

1

Exam #101 Page: 1 Name: _____

101

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #101 Page: 2 Name: _____

101

1. a. (2 pts) Convert 1111 0010 from base 2 to hexadecimal **f2**

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

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

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

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

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

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

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

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

3

Exam #101 Page: 3 Name: _____

101

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -1, what is this number's binary representation in 8-bit two's complement?

11111111

- c. (3 pts)

Given that 11010001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-47

- d. (3 pts)

Given that 10111101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-67

4

Exam #101 Page: 4 Name: _____

101

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lemon grape apple
```

a. (3 pts) What is the value of `argc` in this case? **4**

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

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

d. (3 pts) What is the value of `argv[0][5]`? **I**

4. Given the following declarations:

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

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

- a. (3 pts) `argv[1][2]` **char**
- b. (3 pts) `d` **char ***
- c. (3 pts) `argc` **int**
- d. (3 pts) `a->data` **int**
- e. (3 pts) `&y` **double ***
- f. (3 pts) `w` **Node**
- g. (3 pts) `a->next` **Node ***
- h. (3 pts) `argv[0]` **char ***
- i. (3 pts) `&d` **char ****
- j. (3 pts) `*d` **char**
- k. (3 pts) `a->next->next` **Node ***

6

Exam #101 Page: 6 Name: _____

101

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #102 Page: 1 Name: _____

102

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #102 Page: 2 Name: _____

102

1. a. (2 pts) Convert 3254 from hexadecimal to base 2 **0011 0010 0101 0100**

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

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

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

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

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

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

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

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

3

Exam #102 Page: 3 Name: _____

102

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -26, what is this number's binary representation in 8-bit two's complement?

11100110

c. (3 pts)

Given that 10010010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-110

d. (3 pts)

Given that 11110001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-15

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi apple
```

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

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

c. (3 pts) What is the value of `argv[2][4]`? e

d. (3 pts) What is the value of `argv[0][6]`? t

4. Given the following declarations:

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

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

- a. (3 pts) `b->next` **Node ***
- b. (3 pts) `&b` **Node ****
- c. (3 pts) `b->data` **int**
- d. (3 pts) `&x` **Node ***
- e. (3 pts) `argv[0]` **char ***
- f. (3 pts) `*e` **char**
- g. (3 pts) `argc` **int**
- h. (3 pts) `b->next->next` **Node ***
- i. (3 pts) `argv[1][2]` **char**
- j. (3 pts) `d` **int ***
- k. (3 pts) `z` **int**

6

Exam #102 Page: 6 Name: _____

102

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #103 Page: 2 Name: _____

103

-
1. a. (2 pts) Convert 35 from base 8 to binary **011 101**
- b. (2 pts) Convert 010 011 001 from binary to octal **231**
- c. (2 pts) Convert c5d8 from hexadecimal to base 2 **1100 0101 1101 1000**
- d. (2 pts) Convert 101 110 001 from binary to octal **561**
- e. (2 pts) Convert 1110 1100 0000 1101 from binary to hexadecimal **ec0d**
- f. (2 pts) Convert 110 100 001 from binary to octal **641**
- g. (2 pts) Convert 33 from octal to base 2 **011 011**
- h. (2 pts) Convert 0111 1011 from binary to decimal **123**
- i. (2 pts) Convert 0110 0111 from base 2 to base 10 **103**

3

Exam #103 Page: 3 Name: _____

103

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -60, what is this number's binary representation in 8-bit two's complement?

11000100

- c. (3 pts)

Given that 11111100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-4

- d. (3 pts)

Given that 10001010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-118

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

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

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

```
./runIt banana kiwi apple fig
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) p->next **Node ***
- b. (3 pts) p->data **int**
- c. (3 pts) c **double**
- d. (3 pts) &f **char ***
- e. (3 pts) *g **double**
- f. (3 pts) argv[1][2] **char**
- g. (3 pts) p->next->next **Node ***
- h. (3 pts) argc **int**
- i. (3 pts) argv[0] **char ***
- j. (3 pts) h **int ***
- k. (3 pts) &q **char ****

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #104 Page: 1 Name: _____

104

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #104 Page: 2 Name: _____

104

1. a. (2 pts) Convert 63 from octal to binary **110 011**

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

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

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

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

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

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

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

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

3

Exam #104 Page: 3 Name: _____

104

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -104, what is this number's binary representation in 8-bit two's complement?

10011000

- c. (3 pts)

Given that 10010000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-112

- d. (3 pts)

Given that 10001010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-118

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple cherry mango
```

a. (3 pts) What is the value of `argc` in this case? 4

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

c. (3 pts) What is the value of `argv[0][1]`? /

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

4. Given the following declarations:

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

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

- a. (3 pts) a->next->next **Node ***
- b. (3 pts) argv[0] **char ***
- c. (3 pts) a->next **Node ***
- d. (3 pts) &a **Node ****
- e. (3 pts) y **char**
- f. (3 pts) &y **char ***
- g. (3 pts) argc **int**
- h. (3 pts) *z **int**
- i. (3 pts) c **char ***
- j. (3 pts) argv[1][2] **char**
- k. (3 pts) a->data **int**

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #105 Page: 1 Name: _____

105

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #105 Page: 2 Name: _____

105

1. a. (2 pts) Convert 20 from base 10 to binary **0001 0100**

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

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

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

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

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

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

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

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

3

Exam #105 Page: 3 Name: _____

105

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -11, what is this number's binary representation in 8-bit two's complement?

11110101

- c. (3 pts)

Given that 11111010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-6

- d. (3 pts)

Given that 10111101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-67

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple kiwi lemon
```

a. (3 pts) What is the value of `argc` in this case? 4

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

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

d. (3 pts) What is the value of `argv[1][4]`? e

4. Given the following declarations:

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

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

- a. (3 pts) `*x` **int**
- b. (3 pts) `argv[1][2]` **char**
- c. (3 pts) `w->data` **int**
- d. (3 pts) `w->next` **Node ***
- e. (3 pts) `argc` **int**
- f. (3 pts) `&w` **Node ****
- g. (3 pts) `&s` **double ***
- h. (3 pts) `w->next->next` **Node ***
- i. (3 pts) `s` **double**
- j. (3 pts) `argv[0]` **char ***
- k. (3 pts) `y` **double ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #106 Page: 1 Name: _____

106

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #106 Page: 2 Name: _____

106

-
1. a. (2 pts) Convert 010 001 011 from binary to base 8 **213**
- b. (2 pts) Convert 36 from base 8 to base 2 **011 110**
- c. (2 pts) Convert 100 011 110 from binary to base 8 **436**
- d. (2 pts) Convert 27 from base 8 to base 2 **010 111**
- e. (2 pts) Convert 95ef from hexadecimal to binary **1001 0101 1110 1111**
- f. (2 pts) Convert 9e20 from base 16 to base 2 **1001 1110 0010 0000**
- g. (2 pts) Convert 106 from base 10 to base 2 **0110 1010**
- h. (2 pts) Convert 0001 0010 from base 2 to decimal **18**
- i. (2 pts) Convert 110 001 011 from binary to octal **613**

3

Exam #106 Page: 3 Name: _____

106

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -35, what is this number's binary representation in 8-bit two's complement?

11011101

- c. (3 pts)

Given that 10111011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-69

- d. (3 pts)

Given that 11010111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-41

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi lemon
```

