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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: __________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 7cd2 from hexadecimal to binary

   b. (2 pts) Convert 0010 1001 1000 1101 from base 2 to hexadecimal

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

   d. (2 pts) Convert 138 from decimal to base 2

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

   f. (2 pts) Convert 1000 0000 1101 0001 from binary to hexadecimal

   g. (2 pts) Convert d67f from base 16 to binary

   h. (2 pts) Convert 41 from base 8 to base 2

   i. (2 pts) Convert 70 from decimal to base 2

   j. (2 pts) Convert 0001 from binary to decimal

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

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

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

```
./thing kiwi fig 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][0]?

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 b;
  char c;
  double d;
  Node e;
  int *f;
  char *g;
  double *h;
  Node *p;


  return 0;
}
```

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

a. (2 pts) p->next

b. (2 pts) argc

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

d. (2 pts) &c

e. (2 pts) b

f. (2 pts) f

g. (2 pts) argv[0]

h. (2 pts) p->data

i. (2 pts) &p

j. (2 pts) *p

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

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

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

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

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 174 from base 10 to binary

   b. (2 pts) Convert fb12 from base 16 to base 2

   c. (2 pts) Convert 0011 1101 1011 1010 from binary to base 16

   d. (2 pts) Convert e72d from base 16 to base 2

   e. (2 pts) Convert 0001 1001 from binary to base 10

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

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

   h. (2 pts) Convert 73 from octal to base 2

   i. (2 pts) Convert 1110 1000 from binary to base 10

   j. (2 pts) Convert 45 from decimal to base 2

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

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

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

   ```
   ./thing fig mango
   ```

   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][0]?

3. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) argc

c. (2 pts) p->data

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

e. (2 pts) &b

f. (2 pts) &g

g. (2 pts) *p

h. (2 pts) argv[0]

i. (2 pts) p->next

j. (2 pts) p

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 1111 0011 0001 0011 from base 2 to base 16

    b. (2 pts) Convert 1011 0101 0011 1011 from binary to hexadecimal

    c. (2 pts) Convert 12 from octal to base 2

    d. (2 pts) Convert 7317 from base 16 to binary

    e. (2 pts) Convert fcdd from hexadecimal to base 2

    f. (2 pts) Convert 010 011 010 from base 2 to base 8

    g. (2 pts) Convert 829e from base 16 to binary

    h. (2 pts) Convert 119 from decimal to base 2

    i. (2 pts) Convert 178 from base 10 to binary

    j. (2 pts) Convert 201 from base 10 to base 2

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

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

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

```
./thing lemon guava cherry apple
```

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

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

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

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) e

c. (2 pts) f->next->next

d. (2 pts) f->data

e. (2 pts) b

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

g. (2 pts) f->next

h. (2 pts) &f

i. (2 pts) argv[0]

j. (2 pts) &b

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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







  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 0100 1011 0111 0000 from base 2 to base 16

   b. (2 pts) Convert 5807 from hexadecimal to binary

   c. (2 pts) Convert 1101 0000 1111 1100 from binary to hexadecimal

   d. (2 pts) Convert 0010 1110 from binary to decimal

   e. (2 pts) Convert 33 from octal to binary

   f. (2 pts) Convert 78 from decimal to binary

   g. (2 pts) Convert d451 from hexadecimal to base 2

   h. (2 pts) Convert 7 from octal to base 2

   i. (2 pts) Convert 164 from base 10 to binary

   j. (2 pts) Convert 1101 0101 from binary to decimal

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

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

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

   ```
   ./thing apple banana kiwi guava
   ```

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

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

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

   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 p;
  int q;
  char r;
  Node s;
  double *t;
  int *w;
  char *x;
  Node *y;


  return 0;
}
```

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

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

b. (2 pts) argc

c. (2 pts) *y

d. (2 pts) &x

e. (2 pts) y->data

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

g. (2 pts) y->next

h. (2 pts) &s

i. (2 pts) w

j. (2 pts) argv[0]

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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







  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

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

   b. (2 pts) Convert 4 from octal to base 2

   c. (2 pts) Convert 1011 1110 1101 1101 from binary to hexadecimal

   d. (2 pts) Convert 56 from octal to base 2

   e. (2 pts) Convert 50a1 from hexadecimal to base 2

   f. (2 pts) Convert 1110 1000 from binary to decimal

   g. (2 pts) Convert 7e41 from base 16 to base 2

   h. (2 pts) Convert 51 from octal to binary

   i. (2 pts) Convert 0110 1111 from binary to base 10

   j. (2 pts) Convert 0111 0001 from base 2 to base 10

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

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

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

```
./thing kiwi grape 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[1][1]?

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

3. Given the following declarations:

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


int main(int argc, char *argv[]) {
  char f;
  int g;
  double h;
  Node p;
  char *q;
  int *r;
  double *s;
  Node *t;


  return 0;
}
```

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

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

b. (2 pts) &h

c. (2 pts) *r

d. (2 pts) t->data

e. (2 pts) q

f. (2 pts) t->next->next

g. (2 pts) argc

h. (2 pts) argv[0]

i. (2 pts) g

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 60 from octal to binary

   b. (2 pts) Convert 70 from base 8 to base 2

   c. (2 pts) Convert 1111 0101 from binary to base 10

   d. (2 pts) Convert 22 from decimal to binary

   e. (2 pts) Convert 166 from decimal to base 2

   f. (2 pts) Convert 1aae from base 16 to base 2

   g. (2 pts) Convert 8070 from hexadecimal to base 2

   h. (2 pts) Convert 3 from base 8 to base 2

   i. (2 pts) Convert 10cc from base 16 to binary

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

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

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

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

```
./thing mango 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[2][0]?

