double ****
- j. (2 pts) `argc` **int**
- k. (2 pts) `y->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #326 Page: 8 Name: _____

End of Exam

total points=100

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 1011 from base 2 to base 10 **251**

b. (2 pts) Convert 110 111 110 from binary to octal **676**

c. (2 pts) Convert 0010 1001 from base 2 to base 10 **41**

d. (2 pts) Convert 0101 0001 0000 1000 from base 2 to base 16 **5108**

e. (2 pts) Convert 15 from octal to base 2 **001 101**

f. (2 pts) Convert 184 from decimal to binary **1011 1000**

g. (2 pts) Convert 3de1 from hexadecimal to binary **0011 1101 1110 0001**

h. (2 pts) Convert 65 from octal to binary **110 101**

i. (2 pts) Convert 3 from base 8 to base 2 **011**

j. (2 pts) Convert e7c4 from base 16 to base 2 **1110 0111 1100 0100**

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 banana date lime apple
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][1]`? a

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

d. (2 pts) What is the value of `argv[1][3]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double h;  
    int p;  
    Node q;  
    char r;  
    double *s;  
    int *t;  
    Node *w;  
    char *x;  
  
    return 0;  
}
```

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

- a. (2 pts) `w->data` **int**
- b. (2 pts) `w->next` **Node ***
- c. (2 pts) `x` **char ***
- d. (2 pts) `argv[1][2]` **char**
- e. (2 pts) `argc` **int**
- f. (2 pts) `*w` **Node**
- g. (2 pts) `&w` **Node ****
- h. (2 pts) `r` **char**
- i. (2 pts) `w->next->next` **Node ***
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `&q` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #327 Page: 8 Name: _____

End of Exam

total points=100

Exam #328 Page: 1 Name: _____

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 84 from base 10 to base 2 **0101 0100**

b. (2 pts) Convert 1000 0001 1100 1101 from binary to base 16 **81cd**

c. (2 pts) Convert 206 from decimal to binary **1100 1110**

d. (2 pts) Convert 011 001 from base 2 to octal **31**

e. (2 pts) Convert 1010 0111 from binary to decimal **167**

f. (2 pts) Convert 1011 1001 0101 0111 from binary to hexadecimal **b957**

g. (2 pts) Convert 100 011 111 from base 2 to base 8 **437**

h. (2 pts) Convert 123 from decimal to binary **0111 1011**

i. (2 pts) Convert 1f from hexadecimal to binary **0001 1111**

j. (2 pts) Convert 243 from decimal to binary **1111 0011**

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 guava lime grape date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][1]`? u

c. (2 pts) What is the value of `argv[0][0]`? .

d. (2 pts) What is the value of `argv[2][1]`? i

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double a;  
    int b;  
    Node c;  
    char d;  
    double *e;  
    int *f;  
    Node *g;  
    char *h;  
  
    return 0;  
}
```

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

a. (2 pts) g->next->next **Node ***

b. (2 pts) &c **Node ***

c. (2 pts) g->next **Node ***

d. (2 pts) argc **int**

e. (2 pts) &f **int ****

f. (2 pts) g **Node ***

g. (2 pts) *f **int**

h. (2 pts) argv[1][2] **char**

i. (2 pts) argv[0] **char ***

j. (2 pts) c **Node**

k. (2 pts) g->data **int**

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 parameters 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 parameters 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;
}
```

Exam #328 Page: 8 Name: _____

End of Exam

total points=100

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 1001 1001 0011 1010 from base 2 to base 16 **993a**

b. (2 pts) Convert 0011 1011 from base 2 to decimal **59**

c. (2 pts) Convert 1011 1100 from base 2 to decimal **188**

d. (2 pts) Convert 100 110 001 from base 2 to base 8 **461**

e. (2 pts) Convert 8b09 from hexadecimal to base 2 **1000 1011 0000 1001**

f. (2 pts) Convert 5334 from base 16 to binary **0101 0011 0011 0100**

g. (2 pts) Convert 001 110 011 from binary to base 8 **163**

h. (2 pts) Convert 0 from base 8 to binary **000**

i. (2 pts) Convert 1100 1010 0011 0110 from base 2 to base 16 **ca36**

j. (2 pts) Convert 143 from base 10 to binary **1000 1111**

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 apple date lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

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

c. (2 pts) What is the value of `argv[1][3]`? l

d. (2 pts) What is the value of `argv[2][1]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node x;  
    int y;  
    double z;  
    char a;  
    Node *b;  
    int *c;  
    double *d;  
    char *e;  
  
    return 0;  
}
```

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

- a. (2 pts) `y` **int**
- b. (2 pts) `b->next` **Node ***
- c. (2 pts) `argv[1][2]` **char**
- d. (2 pts) `b->next->next` **Node ***
- e. (2 pts) `&e` **char ****
- f. (2 pts) `argv[0]` **char ***
- g. (2 pts) `b->data` **int**
- h. (2 pts) `*e` **char**
- i. (2 pts) `argc` **int**
- j. (2 pts) `&y` **int ***
- k. (2 pts) `b` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #329 Page: 8 Name: _____

End of Exam

total points=100

Exam #330 Page: 1 Name: _____

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 ca9c from hexadecimal to binary **1100 1010 1001 1100**

b. (2 pts) Convert 011 010 from base 2 to base 8 **32**

c. (2 pts) Convert 1111 0011 0000 1100 from base 2 to hexadecimal **f30c**

d. (2 pts) Convert 1111 0100 from base 2 to decimal **244**

e. (2 pts) Convert 1110 0001 0010 0111 from binary to base 16 **e127**

f. (2 pts) Convert 100 001 011 from base 2 to base 8 **413**

g. (2 pts) Convert 3bb3 from base 16 to base 2 **0011 1011 1011 0011**

h. (2 pts) Convert 33 from base 8 to binary **011 011**

i. (2 pts) Convert 6bf1 from hexadecimal to binary **0110 1011 1111 0001**

j. (2 pts) Convert bb83 from base 16 to base 2 **1011 1011 1000 0011**

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 mango
```

a. (2 pts) What is the value of `argc` in this case? 3

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

c. (2 pts) What is the value of `argv[2][0]`? m

d. (2 pts) What is the value of `argv[1][2]`? m

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node g;  
    double h;  
    int p;  
    char q;  
    Node *r;  
    double *s;  
    int *t;  
    char *w;  
  
    return 0;  
}
```

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

a. (2 pts) r->next->next **Node ***

b. (2 pts) r **Node ***

c. (2 pts) &s **double ****

d. (2 pts) *t **int**

e. (2 pts) argc **int**

f. (2 pts) argv[1][2] **char**

g. (2 pts) r->next **Node ***

h. (2 pts) argv[0] **char ***

i. (2 pts) p **int**

j. (2 pts) r->data **int**

k. (2 pts) &g **Node ***

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 parameters 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 parameters 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;
}
```

Exam #330 Page: 8 Name: _____

End of Exam

total points=100

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 3 from octal to base 2 011

b. (2 pts) Convert c7a4 from hexadecimal to binary 1100 0111 1010 0100

c. (2 pts) Convert 1110 0000 1110 1100 from binary to hexadecimal e0ec

d. (2 pts) Convert 1000 0000 from base 2 to base 10 128

e. (2 pts) Convert 61 from octal to binary 110 001

f. (2 pts) Convert 7 from octal to binary 111

g. (2 pts) Convert e5a3 from hexadecimal to base 2 1110 0101 1010 0011

h. (2 pts) Convert 243 from decimal to binary 1111 0011

i. (2 pts) Convert 0011 0110 0000 1000 from base 2 to base 16 3608

j. (2 pts) Convert 87 from decimal to base 2 0101 0111

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 fig mango
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][2]`? **r**

c. (2 pts) What is the value of `argv[2][0]`? **m**

d. (2 pts) What is the value of `argv[1][0]`? **f**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node d;  
    double e;  
    int f;  
    char g;  
    Node *h;  
    double *p;  
    int *q;  
    char *r;  
  
    return 0;  
}
```

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

a. (2 pts) *h

Node

b. (2 pts) argv[0]

char *

c. (2 pts) &r

char **

d. (2 pts) r

char *

e. (2 pts) h->next->next

Node *

f. (2 pts) d

Node

g. (2 pts) h->next

Node *

h. (2 pts) &g

char *

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

char

j. (2 pts) h->data

int

k. (2 pts) argc

int

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 parameters 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 parameters 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;
}
```

Exam #331 Page: 8 Name: _____

End of Exam

total points=100

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 31 from octal to base 2 **011 001**

b. (2 pts) Convert 011 010 100 from binary to octal **324**

c. (2 pts) Convert 864e from base 16 to base 2 **1000 0110 0100 1110**

d. (2 pts) Convert 17 from base 8 to base 2 **001 111**

e. (2 pts) Convert 52 from decimal to base 2 **0011 0100**

f. (2 pts) Convert 20ca from hexadecimal to binary **0010 0000 1100 1010**

g. (2 pts) Convert 001 101 110 from base 2 to octal **156**

h. (2 pts) Convert 1001 1010 0101 1011 from binary to hexadecimal **9a5b**

i. (2 pts) Convert 0010 1000 0111 1100 from binary to hexadecimal **287c**

j. (2 pts) Convert 99 from base 10 to binary **0110 0011**

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 guava apple grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][2]`? a

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int w;  
    Node x;  
    double y;  
    char z;  
    int *a;  
    Node *b;  
    double *c;  
    char *d;  
  
    return 0;  
}
```

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

- a. (2 pts) `b->next` **Node ***
- b. (2 pts) `argv[0]` **char ***
- c. (2 pts) `b->data` **int**
- d. (2 pts) `argv[1][2]` **char**
- e. (2 pts) `b` **Node ***
- f. (2 pts) `&c` **double ****
- g. (2 pts) `*d` **char**
- h. (2 pts) `&y` **double ***
- i. (2 pts) `b->next->next` **Node ***
- j. (2 pts) `argc` **int**
- k. (2 pts) `z` **char**

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 parameters 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 parameters 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;
}
```

Exam #332 Page: 8 Name: _____

End of Exam

total points=100

Exam #333 Page: 1 Name: _____

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 172 from decimal to base 2 **1010 1100**

b. (2 pts) Convert 2499 from hexadecimal to base 2 **0010 0100 1001 1001**

c. (2 pts) Convert 742e from hexadecimal to base 2 **0111 0100 0010 1110**

d. (2 pts) Convert c82e from base 16 to binary **1100 1000 0010 1110**

e. (2 pts) Convert 0001 1000 0011 0111 from binary to base 16 **1837**

f. (2 pts) Convert 186 from decimal to binary **1011 1010**

g. (2 pts) Convert 111 000 010 from binary to octal **702**

h. (2 pts) Convert 10 from octal to binary **001 000**

i. (2 pts) Convert f293 from base 16 to base 2 **1111 0010 1001 0011**

j. (2 pts) Convert 1111 1110 from binary to base 10 **254**

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 mango apple
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][2]`? **r**

c. (2 pts) What is the value of `argv[1][4]`? **o**

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node r;  
    double s;  
    int t;  
    char w;  
    Node *x;  
    double *y;  
    int *z;  
    char *a;  
  
    return 0;  
}
```

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

- a. (2 pts) `t` `int`
- b. (2 pts) `argv[1][2]` `char`
- c. (2 pts) `x->next->next` `Node *`
- d. (2 pts) `&t` `int *`
- e. (2 pts) `argc` `int`
- f. (2 pts) `x->data` `int`
- g. (2 pts) `x` `Node *`
- h. (2 pts) `argv[0]` `char *`
- i. (2 pts) `x->next` `Node *`
- j. (2 pts) `*y` `double`
- k. (2 pts) `&x` `Node **`

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 parameters 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 parameters 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;
}
```

Exam #333 Page: 8 Name: _____

End of Exam

total points=100

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 1101 1110 from base 2 to decimal **222**

b. (2 pts) Convert 246 from base 10 to base 2 **1111 0110**

c. (2 pts) Convert 1010 1010 from binary to base 10 **170**

d. (2 pts) Convert 001 001 000 from base 2 to base 8 **110**

e. (2 pts) Convert 0110 1110 0101 0101 from binary to hexadecimal **6e55**

f. (2 pts) Convert 1110 1101 0001 0000 from base 2 to base 16 **ed10**

g. (2 pts) Convert 111 000 110 from base 2 to octal **706**

h. (2 pts) Convert 42 from base 8 to binary **100 010**

i. (2 pts) Convert 45 from octal to binary **100 101**

j. (2 pts) Convert 43 from decimal to binary **0010 1011**

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 fig lemon
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][2]`? **g**

c. (2 pts) What is the value of `argv[0][4]`? **n**

d. (2 pts) What is the value of `argv[2][1]`? **e**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double d;  
    Node e;  
    int f;  
    char g;  
    double *h;  
    Node *p;  
    int *q;  
    char *r;  
  
    return 0;  
}
```

