

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

# CS16—Midterm Exam

## E02, F14, Phill Conrad, UC Santa Barbara

### Wednesday, 12/03/2014

Name: \_\_\_\_\_

Umail Address: \_\_\_\_\_@ umail.ucsb.edu

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

1. Please perform the following number conversions.

- a. (2 pts) Convert 010 001 from binary to octal
- b. (2 pts) Convert 0101 0000 from binary to base 10
- c. (2 pts) Convert 1010 1100 1101 1101 from base 2 to hexadecimal
- d. (2 pts) Convert 1000 1101 from base 2 to decimal
- e. (2 pts) Convert 48 from base 10 to base 2
- f. (2 pts) Convert 0100 0001 1010 0111 from base 2 to hexadecimal
- g. (2 pts) Convert 0011 0001 from base 2 to decimal
- h. (2 pts) Convert 110 110 110 from binary to base 8
- i. (2 pts) Convert 110 101 011 from base 2 to base 8
- j. (2 pts) Convert 1000 0100 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 date banana 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[]) {
    int r;
    Node s;
    double t;
    char w;
    int *x;
    Node *y;
    double *z;
    char *a;

    return 0;
}
```

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

- a. (2 pts) `argv[1][2]`
- b. (2 pts) `&y`
- c. (2 pts) `y`
- d. (2 pts) `argv[0]`
- e. (2 pts) `y->next->next`
- f. (2 pts) `s`
- g. (2 pts) `argc`
- h. (2 pts) `y->next`
- i. (2 pts) `y->data`
- j. (2 pts) `*x`
- 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: \_\_\_\_\_@ umail.ucsb.edu

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

1. Please perform the following number conversions.

- a. (2 pts) Convert 0011 1010 0100 0010 from base 2 to base 16
- b. (2 pts) Convert 001 000 011 from base 2 to octal
- c. (2 pts) Convert 227 from base 10 to base 2
- d. (2 pts) Convert 233 from decimal to binary
- e. (2 pts) Convert 1000 0110 from base 2 to base 10
- f. (2 pts) Convert 011 101 000 from base 2 to base 8
- g. (2 pts) Convert 0011 0100 0000 0010 from binary to hexadecimal
- h. (2 pts) Convert 010 001 000 from base 2 to base 8
- i. (2 pts) Convert 011 101 111 from binary to octal
- j. (2 pts) Convert 1011 0000 from binary to decimal

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

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

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

```
./runIt guava cherry
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[2][4]`?
- c. (2 pts) What is the value of `argv[1][4]`?
- 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 s;  
    Node t;  
    int w;  
    char x;  
    double *y;  
    Node *z;  
    int *a;  
    char *b;  
  
    return 0;  
}
```

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

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

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 0111 1111 from binary to decimal
- b. (2 pts) Convert dc09 from hexadecimal to base 2
- c. (2 pts) Convert 110 100 010 from binary to octal
- d. (2 pts) Convert 117 from decimal to base 2
- e. (2 pts) Convert 0110 1010 0010 0001 from base 2 to base 16
- f. (2 pts) Convert 3 from octal to base 2
- g. (2 pts) Convert 1101 1101 1111 0011 from base 2 to hexadecimal
- h. (2 pts) Convert cc17 from base 16 to base 2
- i. (2 pts) Convert 010 000 011 from base 2 to base 8
- j. (2 pts) Convert 0100 1100 from base 2 to decimal

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

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

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

```
./runIt cherry lime apple mango
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[1][5]`?
- 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 z;
    double a;
    Node b;
    char c;
    int *d;
    double *e;
    Node *f;
    char *g;

    return 0;
}
```

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

- a. (2 pts) `f->data`
- b. (2 pts) `&c`
- c. (2 pts) `f->next->next`
- d. (2 pts) `z`
- e. (2 pts) `*e`
- f. (2 pts) `argc`
- g. (2 pts) `f->next`
- h. (2 pts) `&d`
- i. (2 pts) `argv[0]`
- j. (2 pts) `g`
- 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 1101 0111 from base 2 to decimal
  - b. (2 pts) Convert 011 111 101 from binary to octal
  - c. (2 pts) Convert 0111 0110 from base 2 to base 10
  - d. (2 pts) Convert 001 100 010 from base 2 to base 8
  - e. (2 pts) Convert 1101 1010 from base 2 to decimal
  - f. (2 pts) Convert 1111 from binary to base 10
  - g. (2 pts) Convert 0010 1111 from binary to base 10
  - h. (2 pts) Convert 7289 from base 16 to binary
  - i. (2 pts) Convert 001 101 000 from binary to octal
  - j. (2 pts) Convert 010 110 000 from binary to base 8

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

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

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

```
./runIt lemon apple kiwi
```

- a. (2 pts) What is the value of `argc` in this case?
- 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[]) {
    Node q;
    int r;
    double s;
    char t;
    Node *w;
    int *x;
    double *y;
    char *z;

    return 0;
}
```

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

- a. (2 pts) `&z`
- b. (2 pts) `argc`
- c. (2 pts) `*x`
- d. (2 pts) `w->next->next`
- e. (2 pts) `argv[0]`
- f. (2 pts) `w->data`
- g. (2 pts) `t`
- h. (2 pts) `&t`
- i. (2 pts) `w->next`
- j. (2 pts) `argv[1][2]`
- 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: \_\_\_\_\_@ umail.ucsb.edu

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

1. Please perform the following number conversions.

a. (2 pts) Convert 0001 1100 0101 1101 from binary to hexadecimal

b. (2 pts) Convert 38fd from hexadecimal to base 2

c. (2 pts) Convert 31 from base 8 to base 2

d. (2 pts) Convert 1011 1101 0100 0111 from base 2 to hexadecimal

e. (2 pts) Convert 189 from base 10 to base 2

f. (2 pts) Convert 169 from decimal to base 2

g. (2 pts) Convert 1101 1001 from base 2 to decimal

h. (2 pts) Convert 111 110 100 from binary to octal

i. (2 pts) Convert 77 from octal to base 2

j. (2 pts) Convert 111 101 000 from base 2 to octal

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

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

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

```
./runIt date lemon lime
```