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[]) {
  char y;
  double z;
  int a;
  Node b;
  char *c;
  double *d;
  int *e;
  Node *f;


  return 0;
}
```

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

a. (2 pts) a

b. (2 pts) f->next

c. (2 pts) f->data

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

e. (2 pts) &d

f. (2 pts) argv[0]

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

h. (2 pts) f

i. (2 pts) &y

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 6 from decimal to binary

    b. (2 pts) Convert 157 from decimal to binary

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

    d. (2 pts) Convert 101 000 101 from binary to base 8

    e. (2 pts) Convert 8a0b from hexadecimal to binary

    f. (2 pts) Convert b48b from base 16 to base 2

    g. (2 pts) Convert 2a60 from base 16 to base 2

    h. (2 pts) Convert 149 from decimal to base 2

    i. (2 pts) Convert 1101 1010 from binary to base 10

    j. (2 pts) Convert 0011 1001 from base 2 to decimal

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

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

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

```
./thing kiwi banana
```

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

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

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

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) s

b. (2 pts) argv[0]

c. (2 pts) a

d. (2 pts) *a

e. (2 pts) b->next

f. (2 pts) &x

g. (2 pts) b->data

h. (2 pts) argc

i. (2 pts) &b

j. (2 pts) b->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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 94 from decimal to binary

   b. (2 pts) Convert 010 000 001 from binary to octal

   c. (2 pts) Convert 100 010 000 from binary to base 8

   d. (2 pts) Convert 0101 1110 0101 0011 from binary to base 16

   e. (2 pts) Convert 250 from decimal to base 2

   f. (2 pts) Convert 101 101 011 from base 2 to base 8

   g. (2 pts) Convert 011 111 000 from binary to octal

   h. (2 pts) Convert 17 from octal to binary

   i. (2 pts) Convert 205 from base 10 to binary

   j. (2 pts) Convert 010 001 001 from base 2 to base 8

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

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

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

```
./thing banana apple guava kiwi
```

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][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 f;
  double g;
  char h;
  Node p;
  int *q;
  double *r;
  char *s;
  Node *t;


  return 0;
}
```

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

a. (2 pts) &s

b. (2 pts) t->next->next

c. (2 pts) t->data

d. (2 pts) argv[0]

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

f. (2 pts) t->next

g. (2 pts) f

h. (2 pts) *r

i. (2 pts) argc

j. (2 pts) r

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 50 from octal to base 2

   b. (2 pts) Convert 1111 1010 from binary to base 10

   c. (2 pts) Convert 35 from base 8 to base 2

   d. (2 pts) Convert 1110 1010 0011 1101 from binary to hexadecimal

   e. (2 pts) Convert 1101 1101 1100 1111 from binary to hexadecimal

   f. (2 pts) Convert 23 from octal to binary

   g. (2 pts) Convert 001 001 100 from base 2 to octal

   h. (2 pts) Convert 60 from octal to binary

   i. (2 pts) Convert 100 101 110 from binary to base 8

   j. (2 pts) Convert 111 000 001 from binary to base 8

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

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

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

```
./thing grape mango 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][3]?

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

b. (2 pts) *p

c. (2 pts) p->next

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

e. (2 pts) &d

f. (2 pts) d

g. (2 pts) p->next->next

h. (2 pts) p

i. (2 pts) p->data

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert d52e from base 16 to binary

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

    c. (2 pts) Convert 172 from base 10 to binary

    d. (2 pts) Convert 21 from octal to binary

    e. (2 pts) Convert 14 from octal to binary

    f. (2 pts) Convert 130 from decimal to binary

    g. (2 pts) Convert 001 010 000 from binary to octal

    h. (2 pts) Convert 44 from base 10 to binary

    i. (2 pts) Convert 57 from base 10 to binary

    j. (2 pts) Convert 1101 from binary to base 10

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

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

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

```
./thing date fig
```

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

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

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

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


  return 0;
}
```

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

a. (2 pts) b->next

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

c. (2 pts) &t

d. (2 pts) b->data

e. (2 pts) argc

f. (2 pts) *a

g. (2 pts) b

h. (2 pts) &a

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

j. (2 pts) x

k. (2 pts) argv[0]

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

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

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

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

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

    b. (2 pts) Convert 8681 from base 16 to base 2

    c. (2 pts) Convert 154 from decimal to binary

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

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

    f. (2 pts) Convert 0001 1011 from base 2 to base 10

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

    h. (2 pts) Convert 180 from base 10 to binary

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

    j. (2 pts) Convert 101 010 001 from binary to base 8

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

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

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

```
./thing grape banana
```

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

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

c. (2 pts) What is the value of argv[1][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[]) {
  char p;
  double q;
  int r;
  Node s;
  char *t;
  double *w;
  int *x;
  Node *y;


  return 0;
}
```

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

a. (2 pts) &t

b. (2 pts) x

c. (2 pts) &p

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

e. (2 pts) y->next

f. (2 pts) *t

g. (2 pts) argc

h. (2 pts) r

i. (2 pts) argv[0]

j. (2 pts) y->next->next

k. (2 pts) y->data

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

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

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

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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





  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 011 100 100 from base 2 to octal

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

    c. (2 pts) Convert 0011 1111 from base 2 to base 10

    d. (2 pts) Convert 1000 1101 from base 2 to base 10

    e. (2 pts) Convert 87b1 from hexadecimal to base 2

    f. (2 pts) Convert 1d4f from base 16 to base 2

    g. (2 pts) Convert 001 000 111 from binary to base 8

    h. (2 pts) Convert 26 from base 8 to binary

    i. (2 pts) Convert 1111 0101 from base 2 to decimal

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

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

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

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

```
./thing guava lemon kiwi lime
```

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][0]?

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) *p

b. (2 pts) p->next->next

c. (2 pts) p->data

d. (2 pts) f

e. (2 pts) argv[0]

f. (2 pts) &b

g. (2 pts) c

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

i. (2 pts) p->next

j. (2 pts) argc

k. (2 pts) &p

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

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

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

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 183 from decimal to binary

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

   c. (2 pts) Convert 0010 1101 from binary to decimal

   d. (2 pts) Convert 25 from base 10 to binary

   e. (2 pts) Convert 0110 1010 1111 1100 from binary to hexadecimal

   f. (2 pts) Convert 183 from decimal to base 2

   g. (2 pts) Convert 205 from base 10 to binary

   h. (2 pts) Convert 226 from base 10 to binary

   i. (2 pts) Convert 191 from decimal to base 2

   j. (2 pts) Convert 24 from base 8 to binary

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

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

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

```
./thing date fig 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[2][2]?