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

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

b. (2 pts) r **char ***

c. (2 pts) argv[0] **char ***

d. (2 pts) argc **int**

e. (2 pts) p->next **Node ***

f. (2 pts) p->data **int**

g. (2 pts) p->next->next **Node ***

h. (2 pts) g **char**

i. (2 pts) &p **Node ****

j. (2 pts) &d **double ***

k. (2 pts) *h **double**

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 parameters 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 parameters 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;
}
```

Exam #334 Page: 8 Name: _____

End of Exam

total points=100

Exam #335 Page: 1 Name: _____

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 22f8 from hexadecimal to binary **0010 0010 1111 1000**

b. (2 pts) Convert 54 from octal to base 2 **101 100**

c. (2 pts) Convert 46 from octal to base 2 **100 110**

d. (2 pts) Convert 54 from base 8 to binary **101 100**

e. (2 pts) Convert 010 100 011 from binary to octal **243**

f. (2 pts) Convert 1000 0110 1110 1101 from binary to base 16 **86ed**

g. (2 pts) Convert 100 011 010 from binary to octal **432**

h. (2 pts) Convert 18 from base 10 to binary **0001 0010**

i. (2 pts) Convert 5e65 from base 16 to base 2 **0101 1110 0110 0101**

j. (2 pts) Convert 199 from decimal to binary **1100 0111**

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 fig date
```

a. (2 pts) What is the value of `argc` in this case? 3

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

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

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node z;  
    double a;  
    int b;  
    char c;  
    Node *d;  
    double *e;  
    int *f;  
    char *g;  
  
    return 0;  
}
```

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

a. (2 pts) `a` **double**

b. (2 pts) `*f` **int**

c. (2 pts) `d->next->next` **Node ***

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `f` **int ***

f. (2 pts) `d->data` **int**

g. (2 pts) `d->next` **Node ***

h. (2 pts) `argc` **int**

i. (2 pts) `&c` **char ***

j. (2 pts) `&d` **Node ****

k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #335 Page: 8 Name: _____

End of Exam

total points=100

Exam #336 Page: 1 Name: _____

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 df6b from hexadecimal to binary **1101 1111 0110 1011**

b. (2 pts) Convert 3 from octal to base 2 **011**

c. (2 pts) Convert 81c3 from base 16 to base 2 **1000 0001 1100 0011**

d. (2 pts) Convert 238 from base 10 to base 2 **1110 1110**

e. (2 pts) Convert 26 from octal to base 2 **010 110**

f. (2 pts) Convert 0100 1100 from binary to decimal **76**

g. (2 pts) Convert 101 000 010 from binary to base 8 **502**

h. (2 pts) Convert 60 from base 8 to base 2 **110 000**

i. (2 pts) Convert 3 from base 8 to base 2 **011**

j. (2 pts) Convert 555e from base 16 to base 2 **0101 0101 0101 1110**

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 banana grape date
```

a. (2 pts) What is the value of `argc` in this case? 4

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

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int g;  
    Node h;  
    double p;  
    char q;  
    int *r;  
    Node *s;  
    double *t;  
    char *w;  
  
    return 0;  
}
```

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

a. (2 pts) `s->next->next` **Node ***

b. (2 pts) `&s` **Node ****

c. (2 pts) `argv[0]` **char ***

d. (2 pts) `&q` **char ***

e. (2 pts) `s->data` **int**

f. (2 pts) `r` **int ***

g. (2 pts) `argc` **int**

h. (2 pts) `g` **int**

i. (2 pts) `argv[1][2]` **char**

j. (2 pts) `s->next` **Node ***

k. (2 pts) `*t` **double**

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 parameters 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 parameters 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;
}
```

Exam #336 Page: 8 Name: _____

End of Exam

total points=100

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 11 from base 8 to binary **001 001**

b. (2 pts) Convert 110 010 000 from binary to octal **620**

c. (2 pts) Convert 111 from base 10 to binary **0110 1111**

d. (2 pts) Convert 011 110 101 from base 2 to octal **365**

e. (2 pts) Convert 0011 1011 from binary to decimal **59**

f. (2 pts) Convert 111 001 100 from binary to octal **714**

g. (2 pts) Convert 010 010 101 from binary to base 8 **225**

h. (2 pts) Convert 22 from base 8 to base 2 **010 010**

i. (2 pts) Convert 110 110 000 from binary to octal **660**

j. (2 pts) Convert f148 from hexadecimal to base 2 **1111 0001 0100 1000**

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 date kiwi
```

a. (2 pts) What is the value of `argc` in this case? **4**

b. (2 pts) What is the value of `argv[1][0]`? **l**

c. (2 pts) What is the value of `argv[0][4]`? **n**

d. (2 pts) What is the value of `argv[2][3]`? **e**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[0]` **char ***

b. (2 pts) `h->next->next` **Node ***

c. (2 pts) `r` **char ***

d. (2 pts) `*r` **char**

e. (2 pts) `&r` **char ****

f. (2 pts) `&f` **double ***

g. (2 pts) `f` **double**

h. (2 pts) `argc` **int**

i. (2 pts) `argv[1][2]` **char**

j. (2 pts) `h->data` **int**

k. (2 pts) `h->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #337 Page: 8 Name: _____

End of Exam

total points=100

Exam #338 Page: 1 Name: _____

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 0101 0101 1010 1101 from binary to base 16 **55ad**

b. (2 pts) Convert 46 from base 8 to base 2 **100 110**

c. (2 pts) Convert 51 from base 8 to base 2 **101 001**

d. (2 pts) Convert 1101 0110 from base 2 to base 10 **214**

e. (2 pts) Convert 1001 0010 from binary to decimal **146**

f. (2 pts) Convert 24 from base 10 to base 2 **0001 1000**

g. (2 pts) Convert 010 011 010 from binary to octal **232**

h. (2 pts) Convert 179 from decimal to base 2 **1011 0011**

i. (2 pts) Convert 011 110 011 from binary to base 8 **363**

j. (2 pts) Convert 1d89 from base 16 to base 2 **0001 1101 1000 1001**

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 banana kiwi
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][3]`? **u**

c. (2 pts) What is the value of `argv[2][0]`? **k**

d. (2 pts) What is the value of `argv[1][1]`? **a**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node e;  
    double f;  
    int g;  
    char h;  
    Node *p;  
    double *q;  
    int *r;  
    char *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `&q` **double ****
- b. (2 pts) `p->next` **Node ***
- c. (2 pts) `p->next->next` **Node ***
- d. (2 pts) `*s` **char**
- e. (2 pts) `&f` **double ***
- f. (2 pts) `h` **char**
- g. (2 pts) `argc` **int**
- h. (2 pts) `argv[1][2]` **char**
- i. (2 pts) `r` **int ***
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `p->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #338 Page: 8 Name: _____

End of Exam

total points=100

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 100 110 101 from binary to base 8 **465**

b. (2 pts) Convert 83 from base 10 to binary **0101 0011**

c. (2 pts) Convert 44 from octal to base 2 **100 100**

d. (2 pts) Convert 0110 0010 from base 2 to decimal **98**

e. (2 pts) Convert 7562 from hexadecimal to base 2 **0111 0101 0110 0010**

f. (2 pts) Convert 178 from base 10 to binary **1011 0010**

g. (2 pts) Convert 111 101 110 from base 2 to octal **756**

h. (2 pts) Convert 59 from decimal to base 2 **0011 1011**

i. (2 pts) Convert 0100 0011 from binary to decimal **67**

j. (2 pts) Convert b973 from hexadecimal to base 2 **1011 1001 0111 0011**

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 grape lime lemon fig
```

- a. (2 pts) What is the value of `argc` in this case? **5**
- b. (2 pts) What is the value of `argv[2][1]`? **i**
- c. (2 pts) What is the value of `argv[0][5]`? **I**
- d. (2 pts) What is the value of `argv[1][2]`? **a**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int b;  
    double c;  
    Node d;  
    char e;  
    int *f;  
    double *g;  
    Node *h;  
    char *p;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[1][2]` **char**

b. (2 pts) `argc` **int**

c. (2 pts) `*f` **int**

d. (2 pts) `&b` **int ***

e. (2 pts) `c` **double**

f. (2 pts) `h->data` **int**

g. (2 pts) `h->next->next` **Node ***

h. (2 pts) `&p` **char ****

i. (2 pts) `h->next` **Node ***

j. (2 pts) `argv[0]` **char ***

k. (2 pts) `g` **double ***

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 parameters 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 parameters 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;
}
```

Exam #339 Page: 8 Name: _____

End of Exam

total points=100

Exam #340 Page: 1 Name: _____

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 binary to base 10 **242**

b. (2 pts) Convert 1111 0110 from binary to decimal **246**

c. (2 pts) Convert 001 110 010 from base 2 to octal **162**

d. (2 pts) Convert 7 from base 8 to base 2 **111**

e. (2 pts) Convert 229 from decimal to binary **1110 0101**

f. (2 pts) Convert 55 from octal to binary **101 101**

g. (2 pts) Convert 48c8 from base 16 to binary **0100 1000 1100 1000**

h. (2 pts) Convert 70 from octal to base 2 **111 000**

i. (2 pts) Convert 15 from base 8 to base 2 **001 101**

j. (2 pts) Convert 197 from decimal to binary **1100 0101**

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 guava cherry grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][0]`? g

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

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int s;  
    Node t;  
    double w;  
    char x;  
    int *y;  
    Node *z;  
    double *a;  
    char *b;  
  
    return 0;  
}
```

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

a. (2 pts) z->next->next **Node ***

b. (2 pts) argv[1][2] **char**

c. (2 pts) argc **int**

d. (2 pts) t **Node**

e. (2 pts) b **char ***

f. (2 pts) *y **int**

g. (2 pts) z->next **Node ***

h. (2 pts) z->data **int**

i. (2 pts) &a **double ****

j. (2 pts) &x **char ***

k. (2 pts) argv[0] **char ***

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 parameters 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 parameters 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;
}
```

Exam #340 Page: 8 Name: _____

End of Exam

total points=100

Exam #341 Page: 1 Name: _____

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 37c8 from hexadecimal to binary **0011 0111 1100 1000**

b. (2 pts) Convert 176 from decimal to base 2 **1011 0000**

c. (2 pts) Convert 0010 0111 0010 0101 from binary to hexadecimal **2725**

d. (2 pts) Convert aa3a from base 16 to base 2 **1010 1010 0011 1010**

e. (2 pts) Convert c926 from hexadecimal to base 2 **1100 1001 0010 0110**

f. (2 pts) Convert 0100 1101 1111 0010 from binary to hexadecimal **4df2**

g. (2 pts) Convert f2b8 from hexadecimal to binary **1111 0010 1011 1000**

h. (2 pts) Convert 011 010 011 from base 2 to base 8 **323**

i. (2 pts) Convert 0 from octal to binary **000**

j. (2 pts) Convert 96 from decimal to binary **0110 0000**

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 date mango cherry
```

a. (2 pts) What is the value of `argc` in this case? **4**

b. (2 pts) What is the value of `argv[0][1]`? **/**

c. (2 pts) What is the value of `argv[1][0]`? **d**

d. (2 pts) What is the value of `argv[2][4]`? **o**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int z;  
    Node a;  
    double b;  
    char c;  
    int *d;  
    Node *e;  
    double *f;  
    char *g;  
  
    return 0;  
}
```

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

- a. (2 pts) `&d` **int ****
- b. (2 pts) `e->next` **Node ***
- c. (2 pts) `*e` **Node**
- d. (2 pts) `argv[0]` **char ***
- e. (2 pts) `f` **double ***
- f. (2 pts) `argc` **int**
- g. (2 pts) `c` **char**
- h. (2 pts) `argv[1][2]` **char**
- i. (2 pts) `e->next->next` **Node ***
- j. (2 pts) `&b` **double ***
- k. (2 pts) `e->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #341 Page: 8 Name: _____

End of Exam

total points=100

Exam #342 Page: 1 Name: _____

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 105 from base 10 to binary **0110 1001**

b. (2 pts) Convert 824c from base 16 to base 2 **1000 0010 0100 1100**

c. (2 pts) Convert 93 from decimal to binary **0101 1101**

d. (2 pts) Convert 001 100 from base 2 to base 8 **14**

e. (2 pts) Convert 31 from decimal to base 2 **0001 1111**

f. (2 pts) Convert 1000 0000 from base 2 to base 10 **128**

g. (2 pts) Convert 1111 0100 from base 2 to base 10 **244**

h. (2 pts) Convert 210 from decimal to base 2 **1101 0010**

i. (2 pts) Convert 50 from base 8 to base 2 **101 000**

j. (2 pts) Convert 141 from decimal to base 2 **1000 1101**

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 mango lemon
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][2]`? **n**

c. (2 pts) What is the value of `argv[0][6]`? **t**

d. (2 pts) What is the value of `argv[2][2]`? **m**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double a;  
    Node b;  
    int c;  
    char d;  
    double *e;  
    Node *f;  
    int *g;  
    char *h;  
  
    return 0;  
}
```

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

a. (2 pts) `f->next->next` **Node ***

b. (2 pts) `&b` **Node ***

c. (2 pts) `argv[1][2]` **char**

d. (2 pts) `*f` **Node**

e. (2 pts) `argc` **int**

f. (2 pts) `f->next` **Node ***

g. (2 pts) `&f` **Node ****

h. (2 pts) `g` **int ***

i. (2 pts) `argv[0]` **char ***

j. (2 pts) `f->data` **int**

k. (2 pts) `a` **double**

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 parameters 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 parameters 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;
}
```

Exam #342 Page: 8 Name: _____

End of Exam

total points=100

Exam #343 Page: 1 Name: _____

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 1010 1110 0000 1001 from base 2 to base 16 **ae09**

b. (2 pts) Convert 0011 1100 0111 0101 from base 2 to base 16 **3c75**

c. (2 pts) Convert 010 010 110 from base 2 to octal **226**

d. (2 pts) Convert 44 from base 8 to binary **100 100**

e. (2 pts) Convert 0010 1001 0000 from binary to base 16 **290**

f. (2 pts) Convert 110 100 from base 2 to base 8 **64**

g. (2 pts) Convert 1001 1110 from binary to base 10 **158**

h. (2 pts) Convert 5a22 from hexadecimal to base 2 **0101 1010 0010 0010**

i. (2 pts) Convert 011 011 000 from binary to octal **330**

j. (2 pts) Convert 41 from base 10 to binary **0010 1001**

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 apple lemon cherry lime
```

a. (2 pts) What is the value of `argc` in this case? 5

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

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

d. (2 pts) What is the value of `argv[0][6]`? t

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double x;  
    int y;  
    Node z;  
    char a;  
    double *b;  
    int *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[1][2]` **char**
- b. (2 pts) `&x` **double ***
- c. (2 pts) `*d` **Node**
- d. (2 pts) `d->next` **Node ***
- e. (2 pts) `b` **double ***
- f. (2 pts) `d->next->next` **Node ***
- g. (2 pts) `d->data` **int**
- h. (2 pts) `argv[0]` **char ***
- i. (2 pts) `argc` **int**
- j. (2 pts) `z` **Node**
- k. (2 pts) `&b` **double ****

