

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
- b. (2 pts) Convert 67 from base 8 to binary
- c. (2 pts) Convert 1011 1000 from base 2 to base 10
- d. (2 pts) Convert 4ae8 from base 16 to base 2
- e. (2 pts) Convert 101 011 101 from binary to octal
- f. (2 pts) Convert 7f0a from hexadecimal to base 2
- g. (2 pts) Convert 101 000 110 from base 2 to octal
- h. (2 pts) Convert 13 from octal to base 2
- i. (2 pts) Convert 53 from base 8 to binary
- j. (2 pts) Convert 819f from base 16 to binary

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?
- b. (2 pts) What is the value of `argv[1][1]`?
- c. (2 pts) What is the value of `argv[0][6]`?
- d. (2 pts) What is the value of `argv[2][2]`?

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

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;
}
```

**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 010 000 100 from binary to base 8
- b. (2 pts) Convert 1011 0001 0000 0110 from base 2 to base 16
- c. (2 pts) Convert 1110 1110 1000 0001 from base 2 to hexadecimal
- d. (2 pts) Convert 1010 0111 0010 1001 from binary to hexadecimal
- e. (2 pts) Convert 0100 from binary to decimal
- f. (2 pts) Convert 101 100 010 from base 2 to base 8
- g. (2 pts) Convert 1010 0101 0110 0001 from binary to base 16
- h. (2 pts) Convert 100 101 001 from base 2 to base 8
- i. (2 pts) Convert 0101 0001 from binary to base 10
- j. (2 pts) Convert 1010 1101 1101 1111 from binary to hexadecimal

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?
- b. (2 pts) What is the value of `argv[0][6]`?
- c. (2 pts) What is the value of `argv[2][3]`?
- d. (2 pts) What is the value of `argv[1][1]`?

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`
- b. (2 pts) `&t`
- c. (2 pts) `r`
- d. (2 pts) `x`
- e. (2 pts) `argc`
- f. (2 pts) `y->next->next`
- g. (2 pts) `y->next`
- h. (2 pts) `*w`
- i. (2 pts) `argv[0]`
- j. (2 pts) `&p`
- k. (2 pts) `argv[1][2]`

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;
}
```

**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 870f from base 16 to binary
- b. (2 pts) Convert 0110 1011 from binary to decimal
- c. (2 pts) Convert 220 from base 10 to binary
- d. (2 pts) Convert 0011 0011 0001 0100 from binary to base 16
- e. (2 pts) Convert 232 from decimal to binary
- f. (2 pts) Convert 22 from octal to binary
- g. (2 pts) Convert 0100 1111 0101 0010 from base 2 to base 16
- h. (2 pts) Convert 28 from base 10 to base 2
- i. (2 pts) Convert 6 from base 8 to binary
- j. (2 pts) Convert 0100 1001 1100 1010 from base 2 to hexadecimal

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?
- b. (2 pts) What is the value of `argv[2][4]`?
- c. (2 pts) What is the value of `argv[0][0]`?
- d. (2 pts) What is the value of `argv[1][4]`?

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

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

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

Write the full function definition for a function that would have the following prototype. The 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;
}
```

**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 67 from decimal to base 2
- b. (2 pts) Convert 62 from octal to base 2
- c. (2 pts) Convert 1100 0101 from binary to base 10
- d. (2 pts) Convert 34 from octal to binary
- e. (2 pts) Convert 1110 1111 from binary to decimal
- f. (2 pts) Convert 17 from decimal to base 2
- g. (2 pts) Convert 0110 0010 1111 0001 from base 2 to hexadecimal
- h. (2 pts) Convert cd05 from hexadecimal to binary
- i. (2 pts) Convert 1100 1011 from base 2 to base 10
- j. (2 pts) Convert 66 from octal to base 2

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?
- b. (2 pts) What is the value of `argv[1][3]`?
- c. (2 pts) What is the value of `argv[2][3]`?
- d. (2 pts) What is the value of `argv[0][6]`?

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

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;
}
```

**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
- b. (2 pts) Convert 131 from decimal to binary
- c. (2 pts) Convert 1011 0011 from binary to base 10
- d. (2 pts) Convert 77 from octal to base 2
- e. (2 pts) Convert d28d from hexadecimal to binary
- f. (2 pts) Convert aae1 from hexadecimal to base 2
- g. (2 pts) Convert 1100 1110 0010 from binary to hexadecimal
- h. (2 pts) Convert 010 101 001 from base 2 to octal
- i. (2 pts) Convert 149 from decimal to binary
- j. (2 pts) Convert 0111 0011 1111 1011 from binary to base 16

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?
- b. (2 pts) What is the value of `argv[2][4]`?
- c. (2 pts) What is the value of `argv[1][1]`?
- d. (2 pts) What is the value of `argv[0][3]`?