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) *d

b. (2 pts) f

c. (2 pts) f->next

d. (2 pts) f->data

e. (2 pts) f->next->next

f. (2 pts) argv[0]

g. (2 pts) &b

h. (2 pts) &d

i. (2 pts) b

j. (2 pts) argc

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 1110 1000 from base 2 to base 10

   b. (2 pts) Convert 1011 0100 1111 1100 from base 2 to hexadecimal

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

   d. (2 pts) Convert 7627 from hexadecimal to base 2

   e. (2 pts) Convert 60 from octal to binary

   f. (2 pts) Convert 72 from octal to binary

   g. (2 pts) Convert 110 011 111 from binary to base 8

   h. (2 pts) Convert 75 from decimal to binary

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

   j. (2 pts) Convert 011 111 001 from base 2 to octal

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

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

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

```
./thing cherry apple
```

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

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

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

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) &f

c. (2 pts) *c

d. (2 pts) e

e. (2 pts) f->next->next

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

g. (2 pts) y

h. (2 pts) argv[0]

i. (2 pts) f->data

j. (2 pts) &z

k. (2 pts) f->next

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

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

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

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ___________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 2d8b from base 16 to binary

    b. (2 pts) Convert 33 from base 8 to base 2

    c. (2 pts) Convert 521f from hexadecimal to base 2

    d. (2 pts) Convert 211 from base 16 to binary

    e. (2 pts) Convert 1010 0100 from binary to base 10

    f. (2 pts) Convert 1000 0011 0111 0010 from base 2 to hexadecimal

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

    h. (2 pts) Convert 1101 0011 0110 0000 from base 2 to hexadecimal

    i. (2 pts) Convert 43 from decimal to base 2

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

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

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

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

```
./thing kiwi cherry
```

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

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

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

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) s->data

b. (2 pts) &q

c. (2 pts) argv[0]

d. (2 pts) argc

e. (2 pts) s->next

f. (2 pts) q

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

h. (2 pts) &e

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

j. (2 pts) *r

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 0010 0001 1101 0001 from binary to hexadecimal

    b. (2 pts) Convert 0101 0111 from base 2 to base 10

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

    d. (2 pts) Convert 58 from decimal to binary

    e. (2 pts) Convert 0111 1110 from base 2 to decimal

    f. (2 pts) Convert 110 000 000 from base 2 to octal

    g. (2 pts) Convert 2 from octal to base 2

    h. (2 pts) Convert 0110 1111 from base 2 to base 10

    i. (2 pts) Convert 0110 0000 1111 0100 from binary to hexadecimal

    j. (2 pts) Convert 1010 0001 from binary to base 10

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

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

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

```
./thing banana lime guava kiwi
```

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][3]?

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) h->next

b. (2 pts) d

c. (2 pts) argv[0]

d. (2 pts) &f

e. (2 pts) &a

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

g. (2 pts) h->next->next

h. (2 pts) argc

i. (2 pts) f

j. (2 pts) h->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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ___________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 0110 from binary to base 10

   b. (2 pts) Convert 0001 0001 0011 0010 from binary to hexadecimal

   c. (2 pts) Convert 50 from base 8 to base 2

   d. (2 pts) Convert 198 from decimal to base 2

   e. (2 pts) Convert 30 from base 8 to binary

   f. (2 pts) Convert 26 from base 8 to binary

   g. (2 pts) Convert 54 from base 8 to base 2

   h. (2 pts) Convert 247 from base 10 to base 2

   i. (2 pts) Convert 12 from base 8 to base 2

   j. (2 pts) Convert 0011 1101 from base 2 to base 10

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

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

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

```
./thing date kiwi banana
```

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

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

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

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

b. (2 pts) &e

c. (2 pts) *d

d. (2 pts) e->next->next

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

f. (2 pts) argc

g. (2 pts) e->data

h. (2 pts) &a

i. (2 pts) y

j. (2 pts) b

k. (2 pts) e->next

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

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

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

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

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

   b. (2 pts) Convert e2b8 from hexadecimal to base 2

   c. (2 pts) Convert 215 from base 10 to binary

   d. (2 pts) Convert 2316 from hexadecimal to binary

   e. (2 pts) Convert 1011 0111 1011 0011 from base 2 to base 16

   f. (2 pts) Convert 140 from decimal to base 2

   g. (2 pts) Convert b5cb from hexadecimal to base 2

   h. (2 pts) Convert 0110 0000 0111 1100 from binary to base 16

   i. (2 pts) Convert 63 from base 8 to binary

   j. (2 pts) Convert 0110 1001 from base 2 to base 10

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

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

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

```
./thing grape mango
```

   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[]) {
  double y;
  char z;
  int a;
  Node b;
  double *c;
  char *d;
  int *e;
  Node *f;


  return 0;
}
```

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

a. (2 pts) &a

b. (2 pts) b

c. (2 pts) f

d. (2 pts) &d

e. (2 pts) argc

f. (2 pts) argv[0]

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

h. (2 pts) f->next->next

i. (2 pts) *e

j. (2 pts) f->next

k. (2 pts) f->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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: _______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 220 from base 10 to binary

   b. (2 pts) Convert 100 111 001 from base 2 to octal

   c. (2 pts) Convert 110 001 011 from binary to octal

   d. (2 pts) Convert af00 from hexadecimal to base 2

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

   f. (2 pts) Convert 0010 0110 from binary to base 10

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

   h. (2 pts) Convert 1110 1000 0010 1101 from base 2 to base 16

   i. (2 pts) Convert 45 from octal to base 2

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

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

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

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

```
./thing guava lemon banana 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][4]?

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


  return 0;
}
```

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

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

b. (2 pts) e

c. (2 pts) argv[0]

d. (2 pts) r->data

e. (2 pts) argc

f. (2 pts) *r

g. (2 pts) &e

h. (2 pts) r->next->next

i. (2 pts) &r

j. (2 pts) r->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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ___________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 53 from base 10 to binary

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

    c. (2 pts) Convert 0110 1011 from base 2 to base 10

    d. (2 pts) Convert 0110 1010 from binary to base 10

    e. (2 pts) Convert 1011 0111 0111 from base 2 to hexadecimal

    f. (2 pts) Convert 27d3 from base 16 to binary

    g. (2 pts) Convert 101 100 010 from binary to octal

    h. (2 pts) Convert 1000 1110 from base 2 to base 10

    i. (2 pts) Convert 8950 from hexadecimal to binary

    j. (2 pts) Convert 100 010 from base 2 to octal

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

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

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

```
./thing grape mango cherry fig
```

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

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

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

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 w;
  int x;
  char y;
  Node z;
  double *a;
  int *b;
  char *c;
  Node *d;


  return 0;
}
```