- a. (2 pts) What is the value of `argc` in this case?
- 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[]) {  
    int g;  
    Node h;  
    double p;  
    char q;  
    int *r;  
    Node *s;  
    double *t;  
    char *w;  
  
    return 0;  
}
```

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

- a. (2 pts) `argv[0]`
- b. (2 pts) `p`
- c. (2 pts) `s->next->next`
- d. (2 pts) `r`
- e. (2 pts) `&s`
- f. (2 pts) `&p`
- g. (2 pts) `argc`
- h. (2 pts) `s->next`
- i. (2 pts) `argv[1][2]`
- j. (2 pts) `s->data`
- 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: \_\_\_\_\_@ umail.ucsb.edu

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

1. Please perform the following number conversions.

- a. (2 pts) Convert 77 from base 10 to base 2
- b. (2 pts) Convert 10 from base 10 to binary
- c. (2 pts) Convert 154 from decimal to binary
- d. (2 pts) Convert 0001 1001 from base 2 to base 10
- e. (2 pts) Convert 1403 from base 16 to base 2
- f. (2 pts) Convert db84 from base 16 to binary
- g. (2 pts) Convert 1101 1011 from binary to base 10
- h. (2 pts) Convert 011 000 110 from base 2 to base 8
- i. (2 pts) Convert 1001 1111 from binary to decimal
- j. (2 pts) Convert 0010 0000 from base 2 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 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][5]`?
- 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 h;
    Node p;
    int q;
    char r;
    double *s;
    Node *t;
    int *w;
    char *x;

    return 0;
}
```

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

- a. (2 pts) `t->data`
- b. (2 pts) `&x`
- c. (2 pts) `t->next->next`
- d. (2 pts) `argc`
- e. (2 pts) `t->next`
- f. (2 pts) `r`
- g. (2 pts) `argv[1][2]`
- h. (2 pts) `s`
- i. (2 pts) `&p`
- j. (2 pts) `*w`
- 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 44 from base 8 to base 2
- b. (2 pts) Convert 61 from base 8 to base 2
- c. (2 pts) Convert 1000 1000 from binary to decimal
- d. (2 pts) Convert 51 from base 8 to binary
- e. (2 pts) Convert 75 from base 8 to binary
- f. (2 pts) Convert 117 from decimal to base 2
- g. (2 pts) Convert 1000 0101 from base 2 to base 10
- h. (2 pts) Convert ead9 from base 16 to base 2
- i. (2 pts) Convert 011 010 011 from base 2 to base 8
- j. (2 pts) Convert 101 111 000 from binary to base 8

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

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

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

```
./runIt date lime kiwi cherry
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[2][1]`?
- c. (2 pts) What is the value of `argv[1][2]`?
- 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[]) {  
    double e;  
    int f;  
    Node g;  
    char h;  
    double *p;  
    int *q;  
    Node *r;  
    char *s;  
  
    return 0;  
}
```

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

- a. (2 pts) `&q`
- b. (2 pts) `&e`
- c. (2 pts) `*s`
- d. (2 pts) `r->next->next`
- e. (2 pts) `f`
- f. (2 pts) `r->next`
- g. (2 pts) `argv[1][2]`
- h. (2 pts) `argc`
- i. (2 pts) `s`
- j. (2 pts) `argv[0]`
- k. (2 pts) `r->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 0100 1111 0001 0001 from binary to hexadecimal

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

c. (2 pts) Convert 34 from octal to base 2

d. (2 pts) Convert 227 from decimal to base 2

e. (2 pts) Convert 1111 1110 0101 1011 from base 2 to base 16

f. (2 pts) Convert 0011 1011 from binary to base 10

g. (2 pts) Convert 1001 1001 from base 2 to decimal

h. (2 pts) Convert 100 110 110 from base 2 to octal

i. (2 pts) Convert 25 from base 10 to base 2

j. (2 pts) Convert 0100 1010 from binary to decimal

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

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

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

```
./runIt guava mango date
```

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

3. Given the following declarations:

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

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

- a. (2 pts) `&b`
- b. (2 pts) `argc`
- c. (2 pts) `z->next`
- d. (2 pts) `y`
- e. (2 pts) `*b`
- f. (2 pts) `argv[1][2]`
- g. (2 pts) `&x`
- h. (2 pts) `z->data`
- i. (2 pts) `z->next->next`
- j. (2 pts) `s`
- 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 1001 0011 from binary to base 10

b. (2 pts) Convert 110 111 001 from binary to base 8

c. (2 pts) Convert 27 from base 8 to base 2

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

e. (2 pts) Convert 70 from base 8 to binary

f. (2 pts) Convert 110 101 001 from binary to octal

g. (2 pts) Convert 67 from decimal to binary

h. (2 pts) Convert 001 000 101 from base 2 to octal

i. (2 pts) Convert 227 from decimal to binary

j. (2 pts) Convert 71 from base 8 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 kiwi fig
```

- 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[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[]) {  
    Node h;  
    double p;  
    int q;  
    char r;  
    Node *s;  
    double *t;  
    int *w;  
    char *x;  
  
    return 0;  
}
```

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

- a. (2 pts) `x`
- b. (2 pts) `argv[1][2]`
- c. (2 pts) `&t`
- d. (2 pts) `argv[0]`
- e. (2 pts) `s->next->next`
- f. (2 pts) `q`
- g. (2 pts) `s->next`
- h. (2 pts) `argc`
- i. (2 pts) `*t`
- j. (2 pts) `&r`
- k. (2 pts) `s->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 110 001 010 from base 2 to octal
  - b. (2 pts) Convert 53 from base 8 to binary
  - c. (2 pts) Convert 9633 from base 16 to base 2
  - d. (2 pts) Convert 203 from decimal to base 2
  - e. (2 pts) Convert 001 101 111 from base 2 to octal
  - f. (2 pts) Convert 75a from base 16 to base 2
  - g. (2 pts) Convert 0100 0101 from base 2 to base 10
  - h. (2 pts) Convert 1000 1011 from binary to decimal
  - i. (2 pts) Convert 133 from base 10 to base 2
  - j. (2 pts) Convert 18 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 banana 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[1][0]`?
- 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 a;
    Node b;
    int c;
    char d;
    double *e;
    Node *f;
    int *g;
    char *h;

    return 0;
}
```

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

- a. (2 pts) `&g`
- b. (2 pts) `f->next`
- c. (2 pts) `argv[0]`
- d. (2 pts) `argc`
- e. (2 pts) `g`
- f. (2 pts) `*e`
- g. (2 pts) `&b`
- h. (2 pts) `argv[1][2]`
- i. (2 pts) `d`
- j. (2 pts) `f->data`
- k. (2 pts) `f->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 10 from decimal to binary
  - b. (2 pts) Convert 104 from base 10 to base 2
  - c. (2 pts) Convert 1000 0100 from binary to base 10
  - d. (2 pts) Convert 87 from decimal to binary
  - e. (2 pts) Convert 27 from base 10 to binary
  - f. (2 pts) Convert 161 from base 10 to base 2
  - g. (2 pts) Convert 1110 1111 from binary to base 10
  - h. (2 pts) Convert 138f from hexadecimal to binary
  - i. (2 pts) Convert 4f6c from hexadecimal to binary
  - j. (2 pts) Convert 174 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 fig kiwi cherry
```

- 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][6]`?