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]`
- b. (2 pts) `argc`
- c. (2 pts) `*t`
- d. (2 pts) `argv[0]`
- e. (2 pts) `x`
- f. (2 pts) `&x`
- g. (2 pts) `&r`
- h. (2 pts) `y->next->next`
- i. (2 pts) `y->data`
- j. (2 pts) `y->next`
- k. (2 pts) `q`

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;
}
```

**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: \_\_\_\_\_@ uemail.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
- b. (2 pts) Convert 0101 0100 from base 2 to decimal
- c. (2 pts) Convert 72 from base 8 to base 2
- d. (2 pts) Convert 010 110 011 from binary to octal
- e. (2 pts) Convert 0010 1000 1010 1011 from binary to hexadecimal
- f. (2 pts) Convert 1101 1101 0100 1011 from base 2 to hexadecimal
- g. (2 pts) Convert 1111 0001 0000 from base 2 to base 16
- h. (2 pts) Convert 101 111 011 from binary to octal
- i. (2 pts) Convert 54 from decimal to base 2
- j. (2 pts) Convert 50 from octal to binary

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

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

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;
}
```

**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 111 111 101 from base 2 to octal
- b. (2 pts) Convert 1110 1011 1000 from base 2 to hexadecimal
- c. (2 pts) Convert d7d7 from base 16 to base 2
- d. (2 pts) Convert 71 from octal to base 2
- e. (2 pts) Convert 2 from octal to base 2
- f. (2 pts) Convert 7727 from base 16 to binary
- g. (2 pts) Convert 1011 1001 0000 0001 from binary to hexadecimal
- h. (2 pts) Convert 4555 from hexadecimal to binary
- i. (2 pts) Convert fa from hexadecimal to binary
- j. (2 pts) Convert 17 from base 8 to base 2

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

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

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;
}
```

**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 010 101 101 from base 2 to octal
  - b. (2 pts) Convert b184 from hexadecimal to base 2
  - c. (2 pts) Convert 0111 1101 0011 1001 from binary to hexadecimal
  - d. (2 pts) Convert 161 from decimal to base 2
  - e. (2 pts) Convert 124 from decimal to binary
  - f. (2 pts) Convert 0111 1000 from binary to decimal
  - g. (2 pts) Convert 1010 from binary to base 10
  - h. (2 pts) Convert 72 from octal to binary
  - i. (2 pts) Convert f36d from base 16 to base 2
  - j. (2 pts) Convert 0100 0111 1011 1011 from base 2 to hexadecimal

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?
- b. (2 pts) What is the value of `argv[2][2]`?
- c. (2 pts) What is the value of `argv[1][4]`?
- d. (2 pts) What is the value of `argv[0][2]`?

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`
- b. (2 pts) `x`
- c. (2 pts) `&s`
- d. (2 pts) `&t`
- e. (2 pts) `argv[1][2]`
- f. (2 pts) `argc`
- g. (2 pts) `*t`
- h. (2 pts) `y->next`
- i. (2 pts) `argv[0]`
- j. (2 pts) `p`
- k. (2 pts) `y->data`

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;
}
```

**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 9bde from base 16 to base 2
- b. (2 pts) Convert 6bad from hexadecimal to binary
- c. (2 pts) Convert 107 from base 10 to binary
- d. (2 pts) Convert 2cf3 from hexadecimal to binary
- e. (2 pts) Convert 5fbb from hexadecimal to base 2
- f. (2 pts) Convert 100 100 from binary to base 8
- g. (2 pts) Convert 1011 0100 from base 2 to base 10
- h. (2 pts) Convert 011 100 110 from binary to octal
- i. (2 pts) Convert 1011 1101 from binary to base 10
- j. (2 pts) Convert 1110 0011 1010 0101 from base 2 to hexadecimal

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?
- b. (2 pts) What is the value of `argv[1][1]`?
- c. (2 pts) What is the value of `argv[0][1]`?
- d. (2 pts) What is the value of `argv[2][3]`?

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

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;
}
```

**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 1100 1101 from binary to decimal
- b. (2 pts) Convert 3d33 from base 16 to base 2
- c. (2 pts) Convert 50 from base 8 to base 2
- d. (2 pts) Convert 1000 1001 0011 0101 from binary to base 16
- e. (2 pts) Convert 1011 0101 1101 1001 from binary to base 16
- f. (2 pts) Convert 68 from base 10 to base 2
- g. (2 pts) Convert 101 101 101 from base 2 to octal
- h. (2 pts) Convert 220 from base 10 to binary
- i. (2 pts) Convert 0101 1111 from base 2 to base 10
- j. (2 pts) Convert 1111 1110 0110 from base 2 to base 16

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?
- b. (2 pts) What is the value of `argv[0][4]`?
- c. (2 pts) What is the value of `argv[2][1]`?
- d. (2 pts) What is the value of `argv[1][1]`?

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

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;
}
```

**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 0010 0010 0000 from base 2 to hexadecimal
- b. (2 pts) Convert 111 101 110 from base 2 to octal
- c. (2 pts) Convert 1000 1111 0101 1000 from binary to hexadecimal
- d. (2 pts) Convert 0001 0101 0001 1111 from binary to hexadecimal
- e. (2 pts) Convert 100 110 010 from binary to base 8
- f. (2 pts) Convert 222 from base 10 to binary
- g. (2 pts) Convert 011 000 001 from base 2 to base 8
- h. (2 pts) Convert 6416 from hexadecimal to binary
- i. (2 pts) Convert 41 from decimal to binary
- j. (2 pts) Convert 1010 1011 1101 0000 from binary to base 16

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?
- b. (2 pts) What is the value of `argv[2][3]`?
- c. (2 pts) What is the value of `argv[0][2]`?
- d. (2 pts) What is the value of `argv[1][0]`?