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

a. (2 pts) d->next

b. (2 pts) &x

c. (2 pts) d->next->next

d. (2 pts) d->data

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

f. (2 pts) a

g. (2 pts) &c

h. (2 pts) w

i. (2 pts) argc

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: _____________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 0111 1010 0010 1101 from base 2 to hexadecimal

    b. (2 pts) Convert 76 from base 8 to base 2

    c. (2 pts) Convert 88 from decimal to binary

    d. (2 pts) Convert 75 from octal to binary

    e. (2 pts) Convert 73 from base 8 to base 2

    f. (2 pts) Convert 193 from base 10 to binary

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

    h. (2 pts) Convert 0001 0110 0100 1111 from base 2 to base 16

    i. (2 pts) Convert 0101 0011 0110 0111 from base 2 to base 16

    j. (2 pts) Convert 101 011 010 from binary to octal

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

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

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

```
./thing lemon guava fig
```

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

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

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


  return 0;
}
```

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

a. (2 pts) *x

b. (2 pts) t

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

d. (2 pts) argv[0]

e. (2 pts) a->data

f. (2 pts) &r

g. (2 pts) a

h. (2 pts) a->next

i. (2 pts) argc

j. (2 pts) &x

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

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

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

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

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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







  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert ab8f from hexadecimal to base 2

   b. (2 pts) Convert 1100 1011 from binary to base 10

   c. (2 pts) Convert 43 from octal to base 2

   d. (2 pts) Convert 010 100 101 from binary to base 8

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

   f. (2 pts) Convert 1111 0100 0001 1001 from binary to base 16

   g. (2 pts) Convert 27 from octal to binary

   h. (2 pts) Convert 0111 1111 0011 1110 from binary to base 16

   i. (2 pts) Convert 1111 0101 0010 0010 from binary to hexadecimal

   j. (2 pts) Convert 1101 1001 from binary to base 10

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

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

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

```
./thing lemon mango
```

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][0]?

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) b->data

b. (2 pts) &z

c. (2 pts) argv[0]

d. (2 pts) argc

e. (2 pts) &x

f. (2 pts) b->next->next

g. (2 pts) *y

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

i. (2 pts) b->next

j. (2 pts) b

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ________________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 74 from base 8 to base 2

    b. (2 pts) Convert 133 from decimal to base 2

    c. (2 pts) Convert 7d2e from base 16 to base 2

    d. (2 pts) Convert 1101 1110 from binary to decimal

    e. (2 pts) Convert 40 from base 10 to binary

    f. (2 pts) Convert 8df6 from hexadecimal to base 2

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

    h. (2 pts) Convert 1 from octal to base 2

    i. (2 pts) Convert 57 from base 8 to base 2

    j. (2 pts) Convert 011 101 010 from binary to octal

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

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

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

```
./thing date grape apple kiwi
```

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

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

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[]) {
  double p;
  int q;
  char r;
  Node s;
  double *t;
  int *w;
  char *x;
  Node *y;


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) y->data

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

d. (2 pts) *w

e. (2 pts) r

f. (2 pts) y->next

g. (2 pts) x

h. (2 pts) y->next->next

i. (2 pts) argv[0]

j. (2 pts) &r

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ___________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 010 010 001 from binary to base 8

    b. (2 pts) Convert 001 010 000 from base 2 to octal

    c. (2 pts) Convert 0010 0010 1001 0000 from binary to hexadecimal

    d. (2 pts) Convert 1001 1010 0011 1100 from base 2 to base 16

    e. (2 pts) Convert 1001 1000 1010 0101 from base 2 to base 16

    f. (2 pts) Convert 1000 1111 from binary to base 10

    g. (2 pts) Convert 89 from decimal to base 2

    h. (2 pts) Convert 1010 1101 from binary to decimal

    i. (2 pts) Convert 54 from octal to binary

    j. (2 pts) Convert 40 from octal to base 2

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

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

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

```
./thing lemon kiwi date apple
```

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][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 b;
  double c;
  char d;
  Node e;
  int *f;
  double *g;
  char *h;
  Node *p;


  return 0;
}
```

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

a. (2 pts) p->data

b. (2 pts) &c

c. (2 pts) c

d. (2 pts) argc

e. (2 pts) p->next->next

f. (2 pts) *f

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

h. (2 pts) &h

i. (2 pts) f

j. (2 pts) argv[0]

k. (2 pts) p->next

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

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

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

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 141 from base 10 to binary

    b. (2 pts) Convert e279 from base 16 to base 2

    c. (2 pts) Convert 16 from decimal to binary

    d. (2 pts) Convert 2626 from base 16 to base 2

    e. (2 pts) Convert 011 110 111 from base 2 to base 8

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

    g. (2 pts) Convert 3 from base 10 to binary

    h. (2 pts) Convert 0011 0101 0001 0000 from base 2 to hexadecimal

    i. (2 pts) Convert 7bc4 from base 16 to base 2

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

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

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

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

```
./thing grape mango lime
```

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

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

c. (2 pts) What is the value of argv[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[]) {
  char g;
  int h;
  double p;
  Node q;
  char *r;
  int *s;
  double *t;
  Node *w;


  return 0;
}
```

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

a. (2 pts) *t

b. (2 pts) &s

c. (2 pts) w->data

d. (2 pts) &g

e. (2 pts) w

f. (2 pts) argv[0]

g. (2 pts) argc

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

i. (2 pts) w->next->next

j. (2 pts) w->next

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 1011 1111 from binary to base 10

   b. (2 pts) Convert 179 from decimal to binary

   c. (2 pts) Convert 010 001 101 from binary to octal

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

   e. (2 pts) Convert 1101 0010 from base 2 to decimal

   f. (2 pts) Convert 0101 1011 from base 2 to decimal

   g. (2 pts) Convert 5 from decimal to base 2

   h. (2 pts) Convert 1001 1101 1111 1111 from base 2 to hexadecimal

   i. (2 pts) Convert 1d7e from hexadecimal to binary

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

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

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

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

```
./thing lemon kiwi
```

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

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

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

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

3. Given the following declarations:

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


int main(int argc, char *argv[]) {
  char h;
  double p;
  int q;
  Node r;
  char *s;
  double *t;
  int *w;
  Node *x;


  return 0;
}
```

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

a. (2 pts) x->next

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

c. (2 pts) p

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

e. (2 pts) x->data

f. (2 pts) argv[0]

g. (2 pts) &w

h. (2 pts) w

i. (2 pts) *w

j. (2 pts) argc

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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