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 parameters 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 parameters 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;
}
```

Exam #343 Page: 8 Name: _____

End of Exam

total points=100

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 0110 0110 0110 from base 2 to base 16 **666**

b. (2 pts) Convert 223 from base 10 to binary **1101 1111**

c. (2 pts) Convert 1111 0000 from binary to base 10 **240**

d. (2 pts) Convert 77 from decimal to binary **0100 1101**

e. (2 pts) Convert 115 from decimal to binary **0111 0011**

f. (2 pts) Convert 27 from base 10 to base 2 **0001 1011**

g. (2 pts) Convert f08a from base 16 to base 2 **1111 0000 1000 1010**

h. (2 pts) Convert 0 from octal to base 2 **000**

i. (2 pts) Convert 010 111 101 from base 2 to base 8 **275**

j. (2 pts) Convert 15 from base 8 to base 2 **001 101**

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 lime kiwi mango date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][3]`? e

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double h;  
    int p;  
    Node q;  
    char r;  
    double *s;  
    int *t;  
    Node *w;  
    char *x;  
  
    return 0;  
}
```

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

a. (2 pts) `q`

Node

b. (2 pts) `w->next`

Node *

c. (2 pts) `*w`

Node

d. (2 pts) `x`

char *

e. (2 pts) `argv[0]`

char *

f. (2 pts) `&w`

Node **

g. (2 pts) `&h`

double *

h. (2 pts) `w->data`

int

i. (2 pts) `argv[1][2]`

char

j. (2 pts) `argc`

int

k. (2 pts) `w->next->next`

Node *

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 parameters 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 parameters 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;
}
```

Exam #344 Page: 8 Name: _____

End of Exam

total points=100

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 0100 1011 from binary to base 10 **75**

b. (2 pts) Convert 46 from octal to base 2 **100 110**

c. (2 pts) Convert 67 from base 8 to base 2 **110 111**

d. (2 pts) Convert d9e2 from hexadecimal to base 2 **1101 1001 1110 0010**

e. (2 pts) Convert 0101 0110 0101 0100 from binary to base 16 **5654**

f. (2 pts) Convert 1011 0101 from binary to decimal **181**

g. (2 pts) Convert 1001 1010 0111 1011 from binary to base 16 **9a7b**

h. (2 pts) Convert 136 from decimal to binary **1000 1000**

i. (2 pts) Convert 12 from octal to binary **001 010**

j. (2 pts) Convert 64 from base 8 to base 2 **110 100**

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 cherry grape kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][0]`? .

c. (2 pts) What is the value of `argv[1][3]`? r

d. (2 pts) What is the value of `argv[2][2]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int e;  
    Node f;  
    double g;  
    char h;  
    int *p;  
    Node *q;  
    double *r;  
    char *s;  
  
    return 0;  
}
```

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

a. (2 pts) `e`

int

b. (2 pts) `q->next`

Node *

c. (2 pts) `*s`

char

d. (2 pts) `r`

double *

e. (2 pts) `argv[1][2]`

char

f. (2 pts) `&q`

Node **

g. (2 pts) `q->data`

int

h. (2 pts) `q->next->next`

Node *

i. (2 pts) `argc`

int

j. (2 pts) `argv[0]`

char *

k. (2 pts) `&h`

char *

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 parameters 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 parameters 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;
}
```

Exam #345 Page: 8 Name: _____

End of Exam

total points=100

Exam #401 Page: 1 Name: _____

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 7cd2 from hexadecimal to binary **0111 1100 1101 0010**

b. (2 pts) Convert 0010 1001 1000 1101 from base 2 to hexadecimal **298d**

c. (2 pts) Convert 0111 from binary to base 10 **7**

d. (2 pts) Convert 138 from decimal to base 2 **1000 1010**

e. (2 pts) Convert 110 000 110 from binary to base 8 **606**

f. (2 pts) Convert 1000 0000 1101 0001 from binary to hexadecimal **80d1**

g. (2 pts) Convert d67f from base 16 to binary **1101 0110 0111 1111**

h. (2 pts) Convert 41 from base 8 to base 2 **100 001**

i. (2 pts) Convert 70 from decimal to base 2 **0100 0110**

j. (2 pts) Convert 0001 from binary to decimal **1**

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

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

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

```
./thing kiwi fig cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[2][1]`? i

c. (2 pts) What is the value of `argv[1][0]`? k

d. (2 pts) What is the value of `argv[0][6]`? g

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int b;  
    char c;  
    double d;  
    Node e;  
    int *f;  
    char *g;  
    double *h;  
    Node *p;  
  
    return 0;  
}
```

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

a. (2 pts) `p->next` **Node ***

b. (2 pts) `argc` **int**

c. (2 pts) `argv[1][2]` **char**

d. (2 pts) `&c` **char ***

e. (2 pts) `b` **int**

f. (2 pts) `f` **int ***

g. (2 pts) `argv[0]` **char ***

h. (2 pts) `p->data` **int**

i. (2 pts) `&p` **Node ****

j. (2 pts) `*p` **Node**

k. (2 pts) `p->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #401 Page: 8 Name: _____

End of Exam

total points=100

Exam #402 Page: 1 Name: _____

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 174 from base 10 to binary **1010 1110**

b. (2 pts) Convert fb12 from base 16 to base 2 **1111 1011 0001 0010**

c. (2 pts) Convert 0011 1101 1011 1010 from binary to base 16 **3dba**

d. (2 pts) Convert e72d from base 16 to base 2 **1110 0111 0010 1101**

e. (2 pts) Convert 0001 1001 from binary to base 10 **25**

f. (2 pts) Convert 1011 0011 from base 2 to base 10 **179**

g. (2 pts) Convert 110 110 001 from binary to base 8 **661**

h. (2 pts) Convert 73 from octal to base 2 **111 011**

i. (2 pts) Convert 1110 1000 from binary to base 10 **232**

j. (2 pts) Convert 45 from decimal to base 2 **0010 1101**

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

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

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

```
./thing fig mango
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][4]`? **i**

c. (2 pts) What is the value of `argv[1][0]`? **f**

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double b;  
    char c;  
    int d;  
    Node e;  
    double *f;  
    char *g;  
    int *h;  
    Node *p;  
  
    return 0;  
}
```

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

a. (2 pts) p->next->next **Node ***

b. (2 pts) argc **int**

c. (2 pts) p->data **int**

d. (2 pts) argv[1][2] **char**

e. (2 pts) &b **double ***

f. (2 pts) &g **char ****

g. (2 pts) *p **Node**

h. (2 pts) argv[0] **char ***

i. (2 pts) p->next **Node ***

j. (2 pts) p **Node ***

k. (2 pts) c **char**

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 parameters 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 parameters 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;
}
```

Exam #402 Page: 8 Name: _____

End of Exam

total points=100

Exam #403 Page: 1 Name: _____

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 0011 0001 0011 from base 2 to base 16 **f313**
- b. (2 pts) Convert 1011 0101 0011 1011 from binary to hexadecimal **b53b**
- c. (2 pts) Convert 12 from octal to base 2 **001 010**
- d. (2 pts) Convert 7317 from base 16 to binary **0111 0011 0001 0111**
- e. (2 pts) Convert fcdd from hexadecimal to base 2 **1111 1100 1101 1101**
- f. (2 pts) Convert 010 011 010 from base 2 to base 8 **232**
- g. (2 pts) Convert 829e from base 16 to binary **1000 0010 1001 1110**
- h. (2 pts) Convert 119 from decimal to base 2 **0111 0111**
- i. (2 pts) Convert 178 from base 10 to binary **1011 0010**
- j. (2 pts) Convert 201 from base 10 to base 2 **1100 1001**

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

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

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

```
./thing lemon guava cherry apple
```

a. (2 pts) What is the value of `argc` in this case? **5**

b. (2 pts) What is the value of `argv[1][4]`? **n**

c. (2 pts) What is the value of `argv[0][4]`? **i**

d. (2 pts) What is the value of `argv[2][3]`? **v**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double y;  
    int z;  
    char a;  
    Node b;  
    double *c;  
    int *d;  
    char *e;  
    Node *f;  
  
    return 0;  
}
```

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

a. (2 pts) `argc`

`int`

b. (2 pts) `e`

`char *`

c. (2 pts) `f->next->next`

`Node *`

d. (2 pts) `f->data`

`int`

e. (2 pts) `b`

`Node`

f. (2 pts) `argv[1][2]`

`char`

g. (2 pts) `f->next`

`Node *`

h. (2 pts) `&f`

`Node **`

i. (2 pts) `argv[0]`

`char *`

j. (2 pts) `&b`

`Node *`

k. (2 pts) `*c`

`double`

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 parameters 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 parameters 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;
}
```

Exam #403 Page: 8 Name: _____

End of Exam

total points=100

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 0100 1011 0111 0000 from base 2 to base 16 **4b70**

b. (2 pts) Convert 5807 from hexadecimal to binary **0101 1000 0000 0111**

c. (2 pts) Convert 1101 0000 1111 1100 from binary to hexadecimal **d0fc**

d. (2 pts) Convert 0010 1110 from binary to decimal **46**

e. (2 pts) Convert 33 from octal to binary **011 011**

f. (2 pts) Convert 78 from decimal to binary **0100 1110**

g. (2 pts) Convert d451 from hexadecimal to base 2 **1101 0100 0101 0001**

h. (2 pts) Convert 7 from octal to base 2 **111**

i. (2 pts) Convert 164 from base 10 to binary **1010 0100**

j. (2 pts) Convert 1101 0101 from binary to decimal **213**

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

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

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

```
./thing apple banana kiwi guava
```

a. (2 pts) What is the value of `argc` in this case? 5

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

c. (2 pts) What is the value of `argv[0][4]`? i

d. (2 pts) What is the value of `argv[1][0]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double p;  
    int q;  
    char r;  
    Node s;  
    double *t;  
    int *w;  
    char *x;  
    Node *y;  
  
    return 0;  
}
```

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

a. (2 pts) y->next->next **Node ***

b. (2 pts) argc **int**

c. (2 pts) *y **Node**

d. (2 pts) &x **char ****

e. (2 pts) y->data **int**

f. (2 pts) argv[1][2] **char**

g. (2 pts) y->next **Node ***

h. (2 pts) &s **Node ***

i. (2 pts) w **int ***

j. (2 pts) argv[0] **char ***

k. (2 pts) s **Node**

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 parameters 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 parameters 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;
}
```

Exam #404 Page: 8 Name: _____

End of Exam

total points=100

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 1001 0000 from base 2 to base 10 **144**

b. (2 pts) Convert 4 from octal to base 2 **100**

c. (2 pts) Convert 1011 1110 1101 1101 from binary to hexadecimal **bedd**

d. (2 pts) Convert 56 from octal to base 2 **101 110**

e. (2 pts) Convert 50a1 from hexadecimal to base 2 **0101 0000 1010 0001**

f. (2 pts) Convert 1110 1000 from binary to decimal **232**

g. (2 pts) Convert 7e41 from base 16 to base 2 **0111 1110 0100 0001**

h. (2 pts) Convert 51 from octal to binary **101 001**

i. (2 pts) Convert 0110 1111 from binary to base 10 **111**

j. (2 pts) Convert 0111 0001 from base 2 to base 10 **113**

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

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

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

```
./thing kiwi grape fig
```

a. (2 pts) What is the value of `argc` in this case? 4

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

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char f;  
    int g;  
    double h;  
    Node p;  
    char *q;  
    int *r;  
    double *s;  
    Node *t;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[1][2]` **char**
- b. (2 pts) `&h` **double ***
- c. (2 pts) `*r` **int**
- d. (2 pts) `t->data` **int**
- e. (2 pts) `q` **char ***
- f. (2 pts) `t->next->next` **Node ***
- g. (2 pts) `argc` **int**
- h. (2 pts) `argv[0]` **char ***
- i. (2 pts) `g` **int**
- j. (2 pts) `t->next` **Node ***
- k. (2 pts) `&q` **char ****

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 parameters 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 parameters 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;
}
```

Exam #405 Page: 8 Name: _____

End of Exam

total points=100

Exam #406 Page: 1 Name: _____

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 60 from octal to binary **110 000**

b. (2 pts) Convert 70 from base 8 to base 2 **111 000**

c. (2 pts) Convert 1111 0101 from binary to base 10 **245**

d. (2 pts) Convert 22 from decimal to binary **0001 0110**

e. (2 pts) Convert 166 from decimal to base 2 **1010 0110**

f. (2 pts) Convert 1aae from base 16 to base 2 **0001 1010 1010 1110**

g. (2 pts) Convert 8070 from hexadecimal to base 2 **1000 0000 0111 0000**

h. (2 pts) Convert 3 from base 8 to base 2 **011**

i. (2 pts) Convert 10cc from base 16 to binary **0001 0000 1100 1100**

j. (2 pts) Convert 1001 1101 from base 2 to decimal **157**

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

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

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