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

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `w` `char`
- b. (3 pts) `x->data` `int`
- c. (3 pts) `argv[0]` `char *`
- d. (3 pts) `argv[1][2]` `char`
- e. (3 pts) `x->next` `Node *`
- f. (3 pts) `y` `double *`
- g. (3 pts) `argc` `int`
- h. (3 pts) `&y` `double **`
- i. (3 pts) `x->next->next` `Node *`
- j. (3 pts) `*y` `double`
- k. (3 pts) `&s` `double *`

6

Exam #106 Page: 6 Name: _____

106

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #107 Page: 2 Name: _____

107

1. a. (2 pts) Convert 8ab0 from hexadecimal to base 2 **1000 1010 1011 0000**

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

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

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

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

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

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

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

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

3

Exam #107 Page: 3 Name: _____

107

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -70, what is this number's binary representation in 8-bit two's complement?

10111010

- c. (3 pts)

Given that 10100101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-91

- d. (3 pts)

Given that 11110000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-16

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

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

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

```
./runIt apple date cherry kiwi
```

a. (3 pts) What is the value of `argc` in this case? 5

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

c. (3 pts) What is the value of `argv[1][0]`? a

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

4. Given the following declarations:

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

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

- a. (3 pts) *w **Node**
- b. (3 pts) argv[0] **char ***
- c. (3 pts) w->data **int**
- d. (3 pts) &x **char ****
- e. (3 pts) p **double**
- f. (3 pts) s **int ***
- g. (3 pts) w->next **Node ***
- h. (3 pts) argv[1][2] **char**
- i. (3 pts) &h **int ***
- j. (3 pts) w->next->next **Node ***
- k. (3 pts) argc **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #108 Page: 1 Name: _____

108

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #108 Page: 2 Name: _____

108

-
1. a. (2 pts) Convert e30d from hexadecimal to base 2 **1110 0011 0000 1101**
- b. (2 pts) Convert 110 110 000 from base 2 to octal **660**
- c. (2 pts) Convert 10 from base 8 to base 2 **001 000**
- d. (2 pts) Convert 1010 0011 from base 2 to decimal **163**
- e. (2 pts) Convert e9b3 from base 16 to binary **1110 1001 1011 0011**
- f. (2 pts) Convert 001 110 010 from binary to octal **162**
- g. (2 pts) Convert 0110 0110 0000 1001 from base 2 to base 16 **6609**
- h. (2 pts) Convert 010 000 001 from base 2 to octal **201**
- i. (2 pts) Convert 1000 0010 from base 2 to decimal **130**

3

Exam #108 Page: 3 Name: _____

108

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -114, what is this number's binary representation in 8-bit two's complement?

10001110

- c. (3 pts)

Given that 10111001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-71

- d. (3 pts)

Given that 11110000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-16

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

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

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

```
./runIt banana lemon lime
```

a. (3 pts) What is the value of `argc` in this case? 4

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

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

d. (3 pts) What is the value of `argv[1][5]`? a

4. Given the following declarations:

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

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

- a. (3 pts) `&a` **int ***
- b. (3 pts) `&g` **double ****
- c. (3 pts) `argv[1][2]` **char**
- d. (3 pts) `a` **int**
- e. (3 pts) `f->data` **int**
- f. (3 pts) `argc` **int**
- g. (3 pts) `f->next` **Node ***
- h. (3 pts) `argv[0]` **char ***
- i. (3 pts) `*g` **double**
- j. (3 pts) `h` **char ***
- k. (3 pts) `f->next->next` **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #109 Page: 1 Name: _____

109

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #109 Page: 2 Name: _____

109

1. a. (2 pts) Convert 001 001 111 from binary to octal **117**

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

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

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

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

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

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

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

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

3

Exam #109 Page: 3 Name: _____

109

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -20, what is this number's binary representation in 8-bit two's complement?

11101100

c. (3 pts)

Given that 10001001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-119

d. (3 pts)

Given that 10011001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-103

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lemon grape banana
```

a. (3 pts) What is the value of `argc` in this case? 4

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

c. (3 pts) What is the value of `argv[0][1]`? /

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

4. Given the following declarations:

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

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

- a. (3 pts) argv[1][2] **char**
- b. (3 pts) q **char**
- c. (3 pts) &q **char ***
- d. (3 pts) *w **char**
- e. (3 pts) s->data **int**
- f. (3 pts) argc **int**
- g. (3 pts) argv[0] **char ***
- h. (3 pts) s->next **Node ***
- i. (3 pts) s->next->next **Node ***
- j. (3 pts) t **double ***
- k. (3 pts) &r **int ****

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #110 Page: 1 Name: _____

110

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #110 Page: 2 Name: _____

110

-
1. a. (2 pts) Convert 0101 1001 0100 1110 from base 2 to hexadecimal **594e**

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

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

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

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

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

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

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

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

3

Exam #110 Page: 3 Name: _____

110

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -45, what is this number's binary representation in 8-bit two's complement?

11010011

- c. (3 pts)

Given that 11100100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-28

- d. (3 pts)

Given that 10111101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-67

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

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

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

```
./runIt cherry grape
```

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

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

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

d. (3 pts) What is the value of `argv[0][4]`? n

4. Given the following declarations:

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

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

- a. (3 pts) s **Node ***
- b. (3 pts) argv[0] **char ***
- c. (3 pts) &h **Node ***
- d. (3 pts) g **double**
- e. (3 pts) s->data **int**
- f. (3 pts) &t **int ****
- g. (3 pts) s->next->next **Node ***
- h. (3 pts) argc **int**
- i. (3 pts) argv[1][2] **char**
- j. (3 pts) s->next **Node ***
- k. (3 pts) *w **char**

6

Exam #110 Page: 6 Name: _____

110

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #111 Page: 1 Name: _____

111

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #111 Page: 2 Name: _____

111

-
1. a. (2 pts) Convert 1001 1110 from base 2 to base 10 **158**
- b. (2 pts) Convert 1e29 from base 16 to base 2 **0001 1110 0010 1001**
- c. (2 pts) Convert 0011 0100 from base 2 to decimal **52**
- d. (2 pts) Convert 17d2 from base 16 to base 2 **0001 0111 1101 0010**
- e. (2 pts) Convert 6 from base 10 to base 2 **0110**
- f. (2 pts) Convert 1001 1111 from binary to base 10 **159**
- g. (2 pts) Convert 188 from decimal to base 2 **1011 1100**
- h. (2 pts) Convert 1011 1001 0001 0011 from base 2 to hexadecimal **b913**
- i. (2 pts) Convert 101 110 000 from binary to base 8 **560**

3

Exam #111 Page: 3 Name: _____

111

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -80, what is this number's binary representation in 8-bit two's complement?

10110000

- c. (3 pts)

Given that 11001111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-49

- d. (3 pts)

Given that 11010110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-42

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

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

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