  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 3ec from hexadecimal to base 2

   b. (2 pts) Convert 6e27 from hexadecimal to base 2

   c. (2 pts) Convert 0011 0100 1010 1111 from base 2 to hexadecimal

   d. (2 pts) Convert 1110 0101 0010 from binary to hexadecimal

   e. (2 pts) Convert b559 from base 16 to binary

   f. (2 pts) Convert 245 from decimal to base 2

   g. (2 pts) Convert 175 from base 10 to base 2

   h. (2 pts) Convert 0010 0101 from binary to decimal

   i. (2 pts) Convert 1110 0111 1001 0110 from base 2 to base 16

   j. (2 pts) Convert 111 001 001 from base 2 to octal

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

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

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

```
./thing fig guava
```

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

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

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

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


  return 0;
}
```

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

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

b. (2 pts) &g

c. (2 pts) h

d. (2 pts) *s

e. (2 pts) q

f. (2 pts) s->next

g. (2 pts) s->data

h. (2 pts) argv[0]

i. (2 pts) &p

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

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 prameters to the function and the return value should be as described in the comment.

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

Answer in the space below

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

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

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

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

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

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

Answer in the space below.

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

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

And given the same function prototype:

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

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

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

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

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

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

#include "complex.h"

int main() {
  Complex c;

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







  // Show result

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

End of Exam

total points=100

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

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

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

1. Please perform the following number conversions.

    a. (2 pts) Convert 5c48 from hexadecimal to base 2

    b. (2 pts) Convert 16 from decimal to binary

    c. (2 pts) Convert 66 from octal to base 2

    d. (2 pts) Convert 201 from base 10 to binary

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

    f. (2 pts) Convert 1111 0110 1011 1010 from binary to base 16

    g. (2 pts) Convert 0 from octal to binary

    h. (2 pts) Convert 1100 1100 0010 0001 from binary to hexadecimal

    i. (2 pts) Convert 1101 1010 0000 1001 from binary to hexadecimal

    j. (2 pts) Convert 74 from base 8 to base 2

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

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

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

```
./thing banana mango fig date
```

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][3]?

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

3. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) e->next

b. (2 pts) z

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

d. (2 pts) &e

e. (2 pts) argv[0]

f. (2 pts) e->data

g. (2 pts) &z

h. (2 pts) argc

i. (2 pts) *d

j. (2 pts) e->next->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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ___________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
    - Each exam is numbered (e.g. Exam #137).
    - Each pages is numbered (e.g. Page 1, Page 2, etc.)
    - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 101 000 010 from binary to base 8

    b. (2 pts) Convert 1100 1011 0001 1100 from base 2 to base 16

    c. (2 pts) Convert 61 from base 8 to base 2

    d. (2 pts) Convert 85 from base 10 to base 2

    e. (2 pts) Convert 1001 from base 2 to decimal

    f. (2 pts) Convert 9097 from base 16 to base 2

    g. (2 pts) Convert 52 from base 8 to base 2

    h. (2 pts) Convert 0101 0011 1101 0010 from binary to hexadecimal

    i. (2 pts) Convert a420 from base 16 to binary

    j. (2 pts) Convert 43 from octal to base 2

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing lemon apple banana
```

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][2]?

d. (2 pts) What is the value of argv[0][5]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  char t;
  double w;
  Node x;
  int *y;
  char *z;
  double *a;
  Node *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) b->next->next

b. (2 pts) t

c. (2 pts) argv[0]

d. (2 pts) b->next

e. (2 pts) argv[1][2]

f. (2 pts) b

g. (2 pts) argc

h. (2 pts) b->data

i. (2 pts) &z

j. (2 pts) &t

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
    - Each exam is numbered (e.g. Exam #137).
    - Each pages is numbered (e.g. Page 1, Page 2, etc.)
    - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 64 from base 8 to binary

    b. (2 pts) Convert 1001 1100 1010 0001 from base 2 to hexadecimal

    c. (2 pts) Convert 1111 1110 from base 2 to base 10

    d. (2 pts) Convert b210 from base 16 to binary

    e. (2 pts) Convert 95 from base 10 to binary

    f. (2 pts) Convert 60 from base 8 to binary

    g. (2 pts) Convert 173 from decimal to binary

    h. (2 pts) Convert 57 from base 8 to binary

    i. (2 pts) Convert 0100 0101 1101 1011 from binary to hexadecimal

    j. (2 pts) Convert 56 from base 8 to base 2

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing guava apple
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[0][1]?

c. (2 pts) What is the value of argv[1][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[]) {
  double d;
  char e;
  int f;
  Node g;
  double *h;
  char *p;
  int *q;
  Node *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r->next->next

b. (2 pts) r->data

c. (2 pts) q

d. (2 pts) &d

e. (2 pts) r->next

f. (2 pts) argc

g. (2 pts) &r

h. (2 pts) *r

i. (2 pts) argv[1][2]

j. (2 pts) f

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
    - Each exam is numbered (e.g. Exam #137).
    - Each pages is numbered (e.g. Page 1, Page 2, etc.)
    - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0001 0111 from base 2 to base 10

   b. (2 pts) Convert 0101 0110 from binary to base 10

   c. (2 pts) Convert 1110 1100 from binary to base 10

   d. (2 pts) Convert 3dfb from base 16 to base 2

   e. (2 pts) Convert 0100 0010 1000 0111 from base 2 to base 16

   f. (2 pts) Convert 27 from octal to binary

   g. (2 pts) Convert 5703 from base 16 to binary

   h. (2 pts) Convert 0100 0100 from base 2 to decimal

   i. (2 pts) Convert ff2 from base 16 to base 2

   j. (2 pts) Convert 25 from octal to base 2

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing guava grape
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[1][2]?

c. (2 pts) What is the value of argv[0][5]?

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;
  char b;
  int c;
  Node d;
  double *e;
  char *f;
  int *g;
  Node *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &f

b. (2 pts) h->next

c. (2 pts) f

d. (2 pts) &d

e. (2 pts) *f

f. (2 pts) argv[1][2]

g. (2 pts) argc

h. (2 pts) a

i. (2 pts) argv[0]

j. (2 pts) h->next->next

k. (2 pts) h->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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 1101 0011 from binary to decimal

    b. (2 pts) Convert 55 from octal to binary

    c. (2 pts) Convert 65 from octal to base 2

    d. (2 pts) Convert 0111 1100 0100 0111 from binary to hexadecimal

    e. (2 pts) Convert 73 from base 10 to binary

    f. (2 pts) Convert 0010 0010 1001 0001 from binary to base 16

    g. (2 pts) Convert 6aa2 from base 16 to base 2

    h. (2 pts) Convert 1111 0100 1101 0111 from base 2 to base 16

    i. (2 pts) Convert 57 from octal to base 2

    j. (2 pts) Convert 111 000 101 from base 2 to octal

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing fig lemon apple
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[2][0]?

c. (2 pts) What is the value of argv[0][6]?