```
./thing mango cherry
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][2]`? **n**

c. (2 pts) What is the value of `argv[2][0]`? **c**

d. (2 pts) What is the value of `argv[0][3]`? **h**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char y;  
    double z;  
    int a;  
    Node b;  
    char *c;  
    double *d;  
    int *e;  
    Node *f;  
  
    return 0;  
}
```

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

a. (2 pts) `a`

`int`

b. (2 pts) `f->next`

`Node *`

c. (2 pts) `f->data`

`int`

d. (2 pts) `f->next->next`

`Node *`

e. (2 pts) `&d`

`double **`

f. (2 pts) `argv[0]`

`char *`

g. (2 pts) `argv[1][2]`

`char`

h. (2 pts) `f`

`Node *`

i. (2 pts) `&y`

`char *`

j. (2 pts) `argc`

`int`

k. (2 pts) `*d`

`double`

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 parameters 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 parameters 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;
}
```

Exam #406 Page: 8 Name: _____

End of Exam

total points=100

Exam #407 Page: 1 Name: _____

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 6 from decimal to binary **0110**

b. (2 pts) Convert 157 from decimal to binary **1001 1101**

c. (2 pts) Convert 70 from base 8 to binary **111 000**

d. (2 pts) Convert 101 000 101 from binary to base 8 **505**

e. (2 pts) Convert 8a0b from hexadecimal to binary **1000 1010 0000 1011**

f. (2 pts) Convert b48b from base 16 to base 2 **1011 0100 1000 1011**

g. (2 pts) Convert 2a60 from base 16 to base 2 **0010 1010 0110 0000**

h. (2 pts) Convert 149 from decimal to base 2 **1001 0101**

i. (2 pts) Convert 1101 1010 from binary to base 10 **218**

j. (2 pts) Convert 0011 1001 from base 2 to decimal **57**

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

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

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

```
./thing kiwi banana
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][1]`? /

c. (2 pts) What is the value of `argv[1][0]`? k

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char s;  
    double t;  
    int w;  
    Node x;  
    char *y;  
    double *z;  
    int *a;  
    Node *b;  
  
    return 0;  
}
```

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

a. (2 pts) s

char

b. (2 pts) argv[0]

char *

c. (2 pts) a

int *

d. (2 pts) *a

int

e. (2 pts) b->next

Node *

f. (2 pts) &x

Node *

g. (2 pts) b->data

int

h. (2 pts) argc

int

i. (2 pts) &b

Node **

j. (2 pts) b->next->next

Node *

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

char

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 parameters 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 parameters 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;
}
```

Exam #407 Page: 8 Name: _____

End of Exam

total points=100

Exam #408 Page: 1 Name: _____

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 94 from decimal to binary **0101 1110**

b. (2 pts) Convert 010 000 001 from binary to octal **201**

c. (2 pts) Convert 100 010 000 from binary to base 8 **420**

d. (2 pts) Convert 0101 1110 0101 0011 from binary to base 16 **5e53**

e. (2 pts) Convert 250 from decimal to base 2 **1111 1010**

f. (2 pts) Convert 101 101 011 from base 2 to base 8 **553**

g. (2 pts) Convert 011 111 000 from binary to octal **370**

h. (2 pts) Convert 17 from octal to binary **001 111**

i. (2 pts) Convert 205 from base 10 to binary **1100 1101**

j. (2 pts) Convert 010 001 001 from base 2 to base 8 **211**

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

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

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

```
./thing banana apple guava kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

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

c. (2 pts) What is the value of `argv[0][0]`? .

d. (2 pts) What is the value of `argv[1][0]`? b

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int f;  
    double g;  
    char h;  
    Node p;  
    int *q;  
    double *r;  
    char *s;  
    Node *t;  
  
    return 0;  
}
```

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

a. (2 pts) `&s` **char ****

b. (2 pts) `t->next->next` **Node ***

c. (2 pts) `t->data` **int**

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `argv[1][2]` **char**

f. (2 pts) `t->next` **Node ***

g. (2 pts) `f` **int**

h. (2 pts) `*r` **double**

i. (2 pts) `argc` **int**

j. (2 pts) `r` **double ***

k. (2 pts) `&p` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #408 Page: 8 Name: _____

End of Exam

total points=100

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 50 from octal to base 2 **101 000**

b. (2 pts) Convert 1111 1010 from binary to base 10 **250**

c. (2 pts) Convert 35 from base 8 to base 2 **011 101**

d. (2 pts) Convert 1110 1010 0011 1101 from binary to hexadecimal **ea3d**

e. (2 pts) Convert 1101 1101 1100 1111 from binary to hexadecimal **ddcf**

f. (2 pts) Convert 23 from octal to binary **010 011**

g. (2 pts) Convert 001 001 100 from base 2 to octal **114**

h. (2 pts) Convert 60 from octal to binary **110 000**

i. (2 pts) Convert 100 101 110 from binary to base 8 **456**

j. (2 pts) Convert 111 000 001 from binary to base 8 **701**

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

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

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

```
./thing grape mango date
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][1]`? /

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

d. (2 pts) What is the value of `argv[1][3]`? p

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int b;  
    char c;  
    double d;  
    Node e;  
    int *f;  
    char *g;  
    double *h;  
    Node *p;  
  
    return 0;  
}
```

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

a. (2 pts) `argv[0]` **char ***

b. (2 pts) `*p` **Node**

c. (2 pts) `p->next` **Node ***

d. (2 pts) `argv[1][2]` **char**

e. (2 pts) `&d` **double ***

f. (2 pts) `d` **double**

g. (2 pts) `p->next->next` **Node ***

h. (2 pts) `p` **Node ***

i. (2 pts) `p->data` **int**

j. (2 pts) `&f` **int ****

k. (2 pts) `argc` **int**

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 parameters 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 parameters 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;
}
```

Exam #409 Page: 8 Name: _____

End of Exam

total points=100

Exam #410 Page: 1 Name: _____

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 d52e from base 16 to binary **1101 0101 0010 1110**

b. (2 pts) Convert 110 011 000 from base 2 to octal **630**

c. (2 pts) Convert 172 from base 10 to binary **1010 1100**

d. (2 pts) Convert 21 from octal to binary **010 001**

e. (2 pts) Convert 14 from octal to binary **001 100**

f. (2 pts) Convert 130 from decimal to binary **1000 0010**

g. (2 pts) Convert 001 010 000 from binary to octal **120**

h. (2 pts) Convert 44 from base 10 to binary **0010 1100**

i. (2 pts) Convert 57 from base 10 to binary **0011 1001**

j. (2 pts) Convert 1101 from binary to base 10 **13**

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

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

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

```
./thing date fig
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][2]`? **t**

c. (2 pts) What is the value of `argv[2][0]`? **f**

d. (2 pts) What is the value of `argv[0][4]`? **i**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double s;  
    char t;  
    int w;  
    Node x;  
    double *y;  
    char *z;  
    int *a;  
    Node *b;  
  
    return 0;  
}
```

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

a. (2 pts) b->next **Node ***

b. (2 pts) b->next->next **Node ***

c. (2 pts) &t **char ***

d. (2 pts) b->data **int**

e. (2 pts) argc **int**

f. (2 pts) *a **int**

g. (2 pts) b **Node ***

h. (2 pts) &a **int ****

i. (2 pts) argv[1][2] **char**

j. (2 pts) x **Node**

k. (2 pts) argv[0] **char ***

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 parameters 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 parameters 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;
}
```

Exam #410 Page: 8 Name: _____

End of Exam

total points=100

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 110 100 from binary to octal **64**

b. (2 pts) Convert 8681 from base 16 to base 2 **1000 0110 1000 0001**

c. (2 pts) Convert 154 from decimal to binary **1001 1010**

d. (2 pts) Convert 110 100 100 from binary to base 8 **644**

e. (2 pts) Convert 101 110 from binary to octal **56**

f. (2 pts) Convert 0001 1011 from base 2 to base 10 **27**

g. (2 pts) Convert 110 100 100 from base 2 to octal **644**

h. (2 pts) Convert 180 from base 10 to binary **1011 0100**

i. (2 pts) Convert 0011 from binary to base 10 **3**

j. (2 pts) Convert 101 010 001 from binary to base 8 **521**

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

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

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

```
./thing grape banana
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][5]`? **n**

c. (2 pts) What is the value of `argv[1][4]`? **e**

d. (2 pts) What is the value of `argv[2][2]`? **n**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char p;  
    double q;  
    int r;  
    Node s;  
    char *t;  
    double *w;  
    int *x;  
    Node *y;  
  
    return 0;  
}
```

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

- a. (2 pts) &t **char ****
- b. (2 pts) x **int ***
- c. (2 pts) &p **char ***
- d. (2 pts) argv[1][2] **char**
- e. (2 pts) y->next **Node ***
- f. (2 pts) *t **char**
- g. (2 pts) argc **int**
- h. (2 pts) r **int**
- i. (2 pts) argv[0] **char ***
- j. (2 pts) y->next->next **Node ***
- k. (2 pts) y->data **int**

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 parameters 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 parameters 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;
}
```

Exam #411 Page: 8 Name: _____

End of Exam

total points=100

Exam #412 Page: 1 Name: _____

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 011 100 100 from base 2 to octal **344**

b. (2 pts) Convert 41 from base 10 to binary **0010 1001**

c. (2 pts) Convert 0011 1111 from base 2 to base 10 **63**

d. (2 pts) Convert 1000 1101 from base 2 to base 10 **141**

e. (2 pts) Convert 87b1 from hexadecimal to base 2 **1000 0111 1011 0001**

f. (2 pts) Convert 1d4f from base 16 to base 2 **0001 1101 0100 1111**

g. (2 pts) Convert 001 000 111 from binary to base 8 **107**

h. (2 pts) Convert 26 from base 8 to binary **010 110**

i. (2 pts) Convert 1111 0101 from base 2 to decimal **245**

j. (2 pts) Convert 55 from octal to binary **101 101**

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

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

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

```
./thing guava lemon kiwi lime
```

a. (2 pts) What is the value of `argc` in this case? **5**

b. (2 pts) What is the value of `argv[0][1]`? **/**

c. (2 pts) What is the value of `argv[2][0]`? **|**

d. (2 pts) What is the value of `argv[1][3]`? **v**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double b;  
    int c;  
    char d;  
    Node e;  
    double *f;  
    int *g;  
    char *h;  
    Node *p;  
  
    return 0;  
}
```

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

a. (2 pts) `*p` **Node**

b. (2 pts) `p->next->next` **Node ***

c. (2 pts) `p->data` **int**

d. (2 pts) `f` **double ***

e. (2 pts) `argv[0]` **char ***

f. (2 pts) `&b` **double ***

g. (2 pts) `c` **int**

h. (2 pts) `argv[1][2]` **char**

i. (2 pts) `p->next` **Node ***

j. (2 pts) `argc` **int**

k. (2 pts) `&p` **Node ****

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 parameters 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 parameters 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;
}
```

Exam #412 Page: 8 Name: _____

End of Exam

total points=100

Exam #413 Page: 1 Name: _____

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 183 from decimal to binary **1011 0111**

b. (2 pts) Convert 227 from decimal to base 2 **1110 0011**

c. (2 pts) Convert 0010 1101 from binary to decimal **45**

d. (2 pts) Convert 25 from base 10 to binary **0001 1001**

e. (2 pts) Convert 0110 1010 1111 1100 from binary to hexadecimal **6afc**

f. (2 pts) Convert 183 from decimal to base 2 **1011 0111**

g. (2 pts) Convert 205 from base 10 to binary **1100 1101**

h. (2 pts) Convert 226 from base 10 to binary **1110 0010**

i. (2 pts) Convert 191 from decimal to base 2 **1011 1111**

j. (2 pts) Convert 24 from base 8 to binary **010 100**

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

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

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

```
./thing date fig lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][1]`? a

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int y;  
    char z;  
    double a;  
    Node b;  
    int *c;  
    char *d;  
    double *e;  
    Node *f;  
  
    return 0;  
}
```

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

- a. (2 pts) `*d` **char**
- b. (2 pts) `f` **Node ***
- c. (2 pts) `f->next` **Node ***
- d. (2 pts) `f->data` **int**
- e. (2 pts) `f->next->next` **Node ***
- f. (2 pts) `argv[0]` **char ***
- g. (2 pts) `&b` **Node ***
- h. (2 pts) `&d` **char ****
- i. (2 pts) `b` **Node**
- j. (2 pts) `argc` **int**
- k. (2 pts) `argv[1][2]` **char**

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 parameters 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 parameters 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;
}
```

Exam #413 Page: 8 Name: _____

End of Exam

total points=100

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 1110 1000 from base 2 to base 10 **232**

b. (2 pts) Convert 1011 0100 1111 1100 from base 2 to hexadecimal **b4fc**

c. (2 pts) Convert 31 from octal to base 2 **011 001**

d. (2 pts) Convert 7627 from hexadecimal to base 2 **0111 0110 0010 0111**

e. (2 pts) Convert 60 from octal to binary **110 000**

f. (2 pts) Convert 72 from octal to binary **111 010**

g. (2 pts) Convert 110 011 111 from binary to base 8 **637**

h. (2 pts) Convert 75 from decimal to binary **0100 1011**

i. (2 pts) Convert 0110 0001 from base 2 to decimal **97**

j. (2 pts) Convert 011 111 001 from base 2 to octal **371**

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

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

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