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

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;
}
```

**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 1010 0111 1100 from binary to hexadecimal

b. (2 pts) Convert 1001 1010 0010 1000 from base 2 to base 16

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

d. (2 pts) Convert 208 from base 10 to base 2

e. (2 pts) Convert 2 from octal to binary

f. (2 pts) Convert dfcb from base 16 to binary

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

h. (2 pts) Convert 010 101 from binary to octal

i. (2 pts) Convert 27 from base 10 to binary

j. (2 pts) Convert 1011 0111 from base 2 to decimal

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?
- b. (2 pts) What is the value of `argv[2][1]`?
- c. (2 pts) What is the value of `argv[1][5]`?
- d. (2 pts) What is the value of `argv[0][3]`?

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`
- b. (2 pts) `argv[1][2]`
- c. (2 pts) `&q`
- d. (2 pts) `argv[0]`
- e. (2 pts) `s->next->next`
- f. (2 pts) `&e`
- g. (2 pts) `*r`
- h. (2 pts) `s->next`
- i. (2 pts) `r`
- j. (2 pts) `argc`
- k. (2 pts) `f`

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;
}
```

**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 1111 from base 2 to decimal
- b. (2 pts) Convert 25 from octal to binary
- c. (2 pts) Convert 10 from octal to base 2
- d. (2 pts) Convert 010 111 001 from base 2 to base 8
- e. (2 pts) Convert 111 011 001 from binary to base 8
- f. (2 pts) Convert 0111 1001 1100 1000 from binary to hexadecimal
- g. (2 pts) Convert 5c67 from base 16 to binary
- h. (2 pts) Convert 146 from decimal to binary
- i. (2 pts) Convert 1110 0101 from binary to base 10
- j. (2 pts) Convert 0101 0011 from binary to base 10

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?
- b. (2 pts) What is the value of `argv[1][3]`?
- c. (2 pts) What is the value of `argv[0][4]`?
- d. (2 pts) What is the value of `argv[2][0]`?

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

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;
}
```

**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
- b. (2 pts) Convert 001 001 011 from base 2 to base 8
- c. (2 pts) Convert 010 110 001 from binary to octal
- d. (2 pts) Convert b8de from hexadecimal to base 2
- e. (2 pts) Convert 20 from octal to base 2
- f. (2 pts) Convert 53 from octal to base 2
- g. (2 pts) Convert 0101 1110 from binary to base 10
- h. (2 pts) Convert 251 from base 10 to binary
- i. (2 pts) Convert 135 from base 10 to binary
- j. (2 pts) Convert 7f8f from hexadecimal to binary

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

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

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;
}
```

**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
  
- b. (2 pts) Convert 1101 1111 from base 2 to decimal
  
- c. (2 pts) Convert 010 001 101 from binary to octal
  
- d. (2 pts) Convert 0100 0100 from base 2 to decimal
  
- e. (2 pts) Convert 2651 from hexadecimal to binary
  
- f. (2 pts) Convert 0100 0110 0000 1111 from base 2 to hexadecimal
  
- g. (2 pts) Convert 1000 from base 2 to decimal
  
- h. (2 pts) Convert 82d8 from base 16 to binary
  
- i. (2 pts) Convert 0101 0001 from binary to base 10
  
- j. (2 pts) Convert 1b7a from base 16 to binary

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

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

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

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

Write the full function definition for a function that would have the following prototype. The 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;
}
```

**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 125 from decimal to binary
- b. (2 pts) Convert 100 000 101 from base 2 to octal
- c. (2 pts) Convert 1110 1100 from binary to base 10
- d. (2 pts) Convert 0 from base 8 to binary
- e. (2 pts) Convert 45 from octal to base 2
- f. (2 pts) Convert 71 from base 10 to binary
- g. (2 pts) Convert 0101 1010 0011 1001 from binary to base 16
- h. (2 pts) Convert 001 010 010 from base 2 to octal
- i. (2 pts) Convert 0100 0100 from base 2 to decimal
- j. (2 pts) Convert 001 001 110 from binary to base 8

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?
- b. (2 pts) What is the value of `argv[1][0]`?
- c. (2 pts) What is the value of `argv[2][1]`?
- 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`
- b. (2 pts) `h->data`
- c. (2 pts) `c`
- d. (2 pts) `h->next`
- e. (2 pts) `argv[0]`
- f. (2 pts) `argc`
- g. (2 pts) `g`
- h. (2 pts) `&g`
- i. (2 pts) `h->next->next`
- j. (2 pts) `argv[1][2]`
- k. (2 pts) `*e`

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;
}
```

**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 1100 0010 1101 1001 from binary to hexadecimal

b. (2 pts) Convert 60 from decimal to base 2

c. (2 pts) Convert 110 110 100 from binary to octal

d. (2 pts) Convert 43 from base 8 to base 2

e. (2 pts) Convert 0111 1010 from binary to decimal

f. (2 pts) Convert 1110 0001 from binary to decimal

g. (2 pts) Convert 0100 0010 1001 from base 2 to base 16

h. (2 pts) Convert 176 from decimal to binary

i. (2 pts) Convert 14 from decimal to base 2

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

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?
- b. (2 pts) What is the value of `argv[2][5]`?
- c. (2 pts) What is the value of `argv[1][1]`?
- d. (2 pts) What is the value of `argv[0][4]`?