```
./runIt date banana guava lime
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `h` **double ***
- b. (3 pts) `argc` **int**
- c. (3 pts) `q->next->next` **Node ***
- d. (3 pts) `q->next` **Node ***
- e. (3 pts) `&h` **double ****
- f. (3 pts) `*h` **double**
- g. (3 pts) `q->data` **int**
- h. (3 pts) `&e` **int ***
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `f` **Node**
- k. (3 pts) `argv[1][2]` **char**

6

Exam #111 Page: 6 Name: _____

111

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

7

Exam #111 Page: 7 Name: _____

111

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #112 Page: 1 Name: _____

112

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #112 Page: 2 Name: _____

112

-
1. a. (2 pts) Convert 010 110 101 from binary to octal **265**
- b. (2 pts) Convert 7bdc from base 16 to base 2 **0111 1011 1101 1100**
- c. (2 pts) Convert 7 from base 8 to base 2 **111**
- d. (2 pts) Convert 0101 0110 0001 1111 from base 2 to hexadecimal **561f**
- e. (2 pts) Convert 011 010 from binary to base 8 **32**
- f. (2 pts) Convert 011 001 010 from binary to octal **312**
- g. (2 pts) Convert cfb7 from hexadecimal to base 2 **1100 1111 1011 0111**
- h. (2 pts) Convert 0110 1001 from base 2 to base 10 **105**
- i. (2 pts) Convert 103 from base 10 to base 2 **0110 0111**

3

Exam #112 Page: 3 Name: _____

112

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -46, what is this number's binary representation in 8-bit two's complement?

11010010

- c. (3 pts)

Given that 10011001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-103

- d. (3 pts)

Given that 10100111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-89

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit banana cherry fig
```

a. (3 pts) What is the value of `argc` in this case? 4

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `y->data` **int**
- b. (3 pts) `y->next` **Node ***
- c. (3 pts) `argv[1][2]` **char**
- d. (3 pts) `y` **Node ***
- e. (3 pts) `s` **Node**
- f. (3 pts) `argv[0]` **char ***
- g. (3 pts) `&y` **Node ****
- h. (3 pts) `argc` **int**
- i. (3 pts) `&r` **int ***
- j. (3 pts) `y->next->next` **Node ***
- k. (3 pts) `*x` **int**

6

Exam #112 Page: 6 Name: _____

112

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #113 Page: 1 Name: _____

113

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #113 Page: 2 Name: _____

113

1. a. (2 pts) Convert 9f80 from hexadecimal to base 2 **1001 1111 1000 0000**

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

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

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

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

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

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

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

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -80, what is this number's binary representation in 8-bit two's complement?

10110000

c. (3 pts)

Given that 10000011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-125

d. (3 pts)

Given that 11000000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-64

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit fig date
```

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

b. (3 pts) What is the value of `argv[0][1]`? **/**

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

d. (3 pts) What is the value of `argv[1][0]`? **f**

4. Given the following declarations:

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

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

- a. (3 pts) `&r` `char *`
- b. (3 pts) `argc` `int`
- c. (3 pts) `s->data` `int`
- d. (3 pts) `argv[0]` `char *`
- e. (3 pts) `s->next` `Node *`
- f. (3 pts) `&s` `Node **`
- g. (3 pts) `s->next->next` `Node *`
- h. (3 pts) `s` `Node *`
- i. (3 pts) `r` `char`
- j. (3 pts) `*w` `int`
- k. (3 pts) `argv[1][2]` `char`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #114 Page: 2 Name: _____

114

-
1. a. (2 pts) Convert 208 from base 10 to base 2 **1101 0000**
- b. (2 pts) Convert 001 111 from binary to base 8 **17**
- c. (2 pts) Convert 0100 0010 from binary to base 10 **66**
- d. (2 pts) Convert 17 from base 8 to binary **001 111**
- e. (2 pts) Convert 46de from hexadecimal to base 2 **0100 0110 1101 1110**
- f. (2 pts) Convert 001 100 010 from base 2 to base 8 **142**
- g. (2 pts) Convert 123 from decimal to base 2 **0111 1011**
- h. (2 pts) Convert 90 from decimal to base 2 **0101 1010**
- i. (2 pts) Convert 211 from base 10 to base 2 **1101 0011**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -105, what is this number's binary representation in 8-bit two's complement?

10010111

c. (3 pts)

Given that 11000100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-60

d. (3 pts)

Given that 11110100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-12

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit date mango guava fig
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

d. (3 pts) What is the value of `argv[0][6]`? t

4. Given the following declarations:

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

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

- a. (3 pts) `argv[0]` `char *`
- b. (3 pts) `w->next` `Node *`
- c. (3 pts) `&t` `int **`
- d. (3 pts) `h` `double`
- e. (3 pts) `&q` `Node *`
- f. (3 pts) `w->next->next` `Node *`
- g. (3 pts) `argv[1][2]` `char`
- h. (3 pts) `*t` `int`
- i. (3 pts) `argc` `int`
- j. (3 pts) `s` `double *`
- k. (3 pts) `w->data` `int`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #115 Page: 1 Name: _____

115

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #115 Page: 2 Name: _____

115

1. a. (2 pts) Convert 0001 0101 1100 0001 from binary to hexadecimal **15c1**

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

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

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

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

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

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

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

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -11, what is this number's binary representation in 8-bit two's complement?

11110101

c. (3 pts)

Given that 10101110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-82

d. (3 pts)

Given that 10001101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-115

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

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

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

```
./runIt kiwi date mango apple
```

a. (3 pts) What is the value of `argc` in this case? 5

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

c. (3 pts) What is the value of `argv[1][0]`? k

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

4. Given the following declarations:

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

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

- a. (3 pts) `argv[1][2]` **char**
- b. (3 pts) `*d` **char**
- c. (3 pts) `c->next` **Node ***
- d. (3 pts) `argv[0]` **char ***
- e. (3 pts) `argc` **int**
- f. (3 pts) `c->data` **int**
- g. (3 pts) `c->next->next` **Node ***
- h. (3 pts) `y` **Node**
- i. (3 pts) `&d` **char ****
- j. (3 pts) `&x` **double ***
- k. (3 pts) `d` **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #116 Page: 1 Name: _____

116

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #116 Page: 2 Name: _____

116

-
1. a. (2 pts) Convert 0110 1110 0001 1110 from base 2 to hexadecimal **6e1e**
- b. (2 pts) Convert 31 from base 8 to binary **011 001**
- c. (2 pts) Convert 110 101 011 from binary to octal **653**
- d. (2 pts) Convert 1000 0101 from base 2 to decimal **133**
- e. (2 pts) Convert 1001 1010 from binary to base 10 **154**
- f. (2 pts) Convert cc98 from base 16 to binary **1100 1100 1001 1000**
- g. (2 pts) Convert 35 from octal to base 2 **011 101**
- h. (2 pts) Convert 1000 1000 from binary to base 10 **136**
- i. (2 pts) Convert 1001 0000 from base 2 to base 10 **144**

3

Exam #116 Page: 3 Name: _____

116

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -56, what is this number's binary representation in 8-bit two's complement?