```
./thing cherry apple
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][0]`? **c**

c. (2 pts) What is the value of `argv[0][6]`? **g**

d. (2 pts) What is the value of `argv[2][2]`? **p**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double y;  
    char z;  
    int a;  
    Node b;  
    double *c;  
    char *d;  
    int *e;  
    Node *f;  
  
    return 0;  
}
```

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

a. (2 pts) `argc`

`int`

b. (2 pts) `&f`

`Node **`

c. (2 pts) `*c`

`double`

d. (2 pts) `e`

`int *`

e. (2 pts) `f->next->next`

`Node *`

f. (2 pts) `argv[1][2]`

`char`

g. (2 pts) `y`

`double`

h. (2 pts) `argv[0]`

`char *`

i. (2 pts) `f->data`

`int`

j. (2 pts) `&z`

`char *`

k. (2 pts) `f->next`

`Node *`

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 parameters 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 parameters 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;
}
```

Exam #414 Page: 8 Name: _____

End of Exam

total points=100

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 2d8b from base 16 to binary **0010 1101 1000 1011**

b. (2 pts) Convert 33 from base 8 to base 2 **011 011**

c. (2 pts) Convert 521f from hexadecimal to base 2 **0101 0010 0001 1111**

d. (2 pts) Convert 211 from base 16 to binary **0010 0001 0001**

e. (2 pts) Convert 1010 0100 from binary to base 10 **164**

f. (2 pts) Convert 1000 0011 0111 0010 from base 2 to hexadecimal **8372**

g. (2 pts) Convert 011 110 011 from base 2 to base 8 **363**

h. (2 pts) Convert 1101 0011 0110 0000 from base 2 to hexadecimal **d360**

i. (2 pts) Convert 43 from decimal to base 2 **0010 1011**

j. (2 pts) Convert 110 001 from base 2 to octal **61**

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

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

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

```
./thing kiwi cherry
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][1]`? **i**

c. (2 pts) What is the value of `argv[0][3]`? **h**

d. (2 pts) What is the value of `argv[2][1]`? **h**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double e;  
    char f;  
    int g;  
    Node h;  
    double *p;  
    char *q;  
    int *r;  
    Node *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `s->data` `int`
- b. (2 pts) `&q` `char **`
- c. (2 pts) `argv[0]` `char *`
- d. (2 pts) `argc` `int`
- e. (2 pts) `s->next` `Node *`
- f. (2 pts) `q` `char *`
- g. (2 pts) `argv[1][2]` `char`
- h. (2 pts) `&e` `double *`
- i. (2 pts) `s->next->next` `Node *`
- j. (2 pts) `*r` `int`
- k. (2 pts) `h` `Node`

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 parameters 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 parameters 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;
}
```

Exam #415 Page: 8 Name: _____

End of Exam

total points=100

Exam #416 Page: 1 Name: _____

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 0010 0001 1101 0001 from binary to hexadecimal **21d1**

b. (2 pts) Convert 0101 0111 from base 2 to base 10 **87**

c. (2 pts) Convert 1011 0011 from base 2 to base 10 **179**

d. (2 pts) Convert 58 from decimal to binary **0011 1010**

e. (2 pts) Convert 0111 1110 from base 2 to decimal **126**

f. (2 pts) Convert 110 000 000 from base 2 to octal **600**

g. (2 pts) Convert 2 from octal to base 2 **010**

h. (2 pts) Convert 0110 1111 from base 2 to base 10 **111**

i. (2 pts) Convert 0110 0000 1111 0100 from binary to hexadecimal **60f4**

j. (2 pts) Convert 1010 0001 from binary to base 10 **161**

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

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

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

```
./thing banana lime guava kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][5]`? a

c. (2 pts) What is the value of `argv[2][3]`? e

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int a;  
    double b;  
    char c;  
    Node d;  
    int *e;  
    double *f;  
    char *g;  
    Node *h;  
  
    return 0;  
}
```

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

a. (2 pts) `h->next` **Node ***

b. (2 pts) `d` **Node**

c. (2 pts) `argv[0]` **char ***

d. (2 pts) `&f` **double ****

e. (2 pts) `&a` **int ***

f. (2 pts) `argv[1][2]` **char**

g. (2 pts) `h->next->next` **Node ***

h. (2 pts) `argc` **int**

i. (2 pts) `f` **double ***

j. (2 pts) `h->data` **int**

k. (2 pts) `*f` **double**

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 parameters 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 parameters 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;
}
```

Exam #416 Page: 8 Name: _____

End of Exam

total points=100

Exam #417 Page: 1 Name: _____

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 0110 0110 from binary to base 10 **102**

b. (2 pts) Convert 0001 0001 0011 0010 from binary to hexadecimal **1132**

c. (2 pts) Convert 50 from base 8 to base 2 **101 000**

d. (2 pts) Convert 198 from decimal to base 2 **1100 0110**

e. (2 pts) Convert 30 from base 8 to binary **011 000**

f. (2 pts) Convert 26 from base 8 to binary **010 110**

g. (2 pts) Convert 54 from base 8 to base 2 **101 100**

h. (2 pts) Convert 247 from base 10 to base 2 **1111 0111**

i. (2 pts) Convert 12 from base 8 to base 2 **001 010**

j. (2 pts) Convert 0011 1101 from base 2 to base 10 **61**

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

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

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

```
./thing date kiwi banana
```

a. (2 pts) What is the value of `argc` in this case? 4

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

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int x;  
    char y;  
    double z;  
    Node a;  
    int *b;  
    char *c;  
    double *d;  
    Node *e;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[0]` **char ***
- b. (2 pts) `&e` **Node ****
- c. (2 pts) `*d` **double**
- d. (2 pts) `e->next->next` **Node ***
- e. (2 pts) `argv[1][2]` **char**
- f. (2 pts) `argc` **int**
- g. (2 pts) `e->data` **int**
- h. (2 pts) `&a` **Node ***
- i. (2 pts) `y` **char**
- j. (2 pts) `b` **int ***
- k. (2 pts) `e->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #417 Page: 8 Name: _____

End of Exam

total points=100

Exam #418 Page: 1 Name: _____

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 46 from base 8 to base 2 **100 110**

b. (2 pts) Convert e2b8 from hexadecimal to base 2 **1110 0010 1011 1000**

c. (2 pts) Convert 215 from base 10 to binary **1101 0111**

d. (2 pts) Convert 2316 from hexadecimal to binary **0010 0011 0001 0110**

e. (2 pts) Convert 1011 0111 1011 0011 from base 2 to base 16 **b7b3**

f. (2 pts) Convert 140 from decimal to base 2 **1000 1100**

g. (2 pts) Convert b5cb from hexadecimal to base 2 **1011 0101 1100 1011**

h. (2 pts) Convert 0110 0000 0111 1100 from binary to base 16 **607c**

i. (2 pts) Convert 63 from base 8 to binary **110 011**

j. (2 pts) Convert 0110 1001 from base 2 to base 10 **105**

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

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

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

```
./thing grape mango
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[1][1]`? **r**

c. (2 pts) What is the value of `argv[0][4]`? **i**

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double y;  
    char z;  
    int a;  
    Node b;  
    double *c;  
    char *d;  
    int *e;  
    Node *f;  
  
    return 0;  
}
```

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

a. (2 pts) `&a` `int *`

b. (2 pts) `b` `Node`

c. (2 pts) `f` `Node *`

d. (2 pts) `&d` `char **`

e. (2 pts) `argc` `int`

f. (2 pts) `argv[0]` `char *`

g. (2 pts) `argv[1][2]` `char`

h. (2 pts) `f->next->next` `Node *`

i. (2 pts) `*e` `int`

j. (2 pts) `f->next` `Node *`

k. (2 pts) `f->data` `int`

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 parameters 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 parameters 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;
}
```

Exam #418 Page: 8 Name: _____

End of Exam

total points=100

Exam #419 Page: 1 Name: _____

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 220 from base 10 to binary **1101 1100**

b. (2 pts) Convert 100 111 001 from base 2 to octal **471**

c. (2 pts) Convert 110 001 011 from binary to octal **613**

d. (2 pts) Convert af00 from hexadecimal to base 2 **1010 1111 0000 0000**

e. (2 pts) Convert 46 from base 8 to base 2 **100 110**

f. (2 pts) Convert 0010 0110 from binary to base 10 **38**

g. (2 pts) Convert 27 from base 8 to base 2 **010 111**

h. (2 pts) Convert 1110 1000 0010 1101 from base 2 to base 16 **e82d**

i. (2 pts) Convert 45 from octal to base 2 **100 101**

j. (2 pts) Convert 0101 from base 2 to base 10 **5**

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

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

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

```
./thing guava lemon banana kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][6]`? g

c. (2 pts) What is the value of `argv[1][4]`? a

d. (2 pts) What is the value of `argv[2][4]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double d;  
    int e;  
    char f;  
    Node g;  
    double *h;  
    int *p;  
    char *q;  
    Node *r;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[1][2]` **char**
- b. (2 pts) `e` **int**
- c. (2 pts) `argv[0]` **char ***
- d. (2 pts) `r->data` **int**
- e. (2 pts) `argc` **int**
- f. (2 pts) `*r` **Node**
- g. (2 pts) `&e` **int ***
- h. (2 pts) `r->next->next` **Node ***
- i. (2 pts) `&r` **Node ****
- j. (2 pts) `r->next` **Node ***
- k. (2 pts) `q` **char ***

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 parameters 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 parameters 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;
}
```

Exam #419 Page: 8 Name: _____

End of Exam

total points=100

Exam #420 Page: 1 Name: _____

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 53 from base 10 to binary **0011 0101**

b. (2 pts) Convert 17 from base 8 to binary **001 111**

c. (2 pts) Convert 0110 1011 from base 2 to base 10 **107**

d. (2 pts) Convert 0110 1010 from binary to base 10 **106**

e. (2 pts) Convert 1011 0111 0111 from base 2 to hexadecimal **b77**

f. (2 pts) Convert 27d3 from base 16 to binary **0010 0111 1101 0011**

g. (2 pts) Convert 101 100 010 from binary to octal **542**

h. (2 pts) Convert 1000 1110 from base 2 to base 10 **142**

i. (2 pts) Convert 8950 from hexadecimal to binary **1000 1001 0101 0000**

j. (2 pts) Convert 100 010 from base 2 to octal **42**

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

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

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

```
./thing grape mango cherry fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][0]`? m

c. (2 pts) What is the value of `argv[0][6]`? g

d. (2 pts) What is the value of `argv[1][2]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double w;  
    int x;  
    char y;  
    Node z;  
    double *a;  
    int *b;  
    char *c;  
    Node *d;  
  
    return 0;  
}
```

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

- a. (2 pts) `d->next` **Node ***
- b. (2 pts) `&x` **int ***
- c. (2 pts) `d->next->next` **Node ***
- d. (2 pts) `d->data` **int**
- e. (2 pts) `argv[1][2]` **char**
- f. (2 pts) `a` **double ***
- g. (2 pts) `&c` **char ****
- h. (2 pts) `w` **double**
- i. (2 pts) `argc` **int**
- j. (2 pts) `*d` **Node**
- k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #420 Page: 8 Name: _____

End of Exam

total points=100

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 0111 1010 0010 1101 from base 2 to hexadecimal **7a2d**

b. (2 pts) Convert 76 from base 8 to base 2 **111 110**

c. (2 pts) Convert 88 from decimal to binary **0101 1000**

d. (2 pts) Convert 75 from octal to binary **111 101**

e. (2 pts) Convert 73 from base 8 to base 2 **111 011**

f. (2 pts) Convert 193 from base 10 to binary **1100 0001**

g. (2 pts) Convert 010 110 110 from base 2 to octal **266**

h. (2 pts) Convert 0001 0110 0100 1111 from base 2 to base 16 **164f**

i. (2 pts) Convert 0101 0011 0110 0111 from base 2 to base 16 **5367**

j. (2 pts) Convert 101 011 010 from binary to octal **532**

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

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

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

```
./thing lemon guava fig
```

a. (2 pts) What is the value of `argc` in this case? 4

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

c. (2 pts) What is the value of `argv[1][3]`? o

d. (2 pts) What is the value of `argv[0][6]`? g

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char r;  
    int s;  
    double t;  
    Node w;  
    char *x;  
    int *y;  
    double *z;  
    Node *a;  
  
    return 0;  
}
```

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

a. (2 pts) `*x`

char

b. (2 pts) `t`

double

c. (2 pts) `argv[1][2]`

char

d. (2 pts) `argv[0]`

char *

e. (2 pts) `a->data`

int

f. (2 pts) `&r`

char *

g. (2 pts) `a`

Node *

h. (2 pts) `a->next`

Node *

i. (2 pts) `argc`

int

j. (2 pts) `&x`

char **

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

Node *

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 parameters 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 parameters 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;
}
```