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

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;
}
```

**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 f43b from base 16 to base 2
- b. (2 pts) Convert 14 from decimal to base 2
- c. (2 pts) Convert 107b from hexadecimal to base 2
- d. (2 pts) Convert 111 010 001 from base 2 to octal
- e. (2 pts) Convert 208 from decimal to binary
- f. (2 pts) Convert 0001 0011 1010 0101 from binary to base 16
- g. (2 pts) Convert 658 from hexadecimal to base 2
- h. (2 pts) Convert 19e9 from hexadecimal to binary
- i. (2 pts) Convert 1010 1111 from base 2 to decimal
- j. (2 pts) Convert 1110 1111 from base 2 to base 10

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?
- b. (2 pts) What is the value of `argv[0][6]`?
- c. (2 pts) What is the value of `argv[1][2]`?
- d. (2 pts) What is the value of `argv[2][0]`?

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

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;
}
```

**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 16 from base 8 to base 2
- b. (2 pts) Convert 62 from base 8 to base 2
- c. (2 pts) Convert 1111 1110 from base 2 to base 10
- d. (2 pts) Convert 35 from base 8 to base 2
- e. (2 pts) Convert 1011 0011 0111 1111 from binary to hexadecimal
- f. (2 pts) Convert ad82 from base 16 to base 2
- g. (2 pts) Convert 1011 0000 0100 1000 from base 2 to base 16
- h. (2 pts) Convert 50 from base 8 to binary
- i. (2 pts) Convert 011 110 100 from binary to base 8
- j. (2 pts) Convert 1000 1011 from binary to base 10

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?
- b. (2 pts) What is the value of `argv[1][4]`?
- c. (2 pts) What is the value of `argv[2][1]`?
- d. (2 pts) What is the value of `argv[0][2]`?

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

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;
}
```

**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 001 011 010 from binary to octal
- b. (2 pts) Convert 1011 0000 from base 2 to base 10
- c. (2 pts) Convert 27 from base 8 to base 2
- d. (2 pts) Convert 53 from octal to base 2
- e. (2 pts) Convert 1000 1101 0110 0011 from binary to base 16
- f. (2 pts) Convert ea6f from base 16 to base 2
- g. (2 pts) Convert 010 000 000 from base 2 to base 8
- h. (2 pts) Convert 62 from base 10 to binary
- i. (2 pts) Convert 1010 1111 0110 0110 from binary to hexadecimal
- j. (2 pts) Convert 0001 0100 0000 0100 from binary to hexadecimal

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?
- b. (2 pts) What is the value of `argv[2][2]`?
- c. (2 pts) What is the value of `argv[1][5]`?
- d. (2 pts) What is the value of `argv[0][4]`?

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

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;
}
```

**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
  - b. (2 pts) Convert 106 from base 10 to binary
  - c. (2 pts) Convert 4daa from hexadecimal to base 2
  - d. (2 pts) Convert 3934 from hexadecimal to binary
  - e. (2 pts) Convert 34 from octal to binary
  - f. (2 pts) Convert 132 from base 10 to base 2
  - g. (2 pts) Convert 1110 1010 from binary to base 10
  - h. (2 pts) Convert 197 from base 10 to binary
  - i. (2 pts) Convert 36 from base 8 to base 2
  - j. (2 pts) Convert 1010 1111 1110 1111 from base 2 to hexadecimal

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

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

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;
}
```

**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
- b. (2 pts) Convert 0011 1100 from binary to decimal
- c. (2 pts) Convert 100 001 000 from binary to octal
- d. (2 pts) Convert 100 101 010 from binary to base 8
- e. (2 pts) Convert 110 001 101 from binary to octal
- f. (2 pts) Convert 55 from octal to binary
- g. (2 pts) Convert 111 011 000 from base 2 to octal
- h. (2 pts) Convert 13 from octal to binary
- i. (2 pts) Convert 6 from base 8 to binary
- j. (2 pts) Convert 1101 1100 0011 0000 from base 2 to hexadecimal

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?
- b. (2 pts) What is the value of `argv[0][2]`?
- c. (2 pts) What is the value of `argv[1][3]`?
- d. (2 pts) What is the value of `argv[2][2]`?

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

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;
}
```

**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 1000 0001 from binary to hexadecimal
  - b. (2 pts) Convert 1111 0110 0101 1110 from binary to hexadecimal
  - c. (2 pts) Convert 0111 0001 from base 2 to decimal
  - d. (2 pts) Convert 10 from octal to binary
  - e. (2 pts) Convert 170 from decimal to base 2
  - f. (2 pts) Convert 0101 0000 1001 0011 from binary to hexadecimal
  - g. (2 pts) Convert 100 101 100 from binary to base 8
  - h. (2 pts) Convert 55 from base 8 to binary
  - i. (2 pts) Convert 1110 0101 0100 1111 from base 2 to base 16
  - j. (2 pts) Convert 0111 1000 0001 1010 from binary to hexadecimal

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?
- b. (2 pts) What is the value of `argv[1][1]`?
- c. (2 pts) What is the value of `argv[2][2]`?
- 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`
- b. (2 pts) `*h`
- c. (2 pts) `q->next->next`
- d. (2 pts) `argv[0]`
- e. (2 pts) `argv[1][2]`
- f. (2 pts) `r`
- g. (2 pts) `&q`
- h. (2 pts) `argc`
- i. (2 pts) `q->next`
- j. (2 pts) `&f`
- k. (2 pts) `f`