3. Given the following declarations:

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

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

    return 0;
}
```

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

- a. (2 pts) `d->next->next`
- b. (2 pts) `&x`
- c. (2 pts) `y`
- d. (2 pts) `c`
- e. (2 pts) `argv[0]`
- f. (2 pts) `argc`
- g. (2 pts) `d->data`
- h. (2 pts) `*d`
- i. (2 pts) `argv[1][2]`
- j. (2 pts) `d->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 98 from base 10 to binary
- b. (2 pts) Convert 1011 from base 2 to decimal
- c. (2 pts) Convert 41 from base 10 to binary
- d. (2 pts) Convert 100 110 from base 2 to octal
- e. (2 pts) Convert 1000 1011 1000 1001 from base 2 to hexadecimal
- f. (2 pts) Convert 1010 0010 1000 1000 from binary to base 16
- g. (2 pts) Convert 65 from base 10 to base 2
- h. (2 pts) Convert 101 110 100 from binary to base 8
- i. (2 pts) Convert 0100 0001 from base 2 to base 10
- j. (2 pts) Convert 56 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 banana 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][4]`?

3. Given the following declarations:

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

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

- a. (2 pts) `s`
- b. (2 pts) `argv[1][2]`
- c. (2 pts) `s->data`
- d. (2 pts) `argc`
- e. (2 pts) `p`
- f. (2 pts) `s->next->next`
- g. (2 pts) `argv[0]`
- h. (2 pts) `&x`
- i. (2 pts) `&h`
- j. (2 pts) `*t`
- k. (2 pts) `s->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 51 from octal to binary
- b. (2 pts) Convert 197 from decimal to base 2
- c. (2 pts) Convert 101 110 from binary to octal
- d. (2 pts) Convert 1001 1111 0101 0011 from base 2 to base 16
- e. (2 pts) Convert 011 011 101 from base 2 to octal
- f. (2 pts) Convert 17 from octal to base 2
- g. (2 pts) Convert 1110 1011 0000 0111 from binary to base 16
- h. (2 pts) Convert 0100 0001 from binary to base 10
- i. (2 pts) Convert 11 from decimal to binary
- j. (2 pts) Convert 0101 0110 0010 0110 from binary to base 16

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

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

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

```
./runIt fig grape
```

- 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 e;
    double f;
    int g;
    char h;
    Node *p;
    double *q;
    int *r;
    char *s;

    return 0;
}
```

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

- a. (2 pts) `p->next`
- b. (2 pts) `&h`
- c. (2 pts) `p->data`
- d. (2 pts) `argc`
- e. (2 pts) `p->next->next`
- f. (2 pts) `argv[0]`
- g. (2 pts) `&q`
- h. (2 pts) `argv[1][2]`
- i. (2 pts) `s`
- j. (2 pts) `*s`
- 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: \_\_\_\_\_@ umail.ucsb.edu

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

1. Please perform the following number conversions.

- a. (2 pts) Convert d8d0 from base 16 to base 2
- b. (2 pts) Convert 96b0 from hexadecimal to binary
- c. (2 pts) Convert 0100 1101 1011 0101 from binary to base 16
- d. (2 pts) Convert 1111 1011 from base 2 to base 10
- e. (2 pts) Convert 1100 0100 from base 2 to decimal
- f. (2 pts) Convert 33 from base 8 to binary
- g. (2 pts) Convert 237 from base 10 to binary
- h. (2 pts) Convert aaa0 from base 16 to binary
- i. (2 pts) Convert 1010 1101 from base 2 to base 10
- j. (2 pts) Convert 40 from base 8 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 guava kiwi
```

- 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][1]`?
- 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[]) {
    double e;
    int f;
    Node g;
    char h;
    double *p;
    int *q;
    Node *r;
    char *s;

    return 0;
}
```

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

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

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 111 011 from base 2 to octal
- b. (2 pts) Convert 0101 0000 1101 1001 from base 2 to base 16
- c. (2 pts) Convert 0011 1011 1001 0101 from binary to hexadecimal
- d. (2 pts) Convert 41 from base 8 to binary
- e. (2 pts) Convert a83e from base 16 to base 2
- f. (2 pts) Convert 1000 1010 1011 from base 2 to base 16
- g. (2 pts) Convert 151 from decimal to binary
- h. (2 pts) Convert 0011 0010 0101 0000 from binary to hexadecimal
- i. (2 pts) Convert 119 from decimal to binary
- j. (2 pts) Convert 7 from octal 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 banana guava apple
```

- 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][0]`?
- 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 r;  
    int s;  
    Node t;  
    char w;  
    double *x;  
    int *y;  
    Node *z;  
    char *a;  
  
    return 0;  
}
```

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

- a. (2 pts) `y`
- b. (2 pts) `z->data`
- c. (2 pts) `argv[1][2]`
- d. (2 pts) `argc`
- e. (2 pts) `t`
- f. (2 pts) `&s`
- g. (2 pts) `z->next`
- h. (2 pts) `argv[0]`
- i. (2 pts) `z->next->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: \_\_\_\_\_@ umail.ucsb.edu

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

1. Please perform the following number conversions.
  - a. (2 pts) Convert 011 101 100 from base 2 to octal
  - b. (2 pts) Convert 243 from base 10 to base 2
  - c. (2 pts) Convert e0f7 from hexadecimal to base 2
  - d. (2 pts) Convert 4311 from base 16 to binary
  - e. (2 pts) Convert 6 from base 8 to binary
  - f. (2 pts) Convert 9 from base 10 to base 2
  - g. (2 pts) Convert 1110 1000 1101 1001 from base 2 to hexadecimal
  - h. (2 pts) Convert 110 110 001 from binary to base 8
  - i. (2 pts) Convert 106 from base 10 to base 2
  - j. (2 pts) Convert 0010 1001 1110 0101 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 banana lemon grape
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[1][5]`?
- 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[]) {  
    Node d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