Exam #421 Page: 8 Name: _____

End of Exam

total points=100

Exam #422 Page: 1 Name: _____

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 ab8f from hexadecimal to base 2 **1010 1011 1000 1111**

b. (2 pts) Convert 1100 1011 from binary to base 10 **203**

c. (2 pts) Convert 43 from octal to base 2 **100 011**

d. (2 pts) Convert 010 100 101 from binary to base 8 **245**

e. (2 pts) Convert 010 001 001 from base 2 to base 8 **211**

f. (2 pts) Convert 1111 0100 0001 1001 from binary to base 16 **f419**

g. (2 pts) Convert 27 from octal to binary **010 111**

h. (2 pts) Convert 0111 1111 0011 1110 from binary to base 16 **7f3e**

i. (2 pts) Convert 1111 0101 0010 0010 from binary to hexadecimal **f522**

j. (2 pts) Convert 1101 1001 from binary to base 10 **217**

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

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

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

```
./thing lemon mango
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[1][2]`? m

c. (2 pts) What is the value of `argv[0][0]`? .

d. (2 pts) What is the value of `argv[2][1]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char s;  
    double t;  
    int w;  
    Node x;  
    char *y;  
    double *z;  
    int *a;  
    Node *b;  
  
    return 0;  
}
```

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

- a. (2 pts) `b->data` **int**
- b. (2 pts) `&z` **double ****
- c. (2 pts) `argv[0]` **char ***
- d. (2 pts) `argc` **int**
- e. (2 pts) `&x` **Node ***
- f. (2 pts) `b->next->next` **Node ***
- g. (2 pts) `*y` **char**
- h. (2 pts) `argv[1][2]` **char**
- i. (2 pts) `b->next` **Node ***
- j. (2 pts) `b` **Node ***
- k. (2 pts) `s` **char**

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 parameters 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 parameters 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;
}
```

Exam #422 Page: 8 Name: _____

End of Exam

total points=100

Exam #423 Page: 1 Name: _____

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 74 from base 8 to base 2 **111 100**

b. (2 pts) Convert 133 from decimal to base 2 **1000 0101**

c. (2 pts) Convert 7d2e from base 16 to base 2 **0111 1101 0010 1110**

d. (2 pts) Convert 1101 1110 from binary to decimal **222**

e. (2 pts) Convert 40 from base 10 to binary **0010 1000**

f. (2 pts) Convert 8df6 from hexadecimal to base 2 **1000 1101 1111 0110**

g. (2 pts) Convert 001 110 from base 2 to octal **16**

h. (2 pts) Convert 1 from octal to base 2 **001**

i. (2 pts) Convert 57 from base 8 to base 2 **101 111**

j. (2 pts) Convert 011 101 010 from binary to octal **352**

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

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

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

```
./thing date grape apple kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][4]`? i

c. (2 pts) What is the value of `argv[2][0]`? g

d. (2 pts) What is the value of `argv[1][0]`? d

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double p;  
    int q;  
    char r;  
    Node s;  
    double *t;  
    int *w;  
    char *x;  
    Node *y;  
  
    return 0;  
}
```

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

a. (2 pts) `argc`

int

b. (2 pts) `y->data`

int

c. (2 pts) `argv[1][2]`

char

d. (2 pts) `*w`

int

e. (2 pts) `r`

char

f. (2 pts) `y->next`

Node *

g. (2 pts) `x`

char *

h. (2 pts) `y->next->next`

Node *

i. (2 pts) `argv[0]`

char *

j. (2 pts) `&r`

char *

k. (2 pts) `&t`

double **

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 parameters 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 parameters 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;
}
```

Exam #423 Page: 8 Name: _____

End of Exam

total points=100

Exam #424 Page: 1 Name: _____

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 010 010 001 from binary to base 8 **221**

b. (2 pts) Convert 001 010 000 from base 2 to octal **120**

c. (2 pts) Convert 0010 0010 1001 0000 from binary to hexadecimal **2290**

d. (2 pts) Convert 1001 1010 0011 1100 from base 2 to base 16 **9a3c**

e. (2 pts) Convert 1001 1000 1010 0101 from base 2 to base 16 **98a5**

f. (2 pts) Convert 1000 1111 from binary to base 10 **143**

g. (2 pts) Convert 89 from decimal to base 2 **0101 1001**

h. (2 pts) Convert 1010 1101 from binary to decimal **173**

i. (2 pts) Convert 54 from octal to binary **101 100**

j. (2 pts) Convert 40 from octal to base 2 **100 000**

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

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

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

```
./thing lemon kiwi date apple
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][6]`? g

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

d. (2 pts) What is the value of `argv[1][1]`? e

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int b;  
    double c;  
    char d;  
    Node e;  
    int *f;  
    double *g;  
    char *h;  
    Node *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `p->data` **int**
- b. (2 pts) `&c` **double ***
- c. (2 pts) `c` **double**
- d. (2 pts) `argc` **int**
- e. (2 pts) `p->next->next` **Node ***
- f. (2 pts) `*f` **int**
- g. (2 pts) `argv[1][2]` **char**
- h. (2 pts) `&h` **char ****
- i. (2 pts) `f` **int ***
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `p->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #424 Page: 8 Name: _____

End of Exam

total points=100

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 141 from base 10 to binary **1000 1101**

b. (2 pts) Convert e279 from base 16 to base 2 **1110 0010 0111 1001**

c. (2 pts) Convert 16 from decimal to binary **0001 0000**

d. (2 pts) Convert 2626 from base 16 to base 2 **0010 0110 0010 0110**

e. (2 pts) Convert 011 110 111 from base 2 to base 8 **367**

f. (2 pts) Convert 001 010 010 from base 2 to octal **122**

g. (2 pts) Convert 3 from base 10 to binary **0011**

h. (2 pts) Convert 0011 0101 0001 0000 from base 2 to hexadecimal **3510**

i. (2 pts) Convert 7bc4 from base 16 to base 2 **0111 1011 1100 0100**

j. (2 pts) Convert 7 from base 8 to base 2 **111**

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

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

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

```
./thing grape mango lime
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][4]`? e

c. (2 pts) What is the value of `argv[0][6]`? g

d. (2 pts) What is the value of `argv[2][4]`? o

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char g;  
    int h;  
    double p;  
    Node q;  
    char *r;  
    int *s;  
    double *t;  
    Node *w;  
  
    return 0;  
}
```

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

- a. (2 pts) `*t` **double**
- b. (2 pts) `&s` **int ****
- c. (2 pts) `w->data` **int**
- d. (2 pts) `&g` **char ***
- e. (2 pts) `w` **Node ***
- f. (2 pts) `argv[0]` **char ***
- g. (2 pts) `argc` **int**
- h. (2 pts) `argv[1][2]` **char**
- i. (2 pts) `w->next->next` **Node ***
- j. (2 pts) `w->next` **Node ***
- k. (2 pts) `g` **char**

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 parameters 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 parameters 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;
}
```

Exam #425 Page: 8 Name: _____

End of Exam

total points=100

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 1011 1111 from binary to base 10 **191**

b. (2 pts) Convert 179 from decimal to binary **1011 0011**

c. (2 pts) Convert 010 001 101 from binary to octal **215**

d. (2 pts) Convert 100 000 100 from binary to base 8 **404**

e. (2 pts) Convert 1101 0010 from base 2 to decimal **210**

f. (2 pts) Convert 0101 1011 from base 2 to decimal **91**

g. (2 pts) Convert 5 from decimal to base 2 **0101**

h. (2 pts) Convert 1001 1101 1111 1111 from base 2 to hexadecimal **9dff**

i. (2 pts) Convert 1d7e from hexadecimal to binary **0001 1101 0111 1110**

j. (2 pts) Convert 010 010 010 from base 2 to octal **222**

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

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

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

```
./thing lemon kiwi
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][4]`? **i**

c. (2 pts) What is the value of `argv[1][3]`? **o**

d. (2 pts) What is the value of `argv[2][1]`? **i**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char h;  
    double p;  
    int q;  
    Node r;  
    char *s;  
    double *t;  
    int *w;  
    Node *x;  
  
    return 0;  
}
```

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

a. (2 pts) `x->next` **Node ***

b. (2 pts) `x->next->next` **Node ***

c. (2 pts) `p` **double**

d. (2 pts) `argv[1][2]` **char**

e. (2 pts) `x->data` **int**

f. (2 pts) `argv[0]` **char ***

g. (2 pts) `&w` **int ****

h. (2 pts) `w` **int ***

i. (2 pts) `*w` **int**

j. (2 pts) `argc` **int**

k. (2 pts) `&r` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #426 Page: 8 Name: _____

End of Exam

total points=100

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 3ec from hexadecimal to base 2 **0011 1110 1100**
- b. (2 pts) Convert 6e27 from hexadecimal to base 2 **0110 1110 0010 0111**
- c. (2 pts) Convert 0011 0100 1010 1111 from base 2 to hexadecimal **34af**
- d. (2 pts) Convert 1110 0101 0010 from binary to hexadecimal **e52**
- e. (2 pts) Convert b559 from base 16 to binary **1011 0101 0101 1001**
- f. (2 pts) Convert 245 from decimal to base 2 **1111 0101**
- g. (2 pts) Convert 175 from base 10 to base 2 **1010 1111**
- h. (2 pts) Convert 0010 0101 from binary to decimal **37**
- i. (2 pts) Convert 1110 0111 1001 0110 from base 2 to base 16 **e796**
- j. (2 pts) Convert 111 001 001 from base 2 to octal **711**

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

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

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

```
./thing fig guava
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[2][0]`? g

c. (2 pts) What is the value of `argv[0][2]`? t

d. (2 pts) What is the value of `argv[1][2]`? g

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char e;  
    double f;  
    int g;  
    Node h;  
    char *p;  
    double *q;  
    int *r;  
    Node *s;  
  
    return 0;  
}
```

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

a. (2 pts) `s->next->next` **Node ***

b. (2 pts) `&g` **int ***

c. (2 pts) `h` **Node**

d. (2 pts) `*s` **Node**

e. (2 pts) `q` **double ***

f. (2 pts) `s->next` **Node ***

g. (2 pts) `s->data` **int**

h. (2 pts) `argv[0]` **char ***

i. (2 pts) `&p` **char ****

j. (2 pts) `argv[1][2]` **char**

k. (2 pts) `argc` **int**

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 parameters 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 parameters 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;
}
```

Exam #427 Page: 8 Name: _____

End of Exam

total points=100

Exam #428 Page: 1 Name: _____

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 5c48 from hexadecimal to base 2 **0101 1100 0100 1000**

b. (2 pts) Convert 16 from decimal to binary **0001 0000**

c. (2 pts) Convert 66 from octal to base 2 **110 110**

d. (2 pts) Convert 201 from base 10 to binary **1100 1001**

e. (2 pts) Convert 11 from base 8 to base 2 **001 001**

f. (2 pts) Convert 1111 0110 1011 1010 from binary to base 16 **f6ba**

g. (2 pts) Convert 0 from octal to binary **000**

h. (2 pts) Convert 1100 1100 0010 0001 from binary to hexadecimal **cc21**

i. (2 pts) Convert 1101 1010 0000 1001 from binary to hexadecimal **da09**

j. (2 pts) Convert 74 from base 8 to base 2 **111 100**

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

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

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

```
./thing banana mango fig date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[2][4]`? o

c. (2 pts) What is the value of `argv[1][3]`? a

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int x;  
    double y;  
    char z;  
    Node a;  
    int *b;  
    double *c;  
    char *d;  
    Node *e;  
  
    return 0;  
}
```

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

- a. (2 pts) `e->next` **Node ***
- b. (2 pts) `z` **char**
- c. (2 pts) `argv[1][2]` **char**
- d. (2 pts) `&e` **Node ****
- e. (2 pts) `argv[0]` **char ***
- f. (2 pts) `e->data` **int**
- g. (2 pts) `&z` **char ***
- h. (2 pts) `argc` **int**
- i. (2 pts) `*d` **char**
- j. (2 pts) `e->next->next` **Node ***
- k. (2 pts) `b` **int ***

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 parameters 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 parameters 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;
}
```

Exam #428 Page: 8 Name: _____

End of Exam

total points=100

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 101 000 010 from binary to base 8 **502**

b. (2 pts) Convert 1100 1011 0001 1100 from base 2 to base 16 **cb1c**

c. (2 pts) Convert 61 from base 8 to base 2 **110 001**

d. (2 pts) Convert 85 from base 10 to base 2 **0101 0101**

e. (2 pts) Convert 1001 from base 2 to decimal **9**

f. (2 pts) Convert 9097 from base 16 to base 2 **1001 0000 1001 0111**

g. (2 pts) Convert 52 from base 8 to base 2 **101 010**

h. (2 pts) Convert 0101 0011 1101 0010 from binary to hexadecimal **53d2**

i. (2 pts) Convert a420 from base 16 to binary **1010 0100 0010 0000**

j. (2 pts) Convert 43 from octal to base 2 **100 011**

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

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

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

```
./thing lemon apple banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][0]`? l

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int s;  
    char t;  
    double w;  
    Node x;  
    int *y;  
    char *z;  
    double *a;  
    Node *b;  
  
    return 0;  
}
```

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

a. (2 pts) `b->next->next` **Node ***

b. (2 pts) `t` **char**

c. (2 pts) `argv[0]` **char ***

d. (2 pts) `b->next` **Node ***

e. (2 pts) `argv[1][2]` **char**

f. (2 pts) `b` **Node ***

g. (2 pts) `argc` **int**

h. (2 pts) `b->data` **int**

i. (2 pts) `&z` **char ****

j. (2 pts) `&t` **char ***

k. (2 pts) `*y` **int**

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 parameters 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 parameters 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;
}
```

Exam #429 Page: 8 Name: _____

End of Exam

total points=100

Exam #430 Page: 1 Name: _____

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 64 from base 8 to binary **110 100**

b. (2 pts) Convert 1001 1100 1010 0001 from base 2 to hexadecimal **9ca1**

c. (2 pts) Convert 1111 1110 from base 2 to base 10 **254**

d. (2 pts) Convert b210 from base 16 to binary **1011 0010 0001 0000**

e. (2 pts) Convert 95 from base 10 to binary **0101 1111**

f. (2 pts) Convert 60 from base 8 to binary **110 000**

g. (2 pts) Convert 173 from decimal to binary **1010 1101**

h. (2 pts) Convert 57 from base 8 to binary **101 111**

i. (2 pts) Convert 0100 0101 1101 1011 from binary to hexadecimal **45db**

j. (2 pts) Convert 56 from base 8 to base 2 **101 110**

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

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

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