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;
}
```

**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 0000 1101 1101 from base 2 to base 16
- b. (2 pts) Convert 992a from base 16 to base 2
- c. (2 pts) Convert 101 110 from binary to base 8
- d. (2 pts) Convert 220 from base 10 to binary
- e. (2 pts) Convert 110 101 from binary to octal
- f. (2 pts) Convert 81 from base 10 to binary
- g. (2 pts) Convert e7d2 from base 16 to binary
- h. (2 pts) Convert 92 from base 10 to binary
- i. (2 pts) Convert 1101 0111 1100 0011 from binary to hexadecimal
- j. (2 pts) Convert 83ae from base 16 to base 2

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?
- b. (2 pts) What is the value of `argv[2][4]`?
- c. (2 pts) What is the value of `argv[1][2]`?
- d. (2 pts) What is the value of `argv[0][6]`?

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

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;
}
```

**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 100 001 011 from binary to octal
- b. (2 pts) Convert 0101 0011 0101 0011 from base 2 to hexadecimal
- c. (2 pts) Convert 001 010 from binary to octal
- d. (2 pts) Convert 68dd from base 16 to base 2
- e. (2 pts) Convert 253 from decimal to binary
- f. (2 pts) Convert 1110 1011 from base 2 to base 10
- g. (2 pts) Convert 91c2 from hexadecimal to binary
- h. (2 pts) Convert 71 from base 8 to binary
- i. (2 pts) Convert 1010 0001 1101 1010 from base 2 to base 16
- j. (2 pts) Convert 1f98 from hexadecimal to base 2

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?
- b. (2 pts) What is the value of `argv[1][1]`?
- c. (2 pts) What is the value of `argv[0][4]`?
- d. (2 pts) What is the value of `argv[2][0]`?

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`
- b. (2 pts) `x->data`
- c. (2 pts) `&z`
- d. (2 pts) `t`
- e. (2 pts) `argv[0]`
- f. (2 pts) `argc`
- g. (2 pts) `&r`
- h. (2 pts) `x->next->next`
- i. (2 pts) `x`
- j. (2 pts) `argv[1][2]`
- k. (2 pts) `x->next`

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

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

Write the full function definition for a function that would have the following prototype. The 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;
}
```

**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 55 from base 8 to base 2
- b. (2 pts) Convert 11 from octal to base 2
- c. (2 pts) Convert 3b8a from base 16 to base 2
- d. (2 pts) Convert 1100 0101 0001 1110 from binary to base 16
- e. (2 pts) Convert 0101 0011 from binary to base 10
- f. (2 pts) Convert 1e29 from hexadecimal to base 2
- g. (2 pts) Convert 93f0 from base 16 to base 2
- h. (2 pts) Convert 23 from octal to binary
- i. (2 pts) Convert 0100 0011 1001 0100 from binary to hexadecimal
- j. (2 pts) Convert 4bd9 from hexadecimal to base 2

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

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

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;
}
```

**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
- b. (2 pts) Convert 110 111 110 from binary to octal
- c. (2 pts) Convert 0010 1001 from base 2 to base 10
- d. (2 pts) Convert 0101 0001 0000 1000 from base 2 to base 16
- e. (2 pts) Convert 15 from octal to base 2
- f. (2 pts) Convert 184 from decimal to binary
- g. (2 pts) Convert 3de1 from hexadecimal to binary
- h. (2 pts) Convert 65 from octal to binary
- i. (2 pts) Convert 3 from base 8 to base 2
- j. (2 pts) Convert e7c4 from base 16 to base 2

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?
- b. (2 pts) What is the value of `argv[2][1]`?
- c. (2 pts) What is the value of `argv[0][5]`?
- d. (2 pts) What is the value of `argv[1][3]`?

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`
- b. (2 pts) `w->next`
- c. (2 pts) `x`
- d. (2 pts) `argv[1][2]`
- e. (2 pts) `argc`
- f. (2 pts) `*w`
- g. (2 pts) `&w`
- h. (2 pts) `r`
- i. (2 pts) `w->next->next`
- j. (2 pts) `argv[0]`
- k. (2 pts) `&q`

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;
}
```

**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 84 from base 10 to base 2
- b. (2 pts) Convert 1000 0001 1100 1101 from binary to base 16
- c. (2 pts) Convert 206 from decimal to binary
- d. (2 pts) Convert 011 001 from base 2 to octal
- e. (2 pts) Convert 1010 0111 from binary to decimal
- f. (2 pts) Convert 1011 1001 0101 0111 from binary to hexadecimal
- g. (2 pts) Convert 100 011 111 from base 2 to base 8
- h. (2 pts) Convert 123 from decimal to binary
- i. (2 pts) Convert 1f from hexadecimal to binary
- j. (2 pts) Convert 243 from decimal to binary

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

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

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;
}
```

**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

b. (2 pts) Convert 0011 1011 from base 2 to decimal

c. (2 pts) Convert 1011 1100 from base 2 to decimal

d. (2 pts) Convert 100 110 001 from base 2 to base 8

e. (2 pts) Convert 8b09 from hexadecimal to base 2

f. (2 pts) Convert 5334 from base 16 to binary

g. (2 pts) Convert 001 110 011 from binary to base 8

h. (2 pts) Convert 0 from base 8 to binary

i. (2 pts) Convert 1100 1010 0011 0110 from base 2 to base 16

j. (2 pts) Convert 143 from base 10 to binary

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?
- b. (2 pts) What is the value of `argv[0][5]`?
- c. (2 pts) What is the value of `argv[1][3]`?
- d. (2 pts) What is the value of `argv[2][1]`?