11001000

- c. (3 pts)

Given that 11000010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-62

- d. (3 pts)

Given that 10001101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-115

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lemon grape cherry
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][1]`? /

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

d. (3 pts) What is the value of `argv[1][0]`? l

4. Given the following declarations:

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

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

- a. (3 pts) `p` **double**
- b. (3 pts) `argv[0]` **char ***
- c. (3 pts) `*t` **double**
- d. (3 pts) `&h` **int ***
- e. (3 pts) `s` **int ***
- f. (3 pts) `argc` **int**
- g. (3 pts) `&t` **double ****
- h. (3 pts) `r->next->next` **Node ***
- i. (3 pts) `r->data` **int**
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `r->next` **Node ***

6

Exam #116 Page: 6 Name: _____

116

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #117 Page: 2 Name: _____

117

1. a. (2 pts) Convert 1011 0010 from binary to decimal **178**

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

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

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

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

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

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

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

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -90, what is this number's binary representation in 8-bit two's complement?

10100110

c. (3 pts)

Given that 10101100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-84

d. (3 pts)

Given that 11110001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-15

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit cherry fig
```

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

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

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

d. (3 pts) What is the value of `argv[1][5]`? y

4. Given the following declarations:

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

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

- a. (3 pts) h->data **int**
- b. (3 pts) &h **Node ****
- c. (3 pts) argv[1][2] **char**
- d. (3 pts) &d **Node ***
- e. (3 pts) h **Node ***
- f. (3 pts) h->next->next **Node ***
- g. (3 pts) argc **int**
- h. (3 pts) *h **Node**
- i. (3 pts) argv[0] **char ***
- j. (3 pts) h->next **Node ***
- k. (3 pts) d **Node**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #118 Page: 2 Name: _____

118

1. a. (2 pts) Convert 71 from base 8 to base 2 **111 001**

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

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

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

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

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

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

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

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -115, what is this number's binary representation in 8-bit two's complement?

10001101

c. (3 pts)

Given that 11101101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-19

d. (3 pts)

Given that 11011010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-38

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit guava apple fig date
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][4]`? a

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

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

4. Given the following declarations:

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

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

- a. (3 pts) r->next->next **Node ***
- b. (3 pts) r->data **int**
- c. (3 pts) f **int**
- d. (3 pts) s **char ***
- e. (3 pts) argv[1][2] **char**
- f. (3 pts) *s **char**
- g. (3 pts) &r **Node ****
- h. (3 pts) argv[0] **char ***
- i. (3 pts) r->next **Node ***
- j. (3 pts) argc **int**
- k. (3 pts) &g **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #119 Page: 1 Name: _____

119

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #119 Page: 2 Name: _____

119

1. a. (2 pts) Convert 41 from base 10 to base 2 **0010 1001**

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

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

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

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

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

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

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

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -21, what is this number's binary representation in 8-bit two's complement?

11101011

c. (3 pts)

Given that 11010111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-41

d. (3 pts)

Given that 11110011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-13

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit mango cherry banana guava
```

a. (3 pts) What is the value of `argc` in this case? **5**

b. (3 pts) What is the value of `argv[0][4]`? **n**

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `argv[0]` **char ***
- b. (3 pts) `*f` **int**
- c. (3 pts) `g` **Node ***
- d. (3 pts) `&b` **int ***
- e. (3 pts) `g->data` **int**
- f. (3 pts) `g->next->next` **Node ***
- g. (3 pts) `d` **char**
- h. (3 pts) `&e` **double ****
- i. (3 pts) `g->next` **Node ***
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `argc` **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #120 Page: 1 Name: _____

120

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #120 Page: 2 Name: _____

120

1. a. (2 pts) Convert 129 from base 10 to base 2 **1000 0001**

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

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

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

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

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

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

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

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

3

Exam #120 Page: 3 Name: _____

120

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -65, what is this number's binary representation in 8-bit two's complement?

10111111

- c. (3 pts)

Given that 11101011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-21

- d. (3 pts)

Given that 11110010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-14

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit date apple guava
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][6]`? t

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

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

4. Given the following declarations:

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

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

- a. (3 pts) w **char**
- b. (3 pts) x->next **Node ***
- c. (3 pts) argc **int**
- d. (3 pts) &a **char ****
- e. (3 pts) *x **Node**
- f. (3 pts) argv[0] **char ***
- g. (3 pts) argv[1][2] **char**
- h. (3 pts) &s **int ***
- i. (3 pts) x->data **int**
- j. (3 pts) y **int ***
- k. (3 pts) x->next->next **Node ***

6

Exam #120 Page: 6 Name: _____

120

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #121 Page: 1 Name: _____

121

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #121 Page: 2 Name: _____

121

1. a. (2 pts) Convert 61 from base 8 to binary **110 001**

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

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

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

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

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

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

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

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

3

Exam #121 Page: 3 Name: _____

121

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -100, what is this number's binary representation in 8-bit two's complement?

10011100

- c. (3 pts)

Given that 11010101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-43

- d. (3 pts)

Given that 10001100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-116

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit date fig
```

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

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

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

d. (3 pts) What is the value of `argv[0][4]`? **n**

4. Given the following declarations:

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

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

- a. (3 pts) `e->next` **Node ***
- b. (3 pts) `d` **double ***
- c. (3 pts) `argc` **int**
- d. (3 pts) `e->data` **int**
- e. (3 pts) `e->next->next` **Node ***
- f. (3 pts) `argv[1][2]` **char**
- g. (3 pts) `*f` **int**
- h. (3 pts) `argv[0]` **char ***
- i. (3 pts) `a` **Node**
- j. (3 pts) `&e` **Node ****
- k. (3 pts) `&z` **double ***

6

Exam #121 Page: 6 Name: _____

121

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #122 Page: 1 Name: _____

122

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #122 Page: 2 Name: _____

122

-
1. a. (2 pts) Convert f7dc from hexadecimal to binary **1111 0111 1101 1100**
- b. (2 pts) Convert d8d1 from hexadecimal to binary **1101 1000 1101 0001**
- c. (2 pts) Convert 54 from base 8 to binary **101 100**
- d. (2 pts) Convert 157 from base 10 to base 2 **1001 1101**
- e. (2 pts) Convert 0110 0001 0011 1001 from binary to hexadecimal **6139**
- f. (2 pts) Convert 0000 from binary to base 10 **0**
- g. (2 pts) Convert 110 010 110 from binary to base 8 **626**
- h. (2 pts) Convert 1001 0111 1001 1011 from base 2 to base 16 **979b**
- i. (2 pts) Convert 0010 0100 from base 2 to base 10 **36**

3

Exam #122 Page: 3 Name: _____

122

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -124, what is this number's binary representation in 8-bit two's complement?

10000100

- c. (3 pts)

Given that 10010110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-106

- d. (3 pts)

Given that 10111111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-65

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

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

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