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

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

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 baeb from base 16 to binary
  - b. (2 pts) Convert 53 from base 8 to binary
  - c. (2 pts) Convert ced7 from hexadecimal to base 2
  - d. (2 pts) Convert cefb from base 16 to base 2
  - e. (2 pts) Convert fc02 from base 16 to base 2
  - f. (2 pts) Convert 1010 0011 from binary to base 10
  - g. (2 pts) Convert 44 from octal to binary
  - h. (2 pts) Convert 6072 from hexadecimal to base 2
  - i. (2 pts) Convert 0011 0100 from binary to decimal
  - j. (2 pts) Convert 1100 0101 1100 1111 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 lemon
```

- 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[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[]) {  
    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`
- b. (2 pts) `argv[1][2]`
- c. (2 pts) `argv[0]`
- d. (2 pts) `argc`
- e. (2 pts) `g`
- f. (2 pts) `a`
- g. (2 pts) `&g`
- h. (2 pts) `&a`
- i. (2 pts) `f->next->next`
- j. (2 pts) `f->data`
- 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 1110 1100 from binary to base 10
- b. (2 pts) Convert 0111 1111 0101 0011 from binary to base 16
- c. (2 pts) Convert 001 010 from binary to octal
- d. (2 pts) Convert 0010 1011 0011 1101 from base 2 to hexadecimal
- e. (2 pts) Convert 82 from base 10 to base 2
- f. (2 pts) Convert 1101 0110 0100 0010 from binary to hexadecimal
- g. (2 pts) Convert 1001 0100 1111 1000 from base 2 to base 16
- h. (2 pts) Convert c961 from base 16 to base 2
- i. (2 pts) Convert d60e from base 16 to base 2
- j. (2 pts) Convert 1111 0010 0001 0000 from binary to base 16

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

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

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

```
./runIt cherry grape lemon fig
```

- 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[0][5]`?
- 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[]) {  
    int b;  
    double c;  
    Node d;  
    char e;  
    int *f;  
    double *g;  
    Node *h;  
    char *p;  
  
    return 0;  
}
```

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

- a. (2 pts) `g`
- b. (2 pts) `&d`
- c. (2 pts) `c`
- d. (2 pts) `argc`
- e. (2 pts) `h->next->next`
- f. (2 pts) `argv[1][2]`
- g. (2 pts) `h->data`
- h. (2 pts) `*f`
- i. (2 pts) `h->next`
- j. (2 pts) `argv[0]`
- 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 0011 0001 0010 1100 from binary to base 16
- b. (2 pts) Convert 0011 1001 from base 2 to decimal
- c. (2 pts) Convert 74 from base 8 to binary
- d. (2 pts) Convert 55 from base 8 to base 2
- e. (2 pts) Convert 356b from base 16 to base 2
- f. (2 pts) Convert 701e from base 16 to binary
- g. (2 pts) Convert 0011 1110 1110 1001 from binary to hexadecimal
- h. (2 pts) Convert 0101 0001 0001 0010 from base 2 to base 16
- i. (2 pts) Convert 1010 0000 from binary to decimal
- j. (2 pts) Convert 1000 1101 1111 1010 from base 2 to base 16

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

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

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

```
./runIt mango grape lemon 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][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 x;
    double y;
    Node z;
    char a;
    int *b;
    double *c;
    Node *d;
    char *e;

    return 0;
}
```

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

- a. (2 pts) `&y`
- b. (2 pts) `x`
- c. (2 pts) `d->next->next`
- d. (2 pts) `argc`
- e. (2 pts) `argv[0]`
- f. (2 pts) `*d`
- g. (2 pts) `&d`
- h. (2 pts) `b`
- i. (2 pts) `d->data`
- j. (2 pts) `argv[1][2]`
- k. (2 pts) `d->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 1000 1001 1000 1001 from binary to base 16

b. (2 pts) Convert 1101 1100 0100 1000 from base 2 to base 16

c. (2 pts) Convert 100 110 000 from binary to base 8

d. (2 pts) Convert 114 from base 10 to base 2

e. (2 pts) Convert 165 from base 10 to base 2

f. (2 pts) Convert 34 from base 8 to binary

g. (2 pts) Convert 1001 0000 from base 2 to base 10

h. (2 pts) Convert 1111 0111 from base 2 to base 10

i. (2 pts) Convert 1001 0010 from base 2 to decimal

j. (2 pts) Convert 998e 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 apple mango
```

- 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][4]`?
- 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 h;
    Node p;
    double q;
    char r;
    int *s;
    Node *t;
    double *w;
    char *x;

    return 0;
}
```

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

- a. (2 pts) `t->next->next`
- b. (2 pts) `argv[1][2]`
- c. (2 pts) `&t`
- d. (2 pts) `argv[0]`
- e. (2 pts) `*t`
- f. (2 pts) `t->data`
- g. (2 pts) `argc`
- h. (2 pts) `t->next`
- i. (2 pts) `r`
- j. (2 pts) `x`
- 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 110 011 100 from binary to octal
- b. (2 pts) Convert 1001 0110 from base 2 to decimal
- c. (2 pts) Convert 1000 0110 0101 1001 from base 2 to hexadecimal
- d. (2 pts) Convert 1111 1110 from binary to decimal
- e. (2 pts) Convert 1000 1001 0010 1111 from base 2 to base 16
- f. (2 pts) Convert 010 110 from base 2 to base 8
- g. (2 pts) Convert 0011 1010 from binary to base 10
- h. (2 pts) Convert 7f34 from hexadecimal to base 2
- i. (2 pts) Convert 010 111 001 from base 2 to base 8
- j. (2 pts) Convert 3579 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 grape banana
```

- 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][2]`?
- 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[]) {  
    Node w;  
    double x;  
    int y;  
    char z;  
    Node *a;  
    double *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

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

- a. (2 pts) `argc`
- b. (2 pts) `a->next`
- c. (2 pts) `a->data`
- d. (2 pts) `c`
- e. (2 pts) `a->next->next`
- f. (2 pts) `argv[1][2]`
- g. (2 pts) `&w`
- h. (2 pts) `*d`
- i. (2 pts) `x`
- j. (2 pts) `&d`
- 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 77 from base 8 to binary
- b. (2 pts) Convert 011 001 111 from binary to base 8
- c. (2 pts) Convert 1011 1100 from base 2 to base 10
- d. (2 pts) Convert 90 from base 10 to binary
- e. (2 pts) Convert 67 from base 8 to base 2
- f. (2 pts) Convert 001 111 011 from binary to octal
- g. (2 pts) Convert 17 from base 8 to binary
- h. (2 pts) Convert e823 from hexadecimal to base 2
- i. (2 pts) Convert 254 from base 10 to binary
- j. (2 pts) Convert 0110 0001 1011 1010 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 fig mango kiwi 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][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 x;  
    int y;  
    Node z;  
    char a;  
    double *b;  
    int *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

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

- a. (2 pts) `&a`
- b. (2 pts) `d->next`
- c. (2 pts) `c`
- d. (2 pts) `argv[0]`
- e. (2 pts) `*d`
- f. (2 pts) `d->data`
- g. (2 pts) `argc`
- h. (2 pts) `d->next->next`
- i. (2 pts) `&b`
- j. (2 pts) `z`
- 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 0100 0100 from binary to base 10
- b. (2 pts) Convert 221f from hexadecimal to binary
- c. (2 pts) Convert 1010 1010 from binary to base 10
- d. (2 pts) Convert 230 from base 10 to binary
- e. (2 pts) Convert 110 000 101 from base 2 to octal
- f. (2 pts) Convert 110 101 111 from base 2 to octal
- g. (2 pts) Convert 71 from octal to base 2
- h. (2 pts) Convert 011 011 111 from base 2 to base 8
- i. (2 pts) Convert 1100 1000 from binary to base 10
- j. (2 pts) Convert fda4 from hexadecimal 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 kiwi lime mango lemon
```