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

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;
}
```

**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 ca9c from hexadecimal to binary
- b. (2 pts) Convert 011 010 from base 2 to base 8
- c. (2 pts) Convert 1111 0011 0000 1100 from base 2 to hexadecimal
- d. (2 pts) Convert 1111 0100 from base 2 to decimal
- e. (2 pts) Convert 1110 0001 0010 0111 from binary to base 16
- f. (2 pts) Convert 100 001 011 from base 2 to base 8
- g. (2 pts) Convert 3bb3 from base 16 to base 2
- h. (2 pts) Convert 33 from base 8 to binary
- i. (2 pts) Convert 6bf1 from hexadecimal to binary
- j. (2 pts) Convert bb83 from base 16 to base 2

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?
- b. (2 pts) What is the value of `argv[0][5]`?
- c. (2 pts) What is the value of `argv[2][0]`?
- d. (2 pts) What is the value of `argv[1][2]`?

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

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;
}
```

**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
- b. (2 pts) Convert c7a4 from hexadecimal to binary
- c. (2 pts) Convert 1110 0000 1110 1100 from binary to hexadecimal
- d. (2 pts) Convert 1000 0000 from base 2 to base 10
- e. (2 pts) Convert 61 from octal to binary
- f. (2 pts) Convert 7 from octal to binary
- g. (2 pts) Convert e5a3 from hexadecimal to base 2
- h. (2 pts) Convert 243 from decimal to binary
- i. (2 pts) Convert 0011 0110 0000 1000 from base 2 to base 16
- j. (2 pts) Convert 87 from decimal to base 2

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

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

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;
}
```

**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: \_\_\_\_\_@ uemail.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
- b. (2 pts) Convert 011 010 100 from binary to octal
- c. (2 pts) Convert 864e from base 16 to base 2
- d. (2 pts) Convert 17 from base 8 to base 2
- e. (2 pts) Convert 52 from decimal to base 2
- f. (2 pts) Convert 20ca from hexadecimal to binary
- g. (2 pts) Convert 001 101 110 from base 2 to octal
- h. (2 pts) Convert 1001 1010 0101 1011 from binary to hexadecimal
- i. (2 pts) Convert 0010 1000 0111 1100 from binary to hexadecimal
- j. (2 pts) Convert 99 from base 10 to binary

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?
- b. (2 pts) What is the value of `argv[1][2]`?
- c. (2 pts) What is the value of `argv[2][1]`?
- d. (2 pts) What is the value of `argv[0][5]`?

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    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`
- b. (2 pts) `argv[0]`
- c. (2 pts) `b->data`
- d. (2 pts) `argv[1][2]`
- e. (2 pts) `b`
- f. (2 pts) `&c`
- g. (2 pts) `*d`
- h. (2 pts) `&y`
- i. (2 pts) `b->next->next`
- j. (2 pts) `argc`
- k. (2 pts) `z`

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;
}
```

**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 172 from decimal to base 2
- b. (2 pts) Convert 2499 from hexadecimal to base 2
- c. (2 pts) Convert 742e from hexadecimal to base 2
- d. (2 pts) Convert c82e from base 16 to binary
- e. (2 pts) Convert 0001 1000 0011 0111 from binary to base 16
- f. (2 pts) Convert 186 from decimal to binary
- g. (2 pts) Convert 111 000 010 from binary to octal
- h. (2 pts) Convert 10 from octal to binary
- i. (2 pts) Convert f293 from base 16 to base 2
- j. (2 pts) Convert 1111 1110 from binary to base 10

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?
- b. (2 pts) What is the value of `argv[0][2]`?
- c. (2 pts) What is the value of `argv[1][4]`?
- d. (2 pts) What is the value of `argv[2][0]`?

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

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;
}
```

**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: \_\_\_\_\_@ uemail.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
- b. (2 pts) Convert 246 from base 10 to base 2
- c. (2 pts) Convert 1010 1010 from binary to base 10
- d. (2 pts) Convert 001 001 000 from base 2 to base 8
- e. (2 pts) Convert 0110 1110 0101 0101 from binary to hexadecimal
- f. (2 pts) Convert 1110 1101 0001 0000 from base 2 to base 16
- g. (2 pts) Convert 111 000 110 from base 2 to octal
- h. (2 pts) Convert 42 from base 8 to binary
- i. (2 pts) Convert 45 from octal to binary
- j. (2 pts) Convert 43 from decimal to binary

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?
- b. (2 pts) What is the value of `argv[1][2]`?
- c. (2 pts) What is the value of `argv[0][4]`?
- d. (2 pts) What is the value of `argv[2][1]`?

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]`
- b. (2 pts) `r`
- c. (2 pts) `argv[0]`
- d. (2 pts) `argc`
- e. (2 pts) `p->next`
- f. (2 pts) `p->data`
- g. (2 pts) `p->next->next`
- h. (2 pts) `g`
- i. (2 pts) `&p`
- j. (2 pts) `&d`
- k. (2 pts) `*h`

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;
}
```