```
./runIt fig grape lime lemon
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

d. (3 pts) What is the value of `argv[0][6]`? t

4. Given the following declarations:

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

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

- a. (3 pts) `g->data` **int**
- b. (3 pts) `&g` **Node ****
- c. (3 pts) `g->next` **Node ***
- d. (3 pts) `argv[0]` **char ***
- e. (3 pts) `argc` **int**
- f. (3 pts) `argv[1][2]` **char**
- g. (3 pts) `&d` **char ***
- h. (3 pts) `h` **char ***
- i. (3 pts) `g->next->next` **Node ***
- j. (3 pts) `c` **Node**
- k. (3 pts) `*f` **double**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #123 Page: 1 Name: _____

123

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #123 Page: 2 Name: _____

123

1. a. (2 pts) Convert 001 111 001 from binary to base 8 **171**

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

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

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

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

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

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

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

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

3

Exam #123 Page: 3 Name: _____

123

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -31, what is this number's binary representation in 8-bit two's complement?

11100001

- c. (3 pts)

Given that 10000001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-127

- d. (3 pts)

Given that 11011001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-39

4

Exam #123 Page: 4 Name: _____

123

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi grape guava cherry
```

a. (3 pts) What is the value of `argc` in this case? **5**

b. (3 pts) What is the value of `argv[1][0]`? **k**

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `y` **double**
- b. (3 pts) `argv[0]` **char ***
- c. (3 pts) `&z` **Node ***
- d. (3 pts) `d->next` **Node ***
- e. (3 pts) `*e` **char**
- f. (3 pts) `d` **Node ***
- g. (3 pts) `d->next->next` **Node ***
- h. (3 pts) `argc` **int**
- i. (3 pts) `&c` **double ****
- j. (3 pts) `d->data` **int**
- k. (3 pts) `argv[1][2]` **char**

6

Exam #123 Page: 6 Name: _____

123

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

7

Exam #123 Page: 7 Name: _____

123

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #124 Page: 2 Name: _____

124

1. a. (2 pts) Convert 100 101 010 from binary to base 8 **452**

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

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

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

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

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

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

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

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

3

Exam #124 Page: 3 Name: _____

124

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -75, what is this number's binary representation in 8-bit two's complement?

10110101

- c. (3 pts)

Given that 10010100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-108

- d. (3 pts)

Given that 11011000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-40

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit fig lime apple
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][5]`? I

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

d. (3 pts) What is the value of `argv[1][0]`? f

4. Given the following declarations:

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

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

- a. (3 pts) `&x` **char ****
- b. (3 pts) `argv[0]` **char ***
- c. (3 pts) `&q` **double ***
- d. (3 pts) `argv[1][2]` **char**
- e. (3 pts) `t->next` **Node ***
- f. (3 pts) `h` **int**
- g. (3 pts) `t->next->next` **Node ***
- h. (3 pts) `*w` **double**
- i. (3 pts) `t->data` **int**
- j. (3 pts) `argc` **int**
- k. (3 pts) `s` **int ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #125 Page: 1 Name: _____

125

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #125 Page: 2 Name: _____

125

1. a. (2 pts) Convert 217 from base 10 to base 2 **1101 1001**

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

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

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

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

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

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

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

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

3

Exam #125 Page: 3 Name: _____

125

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -109, what is this number's binary representation in 8-bit two's complement?

10010011

- c. (3 pts)

Given that 11111110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-2

- d. (3 pts)

Given that 11110010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-14

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

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

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

```
./runIt date grape
```

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

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

c. (3 pts) What is the value of `argv[0][4]`? **n**

d. (3 pts) What is the value of `argv[1][0]`? **d**

4. Given the following declarations:

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

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

- a. (3 pts) `&r` `int **`
- b. (3 pts) `q->next` `Node *`
- c. (3 pts) `&f` `Node *`
- d. (3 pts) `argv[0]` `char *`
- e. (3 pts) `g` `int`
- f. (3 pts) `argv[1][2]` `char`
- g. (3 pts) `q->next->next` `Node *`
- h. (3 pts) `s` `char *`
- i. (3 pts) `q->data` `int`
- j. (3 pts) `argc` `int`
- k. (3 pts) `*p` `double`

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #126 Page: 1 Name: _____

126

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #126 Page: 2 Name: _____

126

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

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

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

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

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

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

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

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

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

3

Exam #126 Page: 3 Name: _____

126

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

- b. (3 pts)

Given the decimal number -6, what is this number's binary representation in 8-bit two's complement?

1111010

- c. (3 pts)

Given that 11000000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-64

- d. (3 pts)

Given that 10100101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-91

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

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

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

```
./runIt date lime
```

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

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) argv[1][2] **char**
- b. (3 pts) *d **char**
- c. (3 pts) d **char ***
- d. (3 pts) z **char**
- e. (3 pts) argv[0] **char ***
- f. (3 pts) b->next->next **Node ***
- g. (3 pts) b->data **int**
- h. (3 pts) &d **char ****
- i. (3 pts) b->next **Node ***
- j. (3 pts) &z **char ***
- k. (3 pts) argc **int**

6

Exam #126 Page: 6 Name: _____

126

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #127 Page: 2 Name: _____

127

1. a. (2 pts) Convert 5039 from hexadecimal to binary **0101 0000 0011 1001**

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

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

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

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

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

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

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

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

b. (3 pts)

Given the decimal number -41, what is this number's binary representation in 8-bit two's complement?

11010111

c. (3 pts)

Given that 10101010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-86

d. (3 pts)

Given that 10111111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-65

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

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

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

```
./runIt guava lime cherry grape
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

d. (3 pts) What is the value of `argv[0][4]`? n

4. Given the following declarations:

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

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

- a. (3 pts) `&y` `double **`
- b. (3 pts) `y` `double *`
- c. (3 pts) `&t` `Node *`
- d. (3 pts) `z->next` `Node *`
- e. (3 pts) `argc` `int`
- f. (3 pts) `argv[1][2]` `char`
- g. (3 pts) `z->next->next` `Node *`
- h. (3 pts) `t` `Node`
- i. (3 pts) `argv[0]` `char *`
- j. (3 pts) `*y` `double`
- k. (3 pts) `z->data` `int`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #128 Page: 1 Name: _____

128

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #128 Page: 2 Name: _____

128

1. a. (2 pts) Convert 0110 0101 from binary to base 10 **101**

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

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

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

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

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

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

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

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

3

Exam #128 Page: 3 Name: _____

128

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

- b. (3 pts)

Given the decimal number -51, what is this number's binary representation in 8-bit two's complement?