```
./thing guava apple
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[0][1]`? /

c. (2 pts) What is the value of `argv[1][4]`? a

d. (2 pts) What is the value of `argv[2][2]`? p

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double d;  
    char e;  
    int f;  
    Node g;  
    double *h;  
    char *p;  
    int *q;  
    Node *r;  
  
    return 0;  
}
```

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

a. (2 pts) r->next->next **Node ***

b. (2 pts) r->data **int**

c. (2 pts) q **int ***

d. (2 pts) &d **double ***

e. (2 pts) r->next **Node ***

f. (2 pts) argc **int**

g. (2 pts) &r **Node ****

h. (2 pts) *r **Node**

i. (2 pts) argv[1][2] **char**

j. (2 pts) f **int**

k. (2 pts) argv[0] **char ***

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 parameters 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 parameters 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;
}
```

Exam #430 Page: 8 Name: _____

End of Exam

total points=100

Exam #431 Page: 1 Name: _____

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 0001 0111 from base 2 to base 10 **23**

b. (2 pts) Convert 0101 0110 from binary to base 10 **86**

c. (2 pts) Convert 1110 1100 from binary to base 10 **236**

d. (2 pts) Convert 3dfb from base 16 to base 2 **0011 1101 1111 1011**

e. (2 pts) Convert 0100 0010 1000 0111 from base 2 to base 16 **4287**

f. (2 pts) Convert 27 from octal to binary **010 111**

g. (2 pts) Convert 5703 from base 16 to binary **0101 0111 0000 0011**

h. (2 pts) Convert 0100 0100 from base 2 to decimal **68**

i. (2 pts) Convert ff2 from base 16 to base 2 **1111 1111 0010**

j. (2 pts) Convert 25 from octal to base 2 **010 101**

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

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

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

```
./thing guava grape
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[1][2]`? a

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

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double a;  
    char b;  
    int c;  
    Node d;  
    double *e;  
    char *f;  
    int *g;  
    Node *h;  
  
    return 0;  
}
```

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

a. (2 pts) `&f` **char ****

b. (2 pts) `h->next` **Node ***

c. (2 pts) `f` **char ***

d. (2 pts) `&d` **Node ***

e. (2 pts) `*f` **char**

f. (2 pts) `argv[1][2]` **char**

g. (2 pts) `argc` **int**

h. (2 pts) `a` **double**

i. (2 pts) `argv[0]` **char ***

j. (2 pts) `h->next->next` **Node ***

k. (2 pts) `h->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #431 Page: 8 Name: _____

End of Exam

total points=100

Exam #432 Page: 1 Name: _____

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 1101 0011 from binary to decimal **211**

b. (2 pts) Convert 55 from octal to binary **101 101**

c. (2 pts) Convert 65 from octal to base 2 **110 101**

d. (2 pts) Convert 0111 1100 0100 0111 from binary to hexadecimal **7c47**

e. (2 pts) Convert 73 from base 10 to binary **0100 1001**

f. (2 pts) Convert 0010 0010 1001 0001 from binary to base 16 **2291**

g. (2 pts) Convert 6aa2 from base 16 to base 2 **0110 1010 1010 0010**

h. (2 pts) Convert 1111 0100 1101 0111 from base 2 to base 16 **f4d7**

i. (2 pts) Convert 57 from octal to base 2 **101 111**

j. (2 pts) Convert 111 000 101 from base 2 to octal **705**

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

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

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

```
./thing fig lemon apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[2][0]`? l

c. (2 pts) What is the value of `argv[0][6]`? g

d. (2 pts) What is the value of `argv[1][0]`? f

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int h;  
    char p;  
    double q;  
    Node r;  
    int *s;  
    char *t;  
    double *w;  
    Node *x;  
  
    return 0;  
}
```

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

a. (2 pts) `x->data` **int**

b. (2 pts) `x->next->next` **Node ***

c. (2 pts) `r` **Node**

d. (2 pts) `argv[1][2]` **char**

e. (2 pts) `w` **double ***

f. (2 pts) `argc` **int**

g. (2 pts) `x->next` **Node ***

h. (2 pts) `&w` **double ****

i. (2 pts) `&q` **double ***

j. (2 pts) `*t` **char**

k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #432 Page: 8 Name: _____

End of Exam

total points=100

Exam #433 Page: 1 Name: _____

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 18bb from base 16 to binary **0001 1000 1011 1011**

b. (2 pts) Convert 110 from base 10 to base 2 **0110 1110**

c. (2 pts) Convert c367 from base 16 to binary **1100 0011 0110 0111**

d. (2 pts) Convert 1000 0011 0001 from base 2 to base 16 **831**

e. (2 pts) Convert 0010 1100 1101 1111 from base 2 to hexadecimal **2cdf**

f. (2 pts) Convert bc6e from hexadecimal to binary **1011 1100 0110 1110**

g. (2 pts) Convert 5 from octal to base 2 **101**

h. (2 pts) Convert 37 from base 8 to base 2 **011 111**

i. (2 pts) Convert 100 010 011 from binary to octal **423**

j. (2 pts) Convert 011 111 101 from base 2 to octal **375**

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

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

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

```
./thing apple cherry fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[2][5]`? y

c. (2 pts) What is the value of `argv[0][1]`? /

d. (2 pts) What is the value of `argv[1][0]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char e;  
    int f;  
    double g;  
    Node h;  
    char *p;  
    int *q;  
    double *r;  
    Node *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[0]` **char ***
- b. (2 pts) `s->next` **Node ***
- c. (2 pts) `argv[1][2]` **char**
- d. (2 pts) `&q` **int ****
- e. (2 pts) `q` **int ***
- f. (2 pts) `s->next->next` **Node ***
- g. (2 pts) `*r` **double**
- h. (2 pts) `f` **int**
- i. (2 pts) `&f` **int ***
- j. (2 pts) `argc` **int**
- k. (2 pts) `s->data` **int**

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 parameters 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 parameters 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;
}
```

Exam #433 Page: 8 Name: _____

End of Exam

total points=100

Exam #434 Page: 1 Name: _____

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 74 from decimal to base 2 **0100 1010**

b. (2 pts) Convert 0100 0000 from base 2 to decimal **64**

c. (2 pts) Convert 1111 1001 from base 2 to base 10 **249**

d. (2 pts) Convert 31 from octal to binary **011 001**

e. (2 pts) Convert 40 from base 8 to base 2 **100 000**

f. (2 pts) Convert 73 from base 8 to binary **111 011**

g. (2 pts) Convert 16c1 from base 16 to binary **0001 0110 1100 0001**

h. (2 pts) Convert 71 from octal to base 2 **111 001**

i. (2 pts) Convert 12 from octal to base 2 **001 010**

j. (2 pts) Convert 101 010 110 from base 2 to base 8 **526**

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

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

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

```
./thing mango kiwi
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of `argv[1][1]`? a

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

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char e;  
    double f;  
    int g;  
    Node h;  
    char *p;  
    double *q;  
    int *r;  
    Node *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `q` `double *`
- b. (2 pts) `g` `int`
- c. (2 pts) `*r` `int`
- d. (2 pts) `argv[0]` `char *`
- e. (2 pts) `s->next` `Node *`
- f. (2 pts) `&r` `int **`
- g. (2 pts) `argv[1][2]` `char`
- h. (2 pts) `s->data` `int`
- i. (2 pts) `argc` `int`
- j. (2 pts) `&e` `char *`
- k. (2 pts) `s->next->next` `Node *`

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 parameters 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 parameters 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;
}
```

Exam #434 Page: 8 Name: _____

End of Exam

total points=100

Exam #435 Page: 1 Name: _____

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 1000 1110 1111 1101 from binary to base 16 **8efd**

b. (2 pts) Convert 1111 1010 0101 0100 from binary to base 16 **fa54**

c. (2 pts) Convert 71 from base 8 to base 2 **111 001**

d. (2 pts) Convert 1111 0000 0101 1101 from binary to base 16 **f05d**

e. (2 pts) Convert 011 001 100 from base 2 to base 8 **314**

f. (2 pts) Convert 42 from base 8 to base 2 **100 010**

g. (2 pts) Convert c0b2 from hexadecimal to binary **1100 0000 1011 0010**

h. (2 pts) Convert 109 from decimal to base 2 **0110 1101**

i. (2 pts) Convert 111 101 010 from base 2 to octal **752**

j. (2 pts) Convert 010 001 101 from binary to octal **215**

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

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

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

```
./thing banana grape cherry date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[0][0]`? .

c. (2 pts) What is the value of `argv[2][3]`? p

d. (2 pts) What is the value of `argv[1][2]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int b;  
    double c;  
    char d;  
    Node e;  
    int *f;  
    double *g;  
    char *h;  
    Node *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `p` **Node ***
- b. (2 pts) `b` **int**
- c. (2 pts) `&e` **Node ***
- d. (2 pts) `argc` **int**
- e. (2 pts) `p->next` **Node ***
- f. (2 pts) `p->data` **int**
- g. (2 pts) `argv[0]` **char ***
- h. (2 pts) `argv[1][2]` **char**
- i. (2 pts) `p->next->next` **Node ***
- j. (2 pts) `*p` **Node**
- k. (2 pts) `&f` **int ****

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 parameters 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 parameters 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;
}
```

Exam #435 Page: 8 Name: _____

End of Exam

total points=100

Exam #436 Page: 1 Name: _____

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 71 from octal to base 2 **111 001**

b. (2 pts) Convert 9d20 from hexadecimal to binary **1001 1101 0010 0000**

c. (2 pts) Convert 100 011 010 from base 2 to octal **432**

d. (2 pts) Convert 1010 1011 from base 2 to base 10 **171**

e. (2 pts) Convert 65 from base 8 to binary **110 101**

f. (2 pts) Convert 100 010 100 from binary to base 8 **424**

g. (2 pts) Convert 100 100 from base 2 to octal **44**

h. (2 pts) Convert 0001 0011 1001 1001 from binary to base 16 **1399**

i. (2 pts) Convert 1110 0111 1101 1110 from base 2 to base 16 **e7de**

j. (2 pts) Convert 24 from base 8 to base 2 **010 100**

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

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

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

```
./thing apple banana date
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][2]`? t

c. (2 pts) What is the value of `argv[2][5]`? a

d. (2 pts) What is the value of `argv[1][2]`? p

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int s;  
    char t;  
    double w;  
    Node x;  
    int *y;  
    char *z;  
    double *a;  
    Node *b;  
  
    return 0;  
}
```

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

- a. (2 pts) `argc` **int**
- b. (2 pts) `&x` **Node ***
- c. (2 pts) `&b` **Node ****
- d. (2 pts) `s` **int**
- e. (2 pts) `*b` **Node**
- f. (2 pts) `b->data` **int**
- g. (2 pts) `a` **double ***
- h. (2 pts) `argv[0]` **char ***
- i. (2 pts) `argv[1][2]` **char**
- j. (2 pts) `b->next->next` **Node ***
- k. (2 pts) `b->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #436 Page: 8 Name: _____

End of Exam

total points=100

Exam #437 Page: 1 Name: _____

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 0010 1100 from binary to decimal **44**

b. (2 pts) Convert 5749 from hexadecimal to base 2 **0101 0111 0100 1001**

c. (2 pts) Convert 011 110 101 from base 2 to octal **365**

d. (2 pts) Convert 55 from base 10 to binary **0011 0111**

e. (2 pts) Convert 1011 1010 from base 2 to base 10 **186**

f. (2 pts) Convert 10 from octal to base 2 **001 000**

g. (2 pts) Convert 101 111 000 from binary to octal **570**

h. (2 pts) Convert 46 from base 8 to base 2 **100 110**

i. (2 pts) Convert 54 from octal to binary **101 100**

j. (2 pts) Convert 73 from octal to base 2 **111 011**

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

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

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

```
./thing lime mango guava
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[1][1]`? i

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

d. (2 pts) What is the value of `argv[0][1]`? /

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int a;  
    char b;  
    double c;  
    Node d;  
    int *e;  
    char *f;  
    double *g;  
    Node *h;  
  
    return 0;  
}
```

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

- a. (2 pts) `f` **char ***
- b. (2 pts) `*e` **int**
- c. (2 pts) `h->next` **Node ***
- d. (2 pts) `h->data` **int**
- e. (2 pts) `argc` **int**
- f. (2 pts) `&c` **double ***
- g. (2 pts) `argv[1][2]` **char**
- h. (2 pts) `&f` **char ****
- i. (2 pts) `argv[0]` **char ***
- j. (2 pts) `h->next->next` **Node ***
- k. (2 pts) `c` **double**