- 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][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 s;
    double t;
    Node w;
    char x;
    int *y;
    double *z;
    Node *a;
    char *b;

    return 0;
}
```

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

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

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

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

Write the full function definition for a function that would have the following prototype. The 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 56 from decimal to binary
- b. (2 pts) Convert 2 from octal to binary
- c. (2 pts) Convert c06 from hexadecimal to base 2
- d. (2 pts) Convert 111 111 from base 2 to octal
- e. (2 pts) Convert 100 111 000 from base 2 to base 8
- f. (2 pts) Convert 20 from base 10 to base 2
- g. (2 pts) Convert 0111 0110 0111 0010 from binary to base 16
- h. (2 pts) Convert c51 from base 16 to binary
- i. (2 pts) Convert 77 from octal to base 2
- j. (2 pts) Convert 1000 0110 from base 2 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 fig lemon cherry
```

- 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][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) `&g`
- b. (2 pts) `a`
- c. (2 pts) `e->next->next`
- d. (2 pts) `*g`
- e. (2 pts) `argv[1][2]`
- f. (2 pts) `e->data`
- g. (2 pts) `e->next`
- h. (2 pts) `argv[0]`
- i. (2 pts) `&b`
- j. (2 pts) `argc`
- 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 37 from octal to binary
- b. (2 pts) Convert 110 001 000 from base 2 to base 8
- c. (2 pts) Convert f9e6 from base 16 to base 2
- d. (2 pts) Convert 1010 1011 from base 2 to base 10
- e. (2 pts) Convert 127 from base 10 to binary
- f. (2 pts) Convert 1010 1110 from base 2 to decimal
- g. (2 pts) Convert 0010 0000 0110 0011 from base 2 to base 16
- h. (2 pts) Convert 9401 from hexadecimal to binary
- i. (2 pts) Convert 1100 0111 1111 0000 from base 2 to hexadecimal
- j. (2 pts) Convert 001 000 100 from binary to octal

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

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

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

```
./runIt cherry kiwi
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[0][3]`?
- c. (2 pts) What is the value of `argv[2][1]`?
- 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 w;
    double x;
    int y;
    char z;
    Node *a;
    double *b;
    int *c;
    char *d;

    return 0;
}
```

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

- a. (2 pts) `*a`
- b. (2 pts) `argv[0]`
- c. (2 pts) `a->next`
- d. (2 pts) `a->data`
- e. (2 pts) `&x`
- f. (2 pts) `argc`
- g. (2 pts) `z`
- h. (2 pts) `&b`
- i. (2 pts) `d`
- j. (2 pts) `a->next->next`
- 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: \_\_\_\_\_@ 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 af31 from base 16 to base 2
- b. (2 pts) Convert 1001 0101 from binary to decimal
- c. (2 pts) Convert 0011 0000 from base 2 to decimal
- d. (2 pts) Convert 7 from decimal to binary
- e. (2 pts) Convert d5e6 from base 16 to binary
- f. (2 pts) Convert 1110 0000 1100 0110 from binary to base 16
- g. (2 pts) Convert 0010 0010 1001 0001 from binary to hexadecimal
- h. (2 pts) Convert 77 from octal to binary
- i. (2 pts) Convert 32 from octal to base 2
- j. (2 pts) Convert 0100 1110 from base 2 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 cherry guava
```

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

3. Given the following declarations:

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

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

- a. (2 pts) `c`
- b. (2 pts) `argv[1][2]`
- c. (2 pts) `w`
- d. (2 pts) `&z`
- e. (2 pts) `b->data`
- f. (2 pts) `b->next->next`
- g. (2 pts) `&c`
- h. (2 pts) `b->next`
- i. (2 pts) `argc`
- j. (2 pts) `*a`
- 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 111 101 000 from binary to base 8
  - b. (2 pts) Convert 79 from decimal to base 2
  - c. (2 pts) Convert 7 from base 8 to binary
  - d. (2 pts) Convert 147 from base 10 to base 2
  - e. (2 pts) Convert 56 from base 8 to binary
  - f. (2 pts) Convert 7aa2 from hexadecimal to base 2
  - g. (2 pts) Convert 1100 1100 1000 0001 from base 2 to hexadecimal
  - h. (2 pts) Convert 100 001 001 from base 2 to octal
  - i. (2 pts) Convert 0011 0011 1100 0010 from base 2 to hexadecimal
  - j. (2 pts) Convert 1110 1010 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 lemon fig guava 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][4]`?
- 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 r;
    double s;
    Node t;
    char w;
    int *x;
    double *y;
    Node *z;
    char *a;

    return 0;
}
```

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

- a. (2 pts) `y`
- b. (2 pts) `&x`
- c. (2 pts) `z->next->next`
- d. (2 pts) `argv[1][2]`
- e. (2 pts) `argc`
- f. (2 pts) `z->next`
- g. (2 pts) `w`
- h. (2 pts) `&t`
- i. (2 pts) `*y`
- j. (2 pts) `argv[0]`
- k. (2 pts) `z->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 010 011 000 from base 2 to base 8
- b. (2 pts) Convert 1111 0010 from binary to decimal
- c. (2 pts) Convert 110 000 111 from base 2 to base 8
- d. (2 pts) Convert 010 011 110 from base 2 to octal
- e. (2 pts) Convert 0010 1001 from base 2 to base 10
- f. (2 pts) Convert 36 from base 8 to base 2
- g. (2 pts) Convert 111 100 from base 2 to octal
- h. (2 pts) Convert 2b12 from base 16 to binary
- i. (2 pts) Convert 0010 0110 0011 0101 from binary to hexadecimal
- j. (2 pts) Convert 111 101 100 from binary to octal

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

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

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

```
./runIt lime apple grape
```

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

3. Given the following declarations:

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

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

- a. (2 pts) `h->next->next`
- b. (2 pts) `h->data`
- c. (2 pts) `h->next`
- d. (2 pts) `*h`
- e. (2 pts) `&r`
- f. (2 pts) `argv[1][2]`
- g. (2 pts) `&f`
- h. (2 pts) `r`
- i. (2 pts) `argv[0]`
- j. (2 pts) `f`
- 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 145 from decimal to binary
- b. (2 pts) Convert 1010 1100 1101 0000 from base 2 to base 16
- c. (2 pts) Convert 1011 0001 0110 1000 from base 2 to hexadecimal
- d. (2 pts) Convert 1101 1011 0011 1100 from base 2 to hexadecimal
- e. (2 pts) Convert 3 from base 8 to binary
- f. (2 pts) Convert 0001 0101 1101 0000 from base 2 to base 16
- g. (2 pts) Convert 110 010 000 from binary to octal
- h. (2 pts) Convert 54 from base 8 to base 2
- i. (2 pts) Convert 74 from base 8 to base 2
- j. (2 pts) Convert 100 100 100 from binary to octal