11001101

- c. (3 pts)

Given that 10111011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-69

- d. (3 pts)

Given that 11110111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-9

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi cherry fig
```

a. (3 pts) What is the value of `argc` in this case? 4

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

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

d. (3 pts) What is the value of `argv[0][5]`? I

4. Given the following declarations:

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

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

- a. (3 pts) `argv[1][2]` **char**
- b. (3 pts) `&w` **char ****
- c. (3 pts) `s->data` **int**
- d. (3 pts) `s->next` **Node ***
- e. (3 pts) `*r` **int**
- f. (3 pts) `r` **int ***
- g. (3 pts) `argv[0]` **char ***
- h. (3 pts) `&h` **Node ***
- i. (3 pts) `p` **double**
- j. (3 pts) `argc` **int**
- k. (3 pts) `s->next->next` **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #129 Page: 1 Name: _____

129

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #129 Page: 2 Name: _____

129

1. a. (2 pts) Convert aa12 from hexadecimal to binary **1010 1010 0001 0010**

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

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

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

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

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

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

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

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -86, what is this number's binary representation in 8-bit two's complement?

10101010

c. (3 pts)

Given that 10100101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-91

d. (3 pts)

Given that 10010000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-112

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

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

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

```
./runIt guava banana
```

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

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `argc` **int**
- b. (3 pts) `&z` **double ****
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `s` **Node**
- e. (3 pts) `y->data` **int**
- f. (3 pts) `y->next->next` **Node ***
- g. (3 pts) `y->next` **Node ***
- h. (3 pts) `argv[1][2]` **char**
- i. (3 pts) `b` **char ***
- j. (3 pts) `*a` **int**
- k. (3 pts) `&s` **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #130 Page: 1 Name: _____

130

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #130 Page: 2 Name: _____

130

1. a. (2 pts) Convert 219 from base 10 to base 2 **1101 1011**

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

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

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

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

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

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

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

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

3

Exam #130 Page: 3 Name: _____

130

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -110, what is this number's binary representation in 8-bit two's complement?

10010010

c. (3 pts)

Given that 11100110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-26

d. (3 pts)

Given that 11000100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-60

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit banana apple cherry mango
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[0][5]`? I

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

d. (3 pts) What is the value of `argv[1][4]`? n

4. Given the following declarations:

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

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

- a. (3 pts) `b->next->next` **Node ***
- b. (3 pts) `&y` **char ***
- c. (3 pts) `b->next` **Node ***
- d. (3 pts) `argc` **int**
- e. (3 pts) `argv[1][2]` **char**
- f. (3 pts) `w` **int**
- g. (3 pts) `argv[0]` **char ***
- h. (3 pts) `b->data` **int**
- i. (3 pts) `&b` **Node ****
- j. (3 pts) `c` **char ***
- k. (3 pts) `*a` **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #131 Page: 1 Name: _____

131

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #131 Page: 2 Name: _____

131

1. a. (2 pts) Convert 0010 0000 0101 0100 from base 2 to base 16 **2054**

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

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

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

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

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

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

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

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

3

Exam #131 Page: 3 Name: _____

131

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

- b. (3 pts)

Given the decimal number -17, what is this number's binary representation in 8-bit two's complement?

11101111

- c. (3 pts)

Given that 11010000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-48

- d. (3 pts)

Given that 11011110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-34

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit mango apple banana date
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][0]`? m

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

d. (3 pts) What is the value of `argv[0][1]`? /

4. Given the following declarations:

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

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

- a. (3 pts) `y->next->next` **Node ***
- b. (3 pts) `y->data` **int**
- c. (3 pts) `*z` **char**
- d. (3 pts) `argc` **int**
- e. (3 pts) `y` **Node ***
- f. (3 pts) `q` **double**
- g. (3 pts) `y->next` **Node ***
- h. (3 pts) `argv[0]` **char ***
- i. (3 pts) `&s` **Node ***
- j. (3 pts) `&w` **double ****
- k. (3 pts) `argv[1][2]` **char**

6

Exam #131 Page: 6 Name: _____

131

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #132 Page: 1 Name: _____

132

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #132 Page: 2 Name: _____

132

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

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

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

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

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

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

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

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

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

3

Exam #132 Page: 3 Name: _____

132

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -61, what is this number's binary representation in 8-bit two's complement?

11000011

- c. (3 pts)

Given that 11100100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-28

- d. (3 pts)

Given that 11011101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-35

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

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

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

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

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][4]`? n

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `*p` **double**
- b. (3 pts) `f` **char**
- c. (3 pts) `g->next` **Node ***
- d. (3 pts) `g->next->next` **Node ***
- e. (3 pts) `&e` **double ***
- f. (3 pts) `&q` **char ****
- g. (3 pts) `g->data` **int**
- h. (3 pts) `argc` **int**
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `g` **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #133 Page: 2 Name: _____

133

1. a. (2 pts) Convert 1011 1101 from base 2 to base 10 **189**

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

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

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

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

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

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

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

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

3

Exam #133 Page: 3 Name: _____

133

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -50, what is this number's binary representation in 8-bit two's complement?

11001110

- c. (3 pts)

Given that 11110110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-10

- d. (3 pts)

Given that 11000001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-63

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

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

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

```
./runIt lime fig
```

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

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

c. (3 pts) What is the value of `argv[0][5]`? I

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

4. Given the following declarations:

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

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

- a. (3 pts) `&e` **Node ****
- b. (3 pts) `argc` **int**
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `&a` **Node ***
- e. (3 pts) `a` **Node**
- f. (3 pts) `e->next` **Node ***
- g. (3 pts) `argv[1][2]` **char**
- h. (3 pts) `g` **char ***
- i. (3 pts) `e->next->next` **Node ***
- j. (3 pts) `e->data` **int**
- k. (3 pts) `*d` **double**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #134 Page: 1 Name: _____

134

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #134 Page: 2 Name: _____

134

-
1. a. (2 pts) Convert 111 011 101 from binary to octal **735**
- b. (2 pts) Convert 1010 1111 from binary to decimal **175**
- c. (2 pts) Convert 101 100 111 from binary to octal **547**
- d. (2 pts) Convert 1011 1111 from binary to base 10 **191**
- e. (2 pts) Convert 11 from base 8 to binary **001 001**
- f. (2 pts) Convert 1001 0101 0110 0011 from binary to hexadecimal **9563**
- g. (2 pts) Convert 4 from base 8 to base 2 **100**
- h. (2 pts) Convert 0011 1001 1000 1111 from base 2 to hexadecimal **398f**
- i. (2 pts) Convert 110 010 010 from binary to base 8 **622**

3

Exam #134 Page: 3 Name: _____

134

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

- b. (3 pts)

Given the decimal number -120, what is this number's binary representation in 8-bit two's complement?

10001000

- c. (3 pts)

Given that 10001111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-113

- d. (3 pts)

Given that 10101010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-86

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lime lemon guava cherry
```

a. (3 pts) What is the value of `argc` in this case? 5

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

c. (3 pts) What is the value of `argv[2][4]`? n

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

4. Given the following declarations:

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

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

- a. (3 pts) argv[1][2] **char**
- b. (3 pts) s **char**
- c. (3 pts) argc **int**
- d. (3 pts) x **Node ***
- e. (3 pts) *y **char**
- f. (3 pts) &s **char ***
- g. (3 pts) x->data **int**
- h. (3 pts) &y **char ****
- i. (3 pts) x->next **Node ***
- j. (3 pts) x->next->next **Node ***
- k. (3 pts) argv[0] **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #135 Page: 1 Name: _____

135

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #135 Page: 2 Name: _____

135

1. a. (2 pts) Convert 51 from base 10 to base 2 **0011 0011**

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

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

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

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

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

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

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

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -26, what is this number's binary representation in 8-bit two's complement?