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 h;
  char p;
  double q;
  Node r;
  int *s;
  char *t;
  double *w;
  Node *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) x->data

b. (2 pts) x->next->next

c. (2 pts) r

d. (2 pts) argv[1][2]

e. (2 pts) w

f. (2 pts) argc

g. (2 pts) x->next

h. (2 pts) &w

i. (2 pts) &q

j. (2 pts) *t

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5







  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 18bb from base 16 to binary

   b. (2 pts) Convert 110 from base 10 to base 2

   c. (2 pts) Convert c367 from base 16 to binary

   d. (2 pts) Convert 1000 0011 0001 from base 2 to base 16

   e. (2 pts) Convert 0010 1100 1101 1111 from base 2 to hexadecimal

   f. (2 pts) Convert bc6e from hexadecimal to binary

   g. (2 pts) Convert 5 from octal to base 2

   h. (2 pts) Convert 37 from base 8 to base 2

   i. (2 pts) Convert 100 010 011 from binary to octal

   j. (2 pts) Convert 011 111 101 from base 2 to octal

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing apple cherry fig
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[2][5]?

c. (2 pts) What is the value of argv[0][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[]) {
  char e;
  int f;
  double g;
  Node h;
  char *p;
  int *q;
  double *r;
  Node *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0]

b. (2 pts) s->next

c. (2 pts) argv[1][2]

d. (2 pts) &q

e. (2 pts) q

f. (2 pts) s->next->next

g. (2 pts) *r

h. (2 pts) f

i. (2 pts) &f

j. (2 pts) argc

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 74 from decimal to base 2

   b. (2 pts) Convert 0100 0000 from base 2 to decimal

   c. (2 pts) Convert 1111 1001 from base 2 to base 10

   d. (2 pts) Convert 31 from octal to binary

   e. (2 pts) Convert 40 from base 8 to base 2

   f. (2 pts) Convert 73 from base 8 to binary

   g. (2 pts) Convert 16c1 from base 16 to binary

   h. (2 pts) Convert 71 from octal to base 2

   i. (2 pts) Convert 12 from octal to base 2

   j. (2 pts) Convert 101 010 110 from base 2 to base 8

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing mango 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[2][2]?

d. (2 pts) What is the value of argv[0][1]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  char e;
  double f;
  int g;
  Node h;
  char *p;
  double *q;
  int *r;
  Node *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) q

b. (2 pts) g

c. (2 pts) *r

d. (2 pts) argv[0]

e. (2 pts) s->next

f. (2 pts) &r

g. (2 pts) argv[1][2]

h. (2 pts) s->data

i. (2 pts) argc

j. (2 pts) &e

k. (2 pts) s->next->next

4. (20 pts) Given the following struct definition:

```
struct Precip {
  int day;
  double inches;
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 1000 1110 1111 1101 from binary to base 16

    b. (2 pts) Convert 1111 1010 0101 0100 from binary to base 16

    c. (2 pts) Convert 71 from base 8 to base 2

    d. (2 pts) Convert 1111 0000 0101 1101 from binary to base 16

    e. (2 pts) Convert 011 001 100 from base 2 to base 8

    f. (2 pts) Convert 42 from base 8 to base 2

    g. (2 pts) Convert c0b2 from hexadecimal to binary

    h. (2 pts) Convert 109 from decimal to base 2

    i. (2 pts) Convert 111 101 010 from base 2 to octal

    j. (2 pts) Convert 010 001 101 from binary to octal

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing banana grape cherry date
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[0][0]?

c. (2 pts) What is the value of argv[2][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[]) {
  int b;
  double c;
  char d;
  Node e;
  int *f;
  double *g;
  char *h;
  Node *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) p

b. (2 pts) b

c. (2 pts) &e

d. (2 pts) argc

e. (2 pts) p->next

f. (2 pts) p->data

g. (2 pts) argv[0]

h. (2 pts) argv[1][2]

i. (2 pts) p->next->next

j. (2 pts) *p

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5







  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 71 from octal to base 2

   b. (2 pts) Convert 9d20 from hexadecimal to binary

   c. (2 pts) Convert 100 011 010 from base 2 to octal

   d. (2 pts) Convert 1010 1011 from base 2 to base 10

   e. (2 pts) Convert 65 from base 8 to binary

   f. (2 pts) Convert 100 010 100 from binary to base 8

   g. (2 pts) Convert 100 100 from base 2 to octal

   h. (2 pts) Convert 0001 0011 1001 1001 from binary to base 16

   i. (2 pts) Convert 1110 0111 1101 1110 from base 2 to base 16

   j. (2 pts) Convert 24 from base 8 to base 2

2. Assume the `main` function in the program `thing.cpp` starts with:

   ```
   int main(int argc, char *argv[]) {
   ...
   ```

   Further, suppose this program is invoked with the following command line:

   ```
   ./thing apple banana date
   ```

   a. (2 pts) What is the value of `argc` in this case?

   b. (2 pts) What is the value of argv[0][2]?

   c. (2 pts) What is the value of argv[2][5]?

   d. (2 pts) What is the value of argv[1][2]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  char t;
  double w;
  Node x;
  int *y;
  char *z;
  double *a;
  Node *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc

b. (2 pts) &x

c. (2 pts) &b

d. (2 pts) s

e. (2 pts) *b

f. (2 pts) b->data

g. (2 pts) a

h. (2 pts) argv[0]

i. (2 pts) argv[1][2]

j. (2 pts) b->next->next

k. (2 pts) b->next

4. (20 pts) Given the following struct definition:

```
struct Precip {
  int day;
  double inches;
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 0010 1100 from binary to decimal

    b. (2 pts) Convert 5749 from hexadecimal to base 2

    c. (2 pts) Convert 011 110 101 from base 2 to octal

    d. (2 pts) Convert 55 from base 10 to binary

    e. (2 pts) Convert 1011 1010 from base 2 to base 10

    f. (2 pts) Convert 10 from octal to base 2

    g. (2 pts) Convert 101 111 000 from binary to octal

    h. (2 pts) Convert 46 from base 8 to base 2

    i. (2 pts) Convert 54 from octal to binary

    j. (2 pts) Convert 73 from octal to base 2

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing lime mango guava
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[1][1]?