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

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

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

```
./runIt lime date
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[0][1]`?
- c. (2 pts) What is the value of `argv[2][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[]) {
    Node q;
    double r;
    int s;
    char t;
    Node *w;
    double *x;
    int *y;
    char *z;

    return 0;
}
```

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

- a. (2 pts) `y`
- b. (2 pts) `*y`
- c. (2 pts) `argc`
- d. (2 pts) `q`
- e. (2 pts) `&x`
- f. (2 pts) `w->data`
- g. (2 pts) `argv[1][2]`
- h. (2 pts) `w->next->next`
- i. (2 pts) `argv[0]`
- j. (2 pts) `&r`
- k. (2 pts) `w->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 1100 0010 from base 2 to decimal
  
- b. (2 pts) Convert 7e56 from base 16 to binary
  
- c. (2 pts) Convert 1110 0111 from base 2 to base 10
  
- d. (2 pts) Convert 0011 0111 from base 2 to decimal
  
- e. (2 pts) Convert 0110 0011 0001 0011 from base 2 to hexadecimal
  
- f. (2 pts) Convert 010 010 000 from binary to octal
  
- g. (2 pts) Convert 1100 1010 0101 0011 from binary to base 16
  
- h. (2 pts) Convert 110 111 from binary to octal
  
- i. (2 pts) Convert 1001 0010 0000 0111 from binary to base 16
  
- j. (2 pts) Convert 101 111 100 from binary to octal

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

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

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

```
./runIt lemon cherry apple lime
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[2][0]`?
- c. (2 pts) What is the value of `argv[0][5]`?
- 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 q;
    int r;
    Node s;
    char t;
    double *w;
    int *x;
    Node *y;
    char *z;

    return 0;
}
```

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

- a. (2 pts) `argv[0]`
- b. (2 pts) `*x`
- c. (2 pts) `x`
- d. (2 pts) `&z`
- e. (2 pts) `argc`
- f. (2 pts) `argv[1][2]`
- g. (2 pts) `&q`
- h. (2 pts) `s`
- i. (2 pts) `y->next`
- j. (2 pts) `y->data`
- k. (2 pts) `y->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 78d from base 16 to base 2
- b. (2 pts) Convert 001 110 000 from binary to base 8
- c. (2 pts) Convert 110 101 011 from binary to base 8
- d. (2 pts) Convert 60 from base 8 to base 2
- e. (2 pts) Convert 21 from base 8 to base 2
- f. (2 pts) Convert 70 from octal to base 2
- g. (2 pts) Convert 0111 0100 0100 0100 from base 2 to hexadecimal
- h. (2 pts) Convert 163 from decimal to binary
- i. (2 pts) Convert 5c1e from base 16 to base 2
- j. (2 pts) Convert 010 110 100 from base 2 to octal

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

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

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

```
./runIt lemon fig lime guava
```

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

3. Given the following declarations:

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

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

- a. (2 pts) `argc`
- b. (2 pts) `t->data`
- c. (2 pts) `t->next->next`
- d. (2 pts) `&s`
- e. (2 pts) `g`
- f. (2 pts) `r`
- g. (2 pts) `argv[0]`
- h. (2 pts) `*w`
- i. (2 pts) `argv[1][2]`
- j. (2 pts) `t->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: \_\_\_\_\_@ 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 0101 1111 1110 1010 from base 2 to base 16
- b. (2 pts) Convert 219 from decimal to base 2
- c. (2 pts) Convert 0111 1011 from base 2 to decimal
- d. (2 pts) Convert 7efb from base 16 to base 2
- e. (2 pts) Convert 1011 0110 1101 1000 from base 2 to hexadecimal
- f. (2 pts) Convert e366 from hexadecimal to base 2
- g. (2 pts) Convert 1100 0101 from base 2 to base 10
- h. (2 pts) Convert 49d3 from hexadecimal to base 2
- i. (2 pts) Convert 4e92 from hexadecimal to binary
- j. (2 pts) Convert 31 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 fig cherry lemon
```

- 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][4]`?

3. Given the following declarations:

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

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

- a. (2 pts) `argc`
- b. (2 pts) `&d`
- c. (2 pts) `d->next`
- d. (2 pts) `c`
- e. (2 pts) `argv[0]`
- f. (2 pts) `d->next->next`
- g. (2 pts) `g`
- h. (2 pts) `d->data`
- i. (2 pts) `&c`
- j. (2 pts) `*f`
- 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 101 001 001 from base 2 to base 8
- b. (2 pts) Convert 45 from octal to binary
- c. (2 pts) Convert 0110 1000 from binary to decimal
- d. (2 pts) Convert ae5 from base 16 to binary
- e. (2 pts) Convert 46 from base 8 to base 2
- f. (2 pts) Convert 125 from base 10 to base 2
- g. (2 pts) Convert 0110 1111 from binary to base 10
- h. (2 pts) Convert 110 100 011 from base 2 to base 8
- i. (2 pts) Convert 0001 1000 1010 1001 from base 2 to base 16
- j. (2 pts) Convert 0 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 apple lemon
```

- 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][1]`?

3. Given the following declarations:

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

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

    return 0;
}
```

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

- a. (2 pts) `argv[0]`
- b. (2 pts) `argv[1][2]`
- c. (2 pts) `&y`
- d. (2 pts) `b->next`
- e. (2 pts) `argc`
- f. (2 pts) `b->data`
- g. (2 pts) `&c`
- h. (2 pts) `y`
- i. (2 pts) `c`
- j. (2 pts) `*a`
- k. (2 pts) `b->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 65 from octal to binary
- b. (2 pts) Convert 0110 0110 1111 1001 from base 2 to base 16
- c. (2 pts) Convert 159 from decimal to binary
- d. (2 pts) Convert 0110 0111 0010 0110 from base 2 to base 16
- e. (2 pts) Convert 111 100 000 from base 2 to base 8
- f. (2 pts) Convert 53 from base 8 to binary
- g. (2 pts) Convert 0111 0010 from base 2 to decimal
- h. (2 pts) Convert 001 110 100 from base 2 to base 8
- i. (2 pts) Convert 1011 1010 0110 0011 from binary to hexadecimal
- j. (2 pts) Convert 001 011 100 from base 2 to octal

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

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

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