11100110

c. (3 pts)

Given that 11111001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-7

d. (3 pts)

Given that 11000011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-61

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

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

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

```
./runIt grape lemon lime fig
```

a. (3 pts) What is the value of `argc` in this case? 5

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

c. (3 pts) What is the value of `argv[0][6]`? t

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

4. Given the following declarations:

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

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

- a. (3 pts) argc **int**
- b. (3 pts) s->next **Node ***
- c. (3 pts) *r **int**
- d. (3 pts) argv[0] **char ***
- e. (3 pts) &p **char ***
- f. (3 pts) &r **int ****
- g. (3 pts) r **int ***
- h. (3 pts) g **int**
- i. (3 pts) s->next->next **Node ***
- j. (3 pts) argv[1][2] **char**
- k. (3 pts) s->data **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #136 Page: 1 Name: _____

136

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #136 Page: 2 Name: _____

136

-
1. a. (2 pts) Convert 140 from decimal to base 2 **1000 1100**
- b. (2 pts) Convert 11 from decimal to base 2 **1011**
- c. (2 pts) Convert 0100 0110 from binary to base 10 **70**
- d. (2 pts) Convert 728 from base 16 to base 2 **0111 0010 1000**
- e. (2 pts) Convert 7a95 from base 16 to binary **0111 1010 1001 0101**
- f. (2 pts) Convert 0011 0000 from binary to base 10 **48**
- g. (2 pts) Convert 11 from base 10 to base 2 **1011**
- h. (2 pts) Convert 0110 0111 1011 0001 from base 2 to base 16 **67b1**
- i. (2 pts) Convert 41 from octal to base 2 **100 001**

3

Exam #136 Page: 3 Name: _____

136

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -71, what is this number's binary representation in 8-bit two's complement?

10111001

- c. (3 pts)

Given that 10001101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-115

- d. (3 pts)

Given that 11000011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-61

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

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

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

```
./runIt grape fig date
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][0]`? g

c. (3 pts) What is the value of `argv[0][6]`? t

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

4. Given the following declarations:

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

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

- a. (3 pts) `c` `int *`
- b. (3 pts) `argc` `int`
- c. (3 pts) `*c` `int`
- d. (3 pts) `d->data` `int`
- e. (3 pts) `argv[0]` `char *`
- f. (3 pts) `d->next` `Node *`
- g. (3 pts) `d->next->next` `Node *`
- h. (3 pts) `argv[1][2]` `char`
- i. (3 pts) `y` `int`
- j. (3 pts) `&a` `double *`
- k. (3 pts) `&c` `int **`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. a. (2 pts) Convert 64 from base 8 to base 2 110 100
- b. (2 pts) Convert c628 from base 16 to binary 1100 0110 0010 1000
- c. (2 pts) Convert 15 from base 8 to base 2 001 101
- d. (2 pts) Convert 9312 from base 16 to base 2 1001 0011 0001 0010
- e. (2 pts) Convert 0101 1101 1110 0000 from base 2 to base 16 5de0
- f. (2 pts) Convert 110 010 100 from base 2 to base 8 624
- g. (2 pts) Convert 181 from decimal to base 2 1011 0101
- h. (2 pts) Convert 1110 1111 0110 0001 from base 2 to base 16 ef61
- i. (2 pts) Convert 010 011 111 from base 2 to octal 237

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -105, what is this number's binary representation in 8-bit two's complement?

10010111

c. (3 pts)

Given that 11110111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-9

d. (3 pts)

Given that 11011100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-36

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit banana lime
```

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

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) argc **int**
- b. (3 pts) a->next->next **Node ***
- c. (3 pts) &w **Node ***
- d. (3 pts) argv[1][2] **char**
- e. (3 pts) a->data **int**
- f. (3 pts) c **char ***
- g. (3 pts) y **char**
- h. (3 pts) *b **int**
- i. (3 pts) a->next **Node ***
- j. (3 pts) &b **int ****
- k. (3 pts) argv[0] **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #138 Page: 1 Name: _____

138

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #138 Page: 2 Name: _____

138

-
1. a. (2 pts) Convert 0010 0110 1111 from binary to hexadecimal **26f**
- b. (2 pts) Convert 1001 0111 1010 1110 from base 2 to base 16 **97ae**
- c. (2 pts) Convert 6b24 from base 16 to base 2 **0110 1011 0010 0100**
- d. (2 pts) Convert 1110 1111 0101 0011 from base 2 to base 16 **ef53**
- e. (2 pts) Convert 54 from base 8 to base 2 **101 100**
- f. (2 pts) Convert fcd6 from base 16 to binary **1111 1100 1101 0110**
- g. (2 pts) Convert 183 from base 10 to binary **1011 0111**
- h. (2 pts) Convert 0101 1000 from binary to base 10 **88**
- i. (2 pts) Convert 74 from octal to base 2 **111 100**

3

Exam #138 Page: 3 Name: _____

138

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -2, what is this number's binary representation in 8-bit two's complement?

11111110

- c. (3 pts)

Given that 10111000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-72

- d. (3 pts)

Given that 10010000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-112

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lime apple
```

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

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `r` `int *`
- b. (3 pts) `*q` `double`
- c. (3 pts) `p->next` `Node *`
- d. (3 pts) `&e` `Node *`
- e. (3 pts) `p->data` `int`
- f. (3 pts) `p->next->next` `Node *`
- g. (3 pts) `argc` `int`
- h. (3 pts) `argv[0]` `char *`
- i. (3 pts) `&p` `Node **`
- j. (3 pts) `e` `Node`
- k. (3 pts) `argv[1][2]` `char`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #139 Page: 2 Name: _____

139

1. a. (2 pts) Convert 010 001 110 from binary to octal **216**

b. (2 pts) Convert 24 from octal to binary **010 100**

c. (2 pts) Convert 0101 1001 from binary to base 10 **89**

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

e. (2 pts) Convert 1001 0111 from base 2 to decimal **151**

f. (2 pts) Convert 150 from decimal to binary **1001 0110**

g. (2 pts) Convert 97 from decimal to binary **0110 0001**

h. (2 pts) Convert 1110 0000 from base 2 to base 10 **224**

i. (2 pts) Convert 1011 1011 1100 1111 from base 2 to hexadecimal **bpcf**

3

Exam #139 Page: 3 Name: _____

139

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -36, what is this number's binary representation in 8-bit two's complement?

11011100

- c. (3 pts)

Given that 10100011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-93

- d. (3 pts)

Given that 10101001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-87

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

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

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