**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: \_\_\_\_\_@ uemail.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
- b. (2 pts) Convert 54 from octal to base 2
- c. (2 pts) Convert 46 from octal to base 2
- d. (2 pts) Convert 54 from base 8 to binary
- e. (2 pts) Convert 010 100 011 from binary to octal
- f. (2 pts) Convert 1000 0110 1110 1101 from binary to base 16
- g. (2 pts) Convert 100 011 010 from binary to octal
- h. (2 pts) Convert 18 from base 10 to binary
- i. (2 pts) Convert 5e65 from base 16 to base 2
- j. (2 pts) Convert 199 from decimal to binary

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?
- b. (2 pts) What is the value of `argv[2][0]`?
- c. (2 pts) What is the value of `argv[1][1]`?
- 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`
- b. (2 pts) `*f`
- c. (2 pts) `d->next->next`
- d. (2 pts) `argv[0]`
- e. (2 pts) `f`
- f. (2 pts) `d->data`
- g. (2 pts) `d->next`
- h. (2 pts) `argc`
- i. (2 pts) `&c`
- j. (2 pts) `&d`
- k. (2 pts) `argv[1][2]`

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;
}
```

**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 df6b from hexadecimal to binary
  - b. (2 pts) Convert 3 from octal to base 2
  - c. (2 pts) Convert 81c3 from base 16 to base 2
  - d. (2 pts) Convert 238 from base 10 to base 2
  - e. (2 pts) Convert 26 from octal to base 2
  - f. (2 pts) Convert 0100 1100 from binary to decimal
  - g. (2 pts) Convert 101 000 010 from binary to base 8
  - h. (2 pts) Convert 60 from base 8 to base 2
  - i. (2 pts) Convert 3 from base 8 to base 2
  - j. (2 pts) Convert 555e from base 16 to base 2

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?
- b. (2 pts) What is the value of `argv[2][3]`?
- c. (2 pts) What is the value of `argv[1][1]`?
- 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`
- b. (2 pts) `&s`
- c. (2 pts) `argv[0]`
- d. (2 pts) `&q`
- e. (2 pts) `s->data`
- f. (2 pts) `r`
- g. (2 pts) `argc`
- h. (2 pts) `g`
- i. (2 pts) `argv[1][2]`
- j. (2 pts) `s->next`
- k. (2 pts) `*t`

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;
}
```

**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: \_\_\_\_\_@ u mail.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
- b. (2 pts) Convert 110 010 000 from binary to octal
- c. (2 pts) Convert 111 from base 10 to binary
- d. (2 pts) Convert 011 110 101 from base 2 to octal
- e. (2 pts) Convert 0011 1011 from binary to decimal
- f. (2 pts) Convert 111 001 100 from binary to octal
- g. (2 pts) Convert 010 010 101 from binary to base 8
- h. (2 pts) Convert 22 from base 8 to base 2
- i. (2 pts) Convert 110 110 000 from binary to octal
- j. (2 pts) Convert f148 from hexadecimal to base 2

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?
- b. (2 pts) What is the value of `argv[1][0]`?
- c. (2 pts) What is the value of `argv[0][4]`?
- d. (2 pts) What is the value of `argv[2][3]`?

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

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

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

Write the full function definition for a function that would have the following prototype. The 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;
}
```

**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 0101 1010 1101 from binary to base 16

b. (2 pts) Convert 46 from base 8 to base 2

c. (2 pts) Convert 51 from base 8 to base 2

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

e. (2 pts) Convert 1001 0010 from binary to decimal

f. (2 pts) Convert 24 from base 10 to base 2

g. (2 pts) Convert 010 011 010 from binary to octal

h. (2 pts) Convert 179 from decimal to base 2

i. (2 pts) Convert 011 110 011 from binary to base 8

j. (2 pts) Convert 1d89 from base 16 to base 2

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?
- b. (2 pts) What is the value of `argv[0][3]`?
- c. (2 pts) What is the value of `argv[2][0]`?
- d. (2 pts) What is the value of `argv[1][1]`?

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

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;
}
```

**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
  - b. (2 pts) Convert 83 from base 10 to binary
  - c. (2 pts) Convert 44 from octal to base 2
  - d. (2 pts) Convert 0110 0010 from base 2 to decimal
  - e. (2 pts) Convert 7562 from hexadecimal to base 2
  - f. (2 pts) Convert 178 from base 10 to binary
  - g. (2 pts) Convert 111 101 110 from base 2 to octal
  - h. (2 pts) Convert 59 from decimal to base 2
  - i. (2 pts) Convert 0100 0011 from binary to decimal
  - j. (2 pts) Convert b973 from hexadecimal to base 2

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?
- b. (2 pts) What is the value of `argv[2][1]`?
- c. (2 pts) What is the value of `argv[0][5]`?
- d. (2 pts) What is the value of `argv[1][2]`?