```
./runIt grape lime lemon mango
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[2][0]`?
- c. (2 pts) What is the value of `argv[0][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[]) {
    int g;
    double h;
    Node p;
    char q;
    int *r;
    double *s;
    Node *t;
    char *w;

    return 0;
}
```

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

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

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 1011 from base 2 to decimal
  - b. (2 pts) Convert 0010 0001 from binary to decimal
  - c. (2 pts) Convert 141 from decimal to binary
  - d. (2 pts) Convert f311 from base 16 to binary
  - e. (2 pts) Convert 211 from base 10 to base 2
  - f. (2 pts) Convert 0100 1001 1000 1010 from binary to hexadecimal
  - g. (2 pts) Convert 0001 1100 from binary to base 10
  - h. (2 pts) Convert 194 from decimal to base 2
  - i. (2 pts) Convert 1000 0100 0111 1011 from base 2 to base 16
  - j. (2 pts) Convert 62 from octal 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 guava fig mango cherry
```

- 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[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[]) {
    double c;
    int d;
    Node e;
    char f;
    double *g;
    int *h;
    Node *p;
    char *q;

    return 0;
}
```

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

- a. (2 pts) `p->next->next`
- b. (2 pts) `argc`
- c. (2 pts) `&f`
- d. (2 pts) `d`
- e. (2 pts) `g`
- f. (2 pts) `p->next`
- g. (2 pts) `argv[0]`
- h. (2 pts) `argv[1][2]`
- i. (2 pts) `p->data`
- j. (2 pts) `&p`
- 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: \_\_\_\_\_@ umail.ucsb.edu

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

1. Please perform the following number conversions.
  - a. (2 pts) Convert 0111 0011 from binary to decimal
  - b. (2 pts) Convert 110 000 111 from binary to base 8
  - c. (2 pts) Convert 328b from hexadecimal to base 2
  - d. (2 pts) Convert 101 011 101 from binary to octal
  - e. (2 pts) Convert 0100 0100 0000 0101 from base 2 to hexadecimal
  - f. (2 pts) Convert 010 010 101 from binary to octal
  - g. (2 pts) Convert 6db9 from base 16 to binary
  - h. (2 pts) Convert 32 from octal to base 2
  - i. (2 pts) Convert 118 from decimal to binary
  - j. (2 pts) Convert 1101 0101 1000 0010 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 lime kiwi 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[1][2]`?
- 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 t;
    Node w;
    double x;
    char y;
    int *z;
    Node *a;
    double *b;
    char *c;

    return 0;
}
```

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

- a. (2 pts) `c`
- b. (2 pts) `&a`
- c. (2 pts) `a->next->next`
- d. (2 pts) `*z`
- e. (2 pts) `a->next`
- f. (2 pts) `&y`
- g. (2 pts) `t`
- h. (2 pts) `argv[0]`
- i. (2 pts) `argc`
- j. (2 pts) `argv[1][2]`
- k. (2 pts) `a->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 1011 1000 0100 0110 from base 2 to base 16
- b. (2 pts) Convert 0111 1110 from binary to decimal
- c. (2 pts) Convert 32 from decimal to base 2
- d. (2 pts) Convert 0011 1010 from binary to base 10
- e. (2 pts) Convert 001 001 110 from base 2 to base 8
- f. (2 pts) Convert 71 from octal to base 2
- g. (2 pts) Convert 0001 0111 1010 1010 from binary to hexadecimal
- h. (2 pts) Convert 111 100 000 from binary to base 8
- i. (2 pts) Convert 4105 from hexadecimal to base 2
- j. (2 pts) Convert 0111 0001 0110 1100 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 mango 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[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 q;  
    Node r;  
    int s;  
    char t;  
    double *w;  
    Node *x;  
    int *y;  
    char *z;  
  
    return 0;  
}
```

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

- a. (2 pts) `x->next->next`
- b. (2 pts) `argc`
- c. (2 pts) `t`
- d. (2 pts) `x->data`
- e. (2 pts) `&s`
- f. (2 pts) `argv[0]`
- g. (2 pts) `argv[1][2]`
- h. (2 pts) `&z`
- i. (2 pts) `*y`
- j. (2 pts) `x->next`
- 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: \_\_\_\_\_@ umail.ucsb.edu

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

1. Please perform the following number conversions.

- a. (2 pts) Convert e9a8 from hexadecimal to binary
- b. (2 pts) Convert 79 from base 10 to base 2
- c. (2 pts) Convert 25 from octal to base 2
- d. (2 pts) Convert 150 from decimal to base 2
- e. (2 pts) Convert 0111 1101 from base 2 to base 10
- f. (2 pts) Convert 0001 0111 from base 2 to decimal
- g. (2 pts) Convert 0001 1001 from base 2 to base 10
- h. (2 pts) Convert 89 from base 10 to base 2
- i. (2 pts) Convert e2c0 from hexadecimal to binary
- j. (2 pts) Convert 47 from octal 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 guava apple lemon 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][4]`?
- d. (2 pts) What is the value of `argv[0][1]`?

3. Given the following declarations:

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

int main(int argc, char *argv[]) {
    int c;
    double d;
    Node e;
    char f;
    int *g;
    double *h;
    Node *p;
    char *q;

    return 0;
}
```

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

- a. (2 pts) `p->next->next`
- b. (2 pts) `c`
- c. (2 pts) `&d`
- d. (2 pts) `argv[1][2]`
- e. (2 pts) `*h`
- f. (2 pts) `p->data`
- g. (2 pts) `h`
- h. (2 pts) `argc`
- i. (2 pts) `p->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 13 from octal to binary
- b. (2 pts) Convert 9c5 from base 16 to binary
- c. (2 pts) Convert 44aa from base 16 to base 2
- d. (2 pts) Convert 34 from decimal to binary
- e. (2 pts) Convert 60ba from base 16 to binary
- f. (2 pts) Convert 1011 0000 from binary to decimal
- g. (2 pts) Convert c3c9 from hexadecimal to binary
- h. (2 pts) Convert e0e4 from base 16 to base 2
- i. (2 pts) Convert 1010 1100 1101 0111 from base 2 to base 16
- j. (2 pts) Convert 16 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 fig banana
```

- 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][5]`?

3. Given the following declarations:

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

int main(int argc, char *argv[]) {
    int z;
    double a;
    Node b;
    char c;
    int *d;
    double *e;
    Node *f;
    char *g;

    return 0;
}
```