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 parameters 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 parameters 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;
}
```

Exam #437 Page: 8 Name: _____

End of Exam

total points=100

Exam #438 Page: 1 Name: _____

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 010 111 011 from base 2 to base 8 **273**

b. (2 pts) Convert 28ce from base 16 to binary **0010 1000 1100 1110**

c. (2 pts) Convert 1011 0001 0100 1000 from base 2 to hexadecimal **b148**

d. (2 pts) Convert 941c from base 16 to binary **1001 0100 0001 1100**

e. (2 pts) Convert 100 000 from base 2 to base 8 **40**

f. (2 pts) Convert 86 from base 10 to binary **0101 0110**

g. (2 pts) Convert 57 from base 8 to base 2 **101 111**

h. (2 pts) Convert 0100 from binary to base 10 **4**

i. (2 pts) Convert 24 from base 8 to base 2 **010 100**

j. (2 pts) Convert 6 from octal to binary **110**

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

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

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

```
./thing date guava
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][0]`? **.**

c. (2 pts) What is the value of `argv[1][3]`? **e**

d. (2 pts) What is the value of `argv[2][3]`? **v**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double b;  
    char c;  
    int d;  
    Node e;  
    double *f;  
    char *g;  
    int *h;  
    Node *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[0]` **char ***
- b. (2 pts) `&b` **double ***
- c. (2 pts) `p->data` **int**
- d. (2 pts) `&p` **Node ****
- e. (2 pts) `*f` **double**
- f. (2 pts) `e` **Node**
- g. (2 pts) `p->next` **Node ***
- h. (2 pts) `argc` **int**
- i. (2 pts) `p->next->next` **Node ***
- j. (2 pts) `argv[1][2]` **char**
- k. (2 pts) `f` **double ***

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 parameters 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 parameters 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;
}
```

Exam #438 Page: 8 Name: _____

End of Exam

total points=100

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 162 from base 10 to base 2 **1010 0010**

b. (2 pts) Convert 111 000 101 from base 2 to base 8 **705**

c. (2 pts) Convert 159 from decimal to binary **1001 1111**

d. (2 pts) Convert 2006 from base 16 to base 2 **0010 0000 0000 0110**

e. (2 pts) Convert f376 from base 16 to binary **1111 0011 0111 0110**

f. (2 pts) Convert 1111 0000 from base 2 to decimal **240**

g. (2 pts) Convert 32 from octal to binary **011 010**

h. (2 pts) Convert 139 from decimal to base 2 **1000 1011**

i. (2 pts) Convert 111 011 from binary to octal **73**

j. (2 pts) Convert 55 from base 8 to binary **101 101**

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

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

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

```
./thing date grape apple fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][1]`? a

c. (2 pts) What is the value of `argv[0][0]`? .

d. (2 pts) What is the value of `argv[2][4]`? e

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double y;  
    int z;  
    char a;  
    Node b;  
    double *c;  
    int *d;  
    char *e;  
    Node *f;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[1][2]` **char**
- b. (2 pts) `&d` **int ****
- c. (2 pts) `&y` **double ***
- d. (2 pts) `f->next->next` **Node ***
- e. (2 pts) `f->data` **int**
- f. (2 pts) `f->next` **Node ***
- g. (2 pts) `*d` **int**
- h. (2 pts) `f` **Node ***
- i. (2 pts) `z` **int**
- j. (2 pts) `argv[0]` **char ***
- k. (2 pts) `argc` **int**

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 parameters 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 parameters 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;
}
```

Exam #439 Page: 8 Name: _____

End of Exam

total points=100

Exam #440 Page: 1 Name: _____

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 250 from base 10 to base 2 **1111 1010**

b. (2 pts) Convert 1000 0101 1100 0011 from binary to base 16 **85c3**

c. (2 pts) Convert 448a from base 16 to base 2 **0100 0100 1000 1010**

d. (2 pts) Convert 1101 1011 from base 2 to base 10 **219**

e. (2 pts) Convert 011 000 111 from base 2 to base 8 **307**

f. (2 pts) Convert 74 from base 8 to base 2 **111 100**

g. (2 pts) Convert 186 from base 10 to base 2 **1011 1010**

h. (2 pts) Convert 14 from octal to base 2 **001 100**

i. (2 pts) Convert 100 000 from base 2 to base 8 **40**

j. (2 pts) Convert 1100 0010 0011 0001 from binary to base 16 **c231**

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

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

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

```
./thing date mango cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][0]`? .

c. (2 pts) What is the value of `argv[2][0]`? m

d. (2 pts) What is the value of `argv[1][1]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char p;  
    int q;  
    double r;  
    Node s;  
    char *t;  
    int *w;  
    double *x;  
    Node *y;  
  
    return 0;  
}
```

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

a. (2 pts) `y->next` **Node ***

b. (2 pts) `argc` **int**

c. (2 pts) `y->data` **int**

d. (2 pts) `x` **double ***

e. (2 pts) `argv[0]` **char ***

f. (2 pts) `*w` **int**

g. (2 pts) `argv[1][2]` **char**

h. (2 pts) `q` **int**

i. (2 pts) `&s` **Node ***

j. (2 pts) `&t` **char ****

k. (2 pts) `y->next->next` **Node ***

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 parameters 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 parameters 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;
}
```

Exam #440 Page: 8 Name: _____

End of Exam

total points=100

Exam #441 Page: 1 Name: _____

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 17 from base 8 to base 2 **001 111**

b. (2 pts) Convert 17 from octal to binary **001 111**

c. (2 pts) Convert 0011 0010 from binary to base 10 **50**

d. (2 pts) Convert 31 from base 8 to binary **011 001**

e. (2 pts) Convert 0100 0111 from base 2 to base 10 **71**

f. (2 pts) Convert 1000 1011 0101 0101 from binary to base 16 **8b55**

g. (2 pts) Convert 100 from decimal to base 2 **0110 0100**

h. (2 pts) Convert 1011 1010 from binary to base 10 **186**

i. (2 pts) Convert 1101 1010 0101 0010 from binary to base 16 **da52**

j. (2 pts) Convert 0101 1110 0001 1011 from base 2 to hexadecimal **5e1b**

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

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

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

```
./thing kiwi lime grape
```

a. (2 pts) What is the value of `argc` in this case? **4**

b. (2 pts) What is the value of `argv[1][0]`? **k**

c. (2 pts) What is the value of `argv[2][3]`? **e**

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

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char e;  
    int f;  
    double g;  
    Node h;  
    char *p;  
    int *q;  
    double *r;  
    Node *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `s->data` **int**
- b. (2 pts) `s->next` **Node ***
- c. (2 pts) `argv[1][2]` **char**
- d. (2 pts) `p` **char ***
- e. (2 pts) `argv[0]` **char ***
- f. (2 pts) `*r` **double**
- g. (2 pts) `&r` **double ****
- h. (2 pts) `argc` **int**
- i. (2 pts) `h` **Node**
- j. (2 pts) `s->next->next` **Node ***
- k. (2 pts) `&g` **double ***

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 parameters 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 parameters 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;
}
```

Exam #441 Page: 8 Name: _____

End of Exam

total points=100

Exam #442 Page: 1 Name: _____

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 0111 0001 0001 1000 from binary to hexadecimal **7118**

b. (2 pts) Convert 100 010 from base 2 to octal **42**

c. (2 pts) Convert 0110 1000 1100 1001 from binary to base 16 **68c9**

d. (2 pts) Convert 110 000 111 from base 2 to base 8 **607**

e. (2 pts) Convert 157 from base 10 to binary **1001 1101**

f. (2 pts) Convert 1011 1101 from binary to decimal **189**

g. (2 pts) Convert 011 001 100 from base 2 to octal **314**

h. (2 pts) Convert 0010 0010 from base 2 to base 10 **34**

i. (2 pts) Convert 011 111 000 from base 2 to base 8 **370**

j. (2 pts) Convert 42 from base 8 to base 2 **100 010**

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

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

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

```
./thing mango kiwi
```

a. (2 pts) What is the value of `argc` in this case? **3**

b. (2 pts) What is the value of `argv[0][4]`? **i**

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

d. (2 pts) What is the value of `argv[2][1]`? **i**

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char x;  
    double y;  
    int z;  
    Node a;  
    char *b;  
    double *c;  
    int *d;  
    Node *e;  
  
    return 0;  
}
```

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

a. (2 pts) `e->next->next` **Node ***

b. (2 pts) `e->next` **Node ***

c. (2 pts) `x` **char**

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `argc` **int**

f. (2 pts) `&y` **double ***

g. (2 pts) `b` **char ***

h. (2 pts) `argv[1][2]` **char**

i. (2 pts) `*d` **int**

j. (2 pts) `e->data` **int**

k. (2 pts) `&e` **Node ****

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 parameters 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 parameters 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;
}
```

Exam #442 Page: 8 Name: _____

End of Exam

total points=100

Exam #443 Page: 1 Name: _____

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 101 101 011 from base 2 to base 8 **553**

b. (2 pts) Convert 1100 1011 from base 2 to decimal **203**

c. (2 pts) Convert 56a9 from base 16 to base 2 **0101 0110 1010 1001**

d. (2 pts) Convert 010 011 111 from base 2 to octal **237**

e. (2 pts) Convert 1000 0000 1010 0100 from base 2 to hexadecimal **80a4**

f. (2 pts) Convert 010 101 111 from base 2 to octal **257**

g. (2 pts) Convert 100 000 from binary to base 8 **40**

h. (2 pts) Convert 1010 1010 1010 1010 from binary to base 16 **aaaa**

i. (2 pts) Convert 21 from octal to binary **010 001**

j. (2 pts) Convert 11 from octal to base 2 **001 001**

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

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

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

```
./thing lemon grape apple banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of `argv[1][1]`? e

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

d. (2 pts) What is the value of `argv[2][2]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double s;  
    int t;  
    char w;  
    Node x;  
    double *y;  
    int *z;  
    char *a;  
    Node *b;  
  
    return 0;  
}
```

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

- a. (2 pts) `&s` **double ***
- b. (2 pts) `argv[1][2]` **char**
- c. (2 pts) `argc` **int**
- d. (2 pts) `b->data` **int**
- e. (2 pts) `x` **Node**
- f. (2 pts) `b->next->next` **Node ***
- g. (2 pts) `b->next` **Node ***
- h. (2 pts) `*b` **Node**
- i. (2 pts) `&a` **char ****
- j. (2 pts) `b` **Node ***
- k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #443 Page: 8 Name: _____

End of Exam

total points=100

Exam #444 Page: 1 Name: _____

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 011 100 from binary to base 8 **34**

b. (2 pts) Convert 011 011 100 from base 2 to base 8 **334**

c. (2 pts) Convert 1111 1100 0000 1011 from binary to hexadecimal **fc0b**

d. (2 pts) Convert 1011 0100 0010 from binary to hexadecimal **b42**

e. (2 pts) Convert 111 100 010 from base 2 to base 8 **742**

f. (2 pts) Convert 0101 1000 from base 2 to decimal **88**

g. (2 pts) Convert 61ea from hexadecimal to binary **0110 0001 1110 1010**

h. (2 pts) Convert 24 from base 8 to base 2 **010 100**

i. (2 pts) Convert 16 from base 8 to binary **001 110**

j. (2 pts) Convert 0011 0001 1101 1010 from base 2 to hexadecimal **31da**

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

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

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

```
./thing guava fig apple kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

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

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

d. (2 pts) What is the value of `argv[1][4]`? a

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int e;  
    double f;  
    char g;  
    Node h;  
    int *p;  
    double *q;  
    char *r;  
    Node *s;  
  
    return 0;  
}
```

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

a. (2 pts) `s->next` **Node ***

b. (2 pts) `g` **char**

c. (2 pts) `*s` **Node**

d. (2 pts) `argv[0]` **char ***

e. (2 pts) `q` **double ***

f. (2 pts) `&e` **int ***

g. (2 pts) `s->data` **int**

h. (2 pts) `s->next->next` **Node ***

i. (2 pts) `argv[1][2]` **char**

j. (2 pts) `argc` **int**

k. (2 pts) `&p` **int ****

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 parameters 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 parameters 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;
}
```

Exam #444 Page: 8 Name: _____

End of Exam

total points=100

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 5333 from base 16 to base 2 **0101 0011 0011 0011**

b. (2 pts) Convert 40 from decimal to binary **0010 1000**

c. (2 pts) Convert 1110 1001 1110 1100 from base 2 to hexadecimal **e9ec**

d. (2 pts) Convert 1001 0111 0010 1100 from binary to base 16 **972c**

e. (2 pts) Convert 212 from base 10 to binary **1101 0100**

f. (2 pts) Convert 1111 0010 from binary to base 10 **242**

g. (2 pts) Convert bda from base 16 to binary **1011 1101 1010**

h. (2 pts) Convert 1101 1000 from binary to base 10 **216**

i. (2 pts) Convert 0010 1010 1110 from binary to base 16 **2ae**

j. (2 pts) Convert 1100 1101 from binary to decimal **205**

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

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

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

```
./thing mango kiwi cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of `argv[0][4]`? i

c. (2 pts) What is the value of `argv[2][0]`? k

d. (2 pts) What is the value of `argv[1][2]`? n

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    char b;  
    int c;  
    double d;  
    Node e;  
    char *f;  
    int *g;  
    double *h;  
    Node *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `argc` **int**
- b. (2 pts) `p->data` **int**
- c. (2 pts) `*f` **char**
- d. (2 pts) `p->next` **Node ***
- e. (2 pts) `f` **char ***
- f. (2 pts) `&p` **Node ****
- g. (2 pts) `&e` **Node ***
- h. (2 pts) `b` **char**
- i. (2 pts) `p->next->next` **Node ***
- j. (2 pts) `argv[1][2]` **char**
- k. (2 pts) `argv[0]` **char ***

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 parameters 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 parameters 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;
}
```

Exam #445 Page: 8 Name: _____

End of Exam

total points=100