c. (2 pts) What is the value of argv[2][2]?

d. (2 pts) What is the value of argv[0][1]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int a;
  char b;
  double c;
  Node d;
  int *e;
  char *f;
  double *g;
  Node *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) f

b. (2 pts) *e

c. (2 pts) h->next

d. (2 pts) h->data

e. (2 pts) argc

f. (2 pts) &c

g. (2 pts) argv[1][2]

h. (2 pts) &f

i. (2 pts) argv[0]

j. (2 pts) h->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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ____________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
    - Each exam is numbered (e.g. Exam #137).
    - Each pages is numbered (e.g. Page 1, Page 2, etc.)
    - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 111 011 from base 2 to base 8

   b. (2 pts) Convert 28ce from base 16 to binary

   c. (2 pts) Convert 1011 0001 0100 1000 from base 2 to hexadecimal

   d. (2 pts) Convert 941c from base 16 to binary

   e. (2 pts) Convert 100 000 from base 2 to base 8

   f. (2 pts) Convert 86 from base 10 to binary

   g. (2 pts) Convert 57 from base 8 to base 2

   h. (2 pts) Convert 0100 from binary to base 10

   i. (2 pts) Convert 24 from base 8 to base 2

   j. (2 pts) Convert 6 from octal to binary

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing date guava
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[0][0]?

c. (2 pts) What is the value of argv[1][3]?

d. (2 pts) What is the value of argv[2][3]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double b;
  char c;
  int d;
  Node e;
  double *f;
  char *g;
  int *h;
  Node *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0]

b. (2 pts) &b

c. (2 pts) p->data

d. (2 pts) &p

e. (2 pts) *f

f. (2 pts) e

g. (2 pts) p->next

h. (2 pts) argc

i. (2 pts) p->next->next

j. (2 pts) argv[1][2]

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ___________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 162 from base 10 to base 2

    b. (2 pts) Convert 111 000 101 from base 2 to base 8

    c. (2 pts) Convert 159 from decimal to binary

    d. (2 pts) Convert 2006 from base 16 to base 2

    e. (2 pts) Convert f376 from base 16 to binary

    f. (2 pts) Convert 1111 0000 from base 2 to decimal

    g. (2 pts) Convert 32 from octal to binary

    h. (2 pts) Convert 139 from decimal to base 2

    i. (2 pts) Convert 111 011 from binary to octal

    j. (2 pts) Convert 55 from base 8 to binary

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing date grape apple fig
```

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[]) {
  double y;
  int z;
  char a;
  Node b;
  double *c;
  int *d;
  char *e;
  Node *f;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2]

b. (2 pts) &d

c. (2 pts) &y

d. (2 pts) f->next->next

e. (2 pts) f->data

f. (2 pts) f->next

g. (2 pts) *d

h. (2 pts) f

i. (2 pts) z

j. (2 pts) argv[0]

k. (2 pts) argc

4. (20 pts) Given the following struct definition:

```
struct Precip {
  int day;
  double inches;
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 250 from base 10 to base 2

    b. (2 pts) Convert 1000 0101 1100 0011 from binary to base 16

    c. (2 pts) Convert 448a from base 16 to base 2

    d. (2 pts) Convert 1101 1011 from base 2 to base 10

    e. (2 pts) Convert 011 000 111 from base 2 to base 8

    f. (2 pts) Convert 74 from base 8 to base 2

    g. (2 pts) Convert 186 from base 10 to base 2

    h. (2 pts) Convert 14 from octal to base 2

    i. (2 pts) Convert 100 000 from base 2 to base 8

    j. (2 pts) Convert 1100 0010 0011 0001 from binary to base 16

2. Assume the `main` function in the program `thing.cpp` starts with:

   ```
   int main(int argc, char *argv[]) {
   ...
   ```

   Further, suppose this program is invoked with the following command line:

   ```
   ./thing date mango cherry
   ```

   a. (2 pts) What is the value of `argc` in this case?

   b. (2 pts) What is the value of argv[0][0]?

   c. (2 pts) What is the value of argv[2][0]?

   d. (2 pts) What is the value of argv[1][1]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  char p;
  int q;
  double r;
  Node s;
  char *t;
  int *w;
  double *x;
  Node *y;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y->next

b. (2 pts) argc

c. (2 pts) y->data

d. (2 pts) x

e. (2 pts) argv[0]

f. (2 pts) *w

g. (2 pts) argv[1][2]

h. (2 pts) q

i. (2 pts) &s

j. (2 pts) &t

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
    - Each exam is numbered (e.g. Exam #137).
    - Each pages is numbered (e.g. Page 1, Page 2, etc.)
    - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 17 from base 8 to base 2

   b. (2 pts) Convert 17 from octal to binary

   c. (2 pts) Convert 0011 0010 from binary to base 10

   d. (2 pts) Convert 31 from base 8 to binary

   e. (2 pts) Convert 0100 0111 from base 2 to base 10

   f. (2 pts) Convert 1000 1011 0101 0101 from binary to base 16

   g. (2 pts) Convert 100 from decimal to base 2

   h. (2 pts) Convert 1011 1010 from binary to base 10

   i. (2 pts) Convert 1101 1010 0101 0010 from binary to base 16

   j. (2 pts) Convert 0101 1110 0001 1011 from base 2 to hexadecimal

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing kiwi lime 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][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[]) {
  char e;
  int f;
  double g;
  Node h;
  char *p;
  int *q;
  double *r;
  Node *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s->data

b. (2 pts) s->next

c. (2 pts) argv[1][2]

d. (2 pts) p

e. (2 pts) argv[0]

f. (2 pts) *r

g. (2 pts) &r

h. (2 pts) argc

i. (2 pts) h

j. (2 pts) s->next->next

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0111 0001 0001 1000 from binary to hexadecimal

   b. (2 pts) Convert 100 010 from base 2 to octal

   c. (2 pts) Convert 0110 1000 1100 1001 from binary to base 16

   d. (2 pts) Convert 110 000 111 from base 2 to base 8

   e. (2 pts) Convert 157 from base 10 to binary

   f. (2 pts) Convert 1011 1101 from binary to decimal

   g. (2 pts) Convert 011 001 100 from base 2 to octal

   h. (2 pts) Convert 0010 0010 from base 2 to base 10

   i. (2 pts) Convert 011 111 000 from base 2 to base 8

   j. (2 pts) Convert 42 from base 8 to base 2

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing mango kiwi
```

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][2]?

d. (2 pts) What is the value of argv[2][1]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  char x;
  double y;
  int z;
  Node a;
  char *b;
  double *c;
  int *d;
  Node *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) e->next->next

b. (2 pts) e->next

c. (2 pts) x

d. (2 pts) argv[0]

e. (2 pts) argc

f. (2 pts) &y

g. (2 pts) b

h. (2 pts) argv[1][2]

i. (2 pts) *d

j. (2 pts) e->data

k. (2 pts) &e

4. (20 pts) Given the following struct definition:

```
struct Precip {
  int day;
  double inches;
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
    - Each exam is numbered (e.g. Exam #137).
    - Each pages is numbered (e.g. Page 1, Page 2, etc.)
    - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 101 101 011 from base 2 to base 8

    b. (2 pts) Convert 1100 1011 from base 2 to decimal

    c. (2 pts) Convert 56a9 from base 16 to base 2

    d. (2 pts) Convert 010 011 111 from base 2 to octal

    e. (2 pts) Convert 1000 0000 1010 0100 from base 2 to hexadecimal

    f. (2 pts) Convert 010 101 111 from base 2 to octal

    g. (2 pts) Convert 100 000 from binary to base 8

    h. (2 pts) Convert 1010 1010 1010 1010 from binary to base 16

    i. (2 pts) Convert 21 from octal to binary

    j. (2 pts) Convert 11 from octal to base 2

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing lemon grape apple banana
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[1][1]?

c. (2 pts) What is the value of argv[0][5]?

d. (2 pts) What is the value of argv[2][2]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double s;
  int t;
  char w;
  Node x;
  double *y;
  int *z;
  char *a;
  Node *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &s

b. (2 pts) argv[1][2]

c. (2 pts) argc

d. (2 pts) b->data

e. (2 pts) x

f. (2 pts) b->next->next

g. (2 pts) b->next

h. (2 pts) *b

i. (2 pts) &a

j. (2 pts) b

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ___________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

    a. (2 pts) Convert 011 100 from binary to base 8

    b. (2 pts) Convert 011 011 100 from base 2 to base 8

    c. (2 pts) Convert 1111 1100 0000 1011 from binary to hexadecimal

    d. (2 pts) Convert 1011 0100 0010 from binary to hexadecimal

    e. (2 pts) Convert 111 100 010 from base 2 to base 8

    f. (2 pts) Convert 0101 1000 from base 2 to decimal

    g. (2 pts) Convert 61ea from hexadecimal to binary

    h. (2 pts) Convert 24 from base 8 to base 2

    i. (2 pts) Convert 16 from base 8 to binary

    j. (2 pts) Convert 0011 0001 1101 1010 from base 2 to hexadecimal

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing guava fig apple kiwi
```

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][4]?

3. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int e;
  double f;
  char g;
  Node h;
  int *p;
  double *q;
  char *r;
  Node *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s->next

b. (2 pts) g

c. (2 pts) *s

d. (2 pts) argv[0]

e. (2 pts) q

f. (2 pts) &e

g. (2 pts) s->data

h. (2 pts) s->next->next

i. (2 pts) argv[1][2]

j. (2 pts) argc

k. (2 pts) &p

4. (20 pts) Given the following struct definition:

```
struct Precip {
  int day;
  double inches;
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100

Color in first initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1,2)	first name initial	last name initial
Color in last initial:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

CS16—Midterm Exam

E02, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/03/2014

Name: ____

Umail Address: ______________________________@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
    - Each exam is numbered (e.g. Exam #137).
    - Each pages is numbered (e.g. Page 1, Page 2, etc.)
    - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 5333 from base 16 to base 2

   b. (2 pts) Convert 40 from decimal to binary

   c. (2 pts) Convert 1110 1001 1110 1100 from base 2 to hexadecimal

   d. (2 pts) Convert 1001 0111 0010 1100 from binary to base 16

   e. (2 pts) Convert 212 from base 10 to binary

   f. (2 pts) Convert 1111 0010 from binary to base 10

   g. (2 pts) Convert bda from base 16 to binary

   h. (2 pts) Convert 1101 1000 from binary to base 10

   i. (2 pts) Convert 0010 1010 1110 from binary to base 16

   j. (2 pts) Convert 1100 1101 from binary to decimal

2. Assume the `main` function in the program `thing.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./thing mango kiwi cherry
```

   a. (2 pts) What is the value of `argc` in this case?

   b. (2 pts) What is the value of argv[0][4]?

   c. (2 pts) What is the value of argv[2][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[]) {
  char b;
  int c;
  double d;
  Node e;
  char *f;
  int *g;
  double *h;
  Node *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc

b. (2 pts) p->data

c. (2 pts) *f

d. (2 pts) p->next

e. (2 pts) f

f. (2 pts) &p

g. (2 pts) &e

h. (2 pts) b

i. (2 pts) p->next->next

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 prameters to the function and the return value should be as described in the comment.

```
// days is an array with a month's worth of Precip structs
// numDays is the number of days in that month
// return the total rainfall of all days in the month.
double totalRainfall(Precip *days, int numDays);
```

Answer in the space below

5. (20 pts) Given the following struct for representing Complex numbers (which have a real part and an imaginary part):

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

Write the full function definition for a function that would have the following prototype. The prameters to the function and the return value should be as described in the comment.

Note that you MUST follow the struct definition given here; pay close attention to the names of both the members of the struct, and the parameters to the function.

Also note that the parameter p is a *pointer* and write your code accordingly.

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

Answer in the space below.

6. (10 pts) Given the same struct definition as in the previous problem:

```
// Complex number, e.g. a+bi
struct Complex {
  double real; // the a part
  double imag; // the b part
};
```

And given the same function prototype:

```
// p is a pointer to a Complex number struct
// a is the real part of the number.
// b is the imag part (coefficient of i)
void initComplex(Complex *p, double a, double b);
```

And given the following prototype, for a function you are NOT required to write, but may assume is ALREADY DEFINED:

```
string complexToString(Complex c);
```

Fill in the missing line of code in the main program below after the comment that says TODO.

You may assume that the header file `complex.h` contains the struct definition and the function prototype given above.

```
#include <iostream>
using namespace std;

#include "complex.h"

int main() {
  Complex c;

  // TODO: Write a function call to initComplex that sets
  // the real part to 2.3 and the imaginary part to 4.5






  // Show result

  cout << "c=" << complexToString(c) << endl;
  return 0;
}
```

End of Exam

total points=100