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

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

b. (2 pts) `*g`

c. (2 pts) `b`

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

e. (2 pts) `argc`

f. (2 pts) `&z`

g. (2 pts) `&f`

h. (2 pts) `d`

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

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

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 72 from base 8 to base 2
- b. (2 pts) Convert 6778 from base 16 to base 2
- c. (2 pts) Convert 46 from base 10 to base 2
- d. (2 pts) Convert 6106 from base 16 to binary
- e. (2 pts) Convert 0110 0111 from base 2 to base 10
- f. (2 pts) Convert 011 101 101 from binary to base 8
- g. (2 pts) Convert 1101 0111 0110 1000 from binary to hexadecimal
- h. (2 pts) Convert 100 100 010 from base 2 to octal
- i. (2 pts) Convert 010 111 000 from base 2 to base 8
- j. (2 pts) Convert 1100 0111 1101 1110 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 kiwi banana
```

- 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][2]`?

3. Given the following declarations:

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

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

- a. (2 pts) `b->next`
- b. (2 pts) `argv[0]`
- c. (2 pts) `argc`
- d. (2 pts) `b->data`
- e. (2 pts) `c`
- f. (2 pts) `argv[1][2]`
- g. (2 pts) `y`
- h. (2 pts) `*e`
- i. (2 pts) `b->next->next`
- j. (2 pts) `&x`
- 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 0010 1111 from base 2 to decimal
- b. (2 pts) Convert 0010 0001 1010 0001 from binary to hexadecimal
- c. (2 pts) Convert 27 from decimal to binary
- d. (2 pts) Convert ecf0 from base 16 to base 2
- e. (2 pts) Convert 4b12 from base 16 to binary
- f. (2 pts) Convert 4 from octal to base 2
- g. (2 pts) Convert 1000 0001 0101 1001 from base 2 to base 16
- h. (2 pts) Convert 110 001 from base 2 to octal
- i. (2 pts) Convert 0010 0110 from binary to decimal
- j. (2 pts) Convert 63c9 from hexadecimal 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 lemon
```

- 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[1][0]`?
- 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 d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

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

- a. (2 pts) `h->next->next`
- b. (2 pts) `g`
- c. (2 pts) `h->next`
- d. (2 pts) `&g`
- e. (2 pts) `argv[1][2]`
- f. (2 pts) `argc`
- g. (2 pts) `*h`
- h. (2 pts) `h->data`
- i. (2 pts) `&q`
- j. (2 pts) `h`
- 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: \_\_\_\_\_@ 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 011 000 010 from binary to octal
- b. (2 pts) Convert 74 from base 8 to base 2
- c. (2 pts) Convert 24 from octal to base 2
- d. (2 pts) Convert 0100 1001 0011 0010 from base 2 to base 16
- e. (2 pts) Convert 1010 0001 0011 0000 from base 2 to hexadecimal
- f. (2 pts) Convert 0100 0010 from base 2 to decimal
- g. (2 pts) Convert 1000 0011 from binary to decimal
- h. (2 pts) Convert 81ea from hexadecimal to binary
- i. (2 pts) Convert 110 010 000 from base 2 to base 8
- j. (2 pts) Convert 100 100 000 from binary to octal

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

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

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

```
./runIt grape lime
```

- a. (2 pts) What is the value of `argc` in this case?
- b. (2 pts) What is the value of `argv[2][0]`?
- c. (2 pts) What is the value of `argv[0][3]`?
- 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 d;  
    double e;  
    int f;  
    char g;  
    Node *h;  
    double *p;  
    int *q;  
    char *r;  
  
    return 0;  
}
```

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

- a. (2 pts) `&e`
- b. (2 pts) `h->data`
- c. (2 pts) `*h`
- d. (2 pts) `r`
- e. (2 pts) `argc`
- f. (2 pts) `h->next->next`
- g. (2 pts) `d`
- h. (2 pts) `&h`
- i. (2 pts) `argv[0]`
- j. (2 pts) `h->next`
- 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 166 from decimal to base 2
- b. (2 pts) Convert 101 011 010 from base 2 to octal
- c. (2 pts) Convert 4020 from base 16 to base 2
- d. (2 pts) Convert 65 from base 8 to binary
- e. (2 pts) Convert 100 001 000 from base 2 to octal
- f. (2 pts) Convert 110 111 001 from base 2 to octal
- g. (2 pts) Convert 45 from base 10 to binary
- h. (2 pts) Convert 99a from base 16 to binary
- i. (2 pts) Convert 44 from octal to binary
- j. (2 pts) Convert 001 010 111 from binary to octal

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

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

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

```
./runIt fig guava lime banana
```

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

3. Given the following declarations:

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

int main(int argc, char *argv[]) {
    int a;
    double b;
    Node c;
    char d;
    int *e;
    double *f;
    Node *g;
    char *h;

    return 0;
}
```

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

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

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 fe78 from hexadecimal to base 2
- b. (2 pts) Convert 24 from base 8 to base 2
- c. (2 pts) Convert 1110 0101 1000 0001 from binary to hexadecimal
- d. (2 pts) Convert 100 100 001 from binary to octal
- e. (2 pts) Convert 244 from base 10 to base 2
- f. (2 pts) Convert 222 from decimal to binary
- g. (2 pts) Convert 0111 1111 0010 1011 from binary to base 16
- h. (2 pts) Convert 101 100 000 from base 2 to octal
- i. (2 pts) Convert 41 from base 8 to base 2
- j. (2 pts) Convert 3788 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 mango fig kiwi
```

- 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[]) {
    int r;
    Node s;
    double t;
    char w;
    int *x;
    Node *y;
    double *z;
    char *a;

    return 0;
}
```

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

- a. (2 pts) `y`
- b. (2 pts) `t`
- c. (2 pts) `y->data`
- d. (2 pts) `argc`
- e. (2 pts) `&r`
- f. (2 pts) `&y`
- g. (2 pts) `argv[0]`
- h. (2 pts) `y->next->next`
- i. (2 pts) `y->next`
- j. (2 pts) `*y`
- 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: \_\_\_\_\_@ 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 20 from octal to base 2
- b. (2 pts) Convert 10 from decimal to binary
- c. (2 pts) Convert d362 from base 16 to binary
- d. (2 pts) Convert 0001 1100 from binary to base 10
- e. (2 pts) Convert 1101 1000 0100 0000 from base 2 to hexadecimal
- f. (2 pts) Convert 7801 from hexadecimal to base 2
- g. (2 pts) Convert 0010 1001 0001 1011 from base 2 to hexadecimal
- h. (2 pts) Convert 55 from base 10 to binary
- i. (2 pts) Convert 010 011 101 from binary to octal
- j. (2 pts) Convert d372 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 fig cherry banana
```

- 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][2]`?
- 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[]) {
    int h;
    Node p;
    double q;
    char r;
    int *s;
    Node *t;
    double *w;
    char *x;

    return 0;
}
```

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

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