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

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;
}
```

**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 0010 from binary to base 10

b. (2 pts) Convert 1111 0110 from binary to decimal

c. (2 pts) Convert 001 110 010 from base 2 to octal

d. (2 pts) Convert 7 from base 8 to base 2

e. (2 pts) Convert 229 from decimal to binary

f. (2 pts) Convert 55 from octal to binary

g. (2 pts) Convert 48c8 from base 16 to binary

h. (2 pts) Convert 70 from octal to base 2

i. (2 pts) Convert 15 from base 8 to base 2

j. (2 pts) Convert 197 from decimal to binary

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

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

c. (2 pts) `argc`

d. (2 pts) `t`

e. (2 pts) `b`

f. (2 pts) `*y`

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

h. (2 pts) `z->data`

i. (2 pts) `&a`

j. (2 pts) `&x`

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

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;
}
```

**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 37c8 from hexadecimal to binary
- b. (2 pts) Convert 176 from decimal to base 2
- c. (2 pts) Convert 0010 0111 0010 0101 from binary to hexadecimal
- d. (2 pts) Convert aa3a from base 16 to base 2
- e. (2 pts) Convert c926 from hexadecimal to base 2
- f. (2 pts) Convert 0100 1101 1111 0010 from binary to hexadecimal
- g. (2 pts) Convert f2b8 from hexadecimal to binary
- h. (2 pts) Convert 011 010 011 from base 2 to base 8
- i. (2 pts) Convert 0 from octal to binary
- j. (2 pts) Convert 96 from decimal to binary

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?
- b. (2 pts) What is the value of `argv[0][1]`?
- c. (2 pts) What is the value of `argv[1][0]`?
- d. (2 pts) What is the value of `argv[2][4]`?

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

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;
}
```

**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: \_\_\_\_\_@ uemail.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
- b. (2 pts) Convert 824c from base 16 to base 2
- c. (2 pts) Convert 93 from decimal to binary
- d. (2 pts) Convert 001 100 from base 2 to base 8
- e. (2 pts) Convert 31 from decimal to base 2
- f. (2 pts) Convert 1000 0000 from base 2 to base 10
- g. (2 pts) Convert 1111 0100 from base 2 to base 10
- h. (2 pts) Convert 210 from decimal to base 2
- i. (2 pts) Convert 50 from base 8 to base 2
- j. (2 pts) Convert 141 from decimal to base 2

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?
- b. (2 pts) What is the value of `argv[1][2]`?
- c. (2 pts) What is the value of `argv[0][6]`?
- d. (2 pts) What is the value of `argv[2][2]`?

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

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;
}
```

**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 1110 0000 1001 from base 2 to base 16

b. (2 pts) Convert 0011 1100 0111 0101 from base 2 to base 16

c. (2 pts) Convert 010 010 110 from base 2 to octal

d. (2 pts) Convert 44 from base 8 to binary

e. (2 pts) Convert 0010 1001 0000 from binary to base 16

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

g. (2 pts) Convert 1001 1110 from binary to base 10

h. (2 pts) Convert 5a22 from hexadecimal to base 2

i. (2 pts) Convert 011 011 000 from binary to octal

j. (2 pts) Convert 41 from base 10 to binary

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?
- b. (2 pts) What is the value of `argv[2][3]`?
- c. (2 pts) What is the value of `argv[1][1]`?
- d. (2 pts) What is the value of `argv[0][6]`?

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

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;
}
```

**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
  - b. (2 pts) Convert 223 from base 10 to binary
  - c. (2 pts) Convert 1111 0000 from binary to base 10
  - d. (2 pts) Convert 77 from decimal to binary
  - e. (2 pts) Convert 115 from decimal to binary
  - f. (2 pts) Convert 27 from base 10 to base 2
  - g. (2 pts) Convert f08a from base 16 to base 2
  - h. (2 pts) Convert 0 from octal to base 2
  - i. (2 pts) Convert 010 111 101 from base 2 to base 8
  - j. (2 pts) Convert 15 from base 8 to base 2

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?
- b. (2 pts) What is the value of `argv[1][3]`?
- c. (2 pts) What is the value of `argv[2][2]`?
- d. (2 pts) What is the value of `argv[0][5]`?

3. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    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`
- b. (2 pts) `w->next`
- c. (2 pts) `*w`
- d. (2 pts) `x`
- e. (2 pts) `argv[0]`
- f. (2 pts) `&w`
- g. (2 pts) `&h`
- h. (2 pts) `w->data`
- i. (2 pts) `argv[1][2]`
- j. (2 pts) `argc`
- k. (2 pts) `w->next->next`

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

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

Write the full function definition for a function that would have the following prototype. The 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;
}
```

**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
- b. (2 pts) Convert 46 from octal to base 2
- c. (2 pts) Convert 67 from base 8 to base 2
- d. (2 pts) Convert d9e2 from hexadecimal to base 2
- e. (2 pts) Convert 0101 0110 0101 0100 from binary to base 16
- f. (2 pts) Convert 1011 0101 from binary to decimal
- g. (2 pts) Convert 1001 1010 0111 1011 from binary to base 16
- h. (2 pts) Convert 136 from decimal to binary
- i. (2 pts) Convert 12 from octal to binary
- j. (2 pts) Convert 64 from base 8 to base 2

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?
- b. (2 pts) What is the value of `argv[0][0]`?
- c. (2 pts) What is the value of `argv[1][3]`?
- d. (2 pts) What is the value of `argv[2][2]`?

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

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;
}
```

**End of Exam**

total points=100