```
./runIt fig apple kiwi lime
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `h->next` **Node ***
- b. (3 pts) `argv[0]` **char ***
- c. (3 pts) `*p` **char**
- d. (3 pts) `&e` **char ***
- e. (3 pts) `argv[1][2]` **char**
- f. (3 pts) `argc` **int**
- g. (3 pts) `h->data` **int**
- h. (3 pts) `&h` **Node ****
- i. (3 pts) `g` **double ***
- j. (3 pts) `h->next->next` **Node ***
- k. (3 pts) `d` **Node**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #140 Page: 1 Name: _____

140

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #140 Page: 2 Name: _____

140

-
1. a. (2 pts) Convert 100 111 111 from binary to octal **477**
- b. (2 pts) Convert 1111 0100 1010 0010 from binary to base 16 **f4a2**
- c. (2 pts) Convert 254 from base 10 to binary **1111 1110**
- d. (2 pts) Convert 54 from base 10 to binary **0011 0110**
- e. (2 pts) Convert 7c3 from base 16 to base 2 **0111 1100 0011**
- f. (2 pts) Convert 100 110 000 from binary to octal **460**
- g. (2 pts) Convert 54 from octal to binary **101 100**
- h. (2 pts) Convert 1000 0110 0111 0010 from base 2 to base 16 **8672**
- i. (2 pts) Convert 1010 1110 0100 0011 from binary to base 16 **ae43**

3

Exam #140 Page: 3 Name: _____

140

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -80, what is this number's binary representation in 8-bit two's complement?

10110000

- c. (3 pts)

Given that 10110110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-74

- d. (3 pts)

Given that 10101001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-87

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit guava fig apple
```

a. (3 pts) What is the value of `argc` in this case? 4

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

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

d. (3 pts) What is the value of `argv[0][1]`? /

4. Given the following declarations:

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

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

- a. (3 pts) `b` `char *`
- b. (3 pts) `z->data` `int`
- c. (3 pts) `t` `Node`
- d. (3 pts) `z->next` `Node *`
- e. (3 pts) `*a` `double`
- f. (3 pts) `&z` `Node **`
- g. (3 pts) `z->next->next` `Node *`
- h. (3 pts) `&x` `char *`
- i. (3 pts) `argc` `int`
- j. (3 pts) `argv[0]` `char *`
- k. (3 pts) `argv[1][2]` `char`

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #141 Page: 1 Name: _____

141

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #141 Page: 2 Name: _____

141

1. a. (2 pts) Convert e48a from hexadecimal to base 2 **1110 0100 1000 1010**

- b. (2 pts) Convert 1010 1110 from binary to decimal **174**

- c. (2 pts) Convert 236 from decimal to binary **1110 1100**

- d. (2 pts) Convert 1100 0010 from base 2 to base 10 **194**

- e. (2 pts) Convert 1110 1011 from base 2 to base 10 **235**

- f. (2 pts) Convert 14 from octal to base 2 **001 100**

- g. (2 pts) Convert 27 from base 8 to binary **010 111**

- h. (2 pts) Convert 1110 from base 2 to base 10 **14**

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -115, what is this number's binary representation in 8-bit two's complement?

10001101

c. (3 pts)

Given that 10100000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-96

d. (3 pts)

Given that 11000010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-62

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit guava cherry
```

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

b. (3 pts) What is the value of `argv[0][4]`? **n**

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `x` `int *`
- b. (3 pts) `&r` `int *`
- c. (3 pts) `argc` `int`
- d. (3 pts) `p` `Node`
- e. (3 pts) `t->next` `Node *`
- f. (3 pts) `&y` `char **`
- g. (3 pts) `argv[1][2]` `char`
- h. (3 pts) `t->data` `int`
- i. (3 pts) `*t` `Node`
- j. (3 pts) `argv[0]` `char *`
- k. (3 pts) `t->next->next` `Node *`

6

Exam #141 Page: 6 Name: _____

141

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #142 Page: 1 Name: _____

142

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #142 Page: 2 Name: _____

142

1. a. (2 pts) Convert 0001 0101 from base 2 to base 10 **21**

- b. (2 pts) Convert 100 000 000 from base 2 to base 8 **400**

- c. (2 pts) Convert 0010 0010 1010 0101 from binary to base 16 **22a5**

- d. (2 pts) Convert 0001 1110 1111 1100 from base 2 to hexadecimal **1efc**

- e. (2 pts) Convert 010 000 010 from base 2 to base 8 **202**

- f. (2 pts) Convert 011 001 000 from base 2 to octal **310**

- g. (2 pts) Convert 95 from base 10 to base 2 **0101 1111**

- h. (2 pts) Convert 0111 0111 from binary to base 10 **119**

- i. (2 pts) Convert 6 from octal to base 2 **110**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -11, what is this number's binary representation in 8-bit two's complement?

11110101

c. (3 pts)

Given that 11100010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-30

d. (3 pts)

Given that 11110110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-10

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

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

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

```
./runIt banana cherry
```

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

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

c. (3 pts) What is the value of `argv[0][5]`? I

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

4. Given the following declarations:

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

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

- a. (3 pts) *z **char**
- b. (3 pts) x->next **Node ***
- c. (3 pts) argv[1][2] **char**
- d. (3 pts) x->next->next **Node ***
- e. (3 pts) argc **int**
- f. (3 pts) y **int ***
- g. (3 pts) &q **double ***
- h. (3 pts) r **Node**
- i. (3 pts) argv[0] **char ***
- j. (3 pts) &w **double ****
- k. (3 pts) x->data **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #143 Page: 1 Name: _____

143

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #143 Page: 2 Name: _____

143

-
1. a. (2 pts) Convert 0101 1010 1100 1011 from base 2 to hexadecimal **5acb**
- b. (2 pts) Convert 0011 1010 from base 2 to decimal **58**
- c. (2 pts) Convert 0001 0000 1000 0110 from binary to hexadecimal **1086**
- d. (2 pts) Convert 170 from base 10 to base 2 **1010 1010**
- e. (2 pts) Convert 36 from base 10 to binary **0010 0100**
- f. (2 pts) Convert 77 from octal to binary **111 111**
- g. (2 pts) Convert 9 from base 10 to base 2 **1001**
- h. (2 pts) Convert 1111 1110 from binary to base 10 **254**
- i. (2 pts) Convert 111 001 000 from base 2 to octal **710**

3

Exam #143 Page: 3 Name: _____

143

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -46, what is this number's binary representation in 8-bit two's complement?

11010010

- c. (3 pts)

Given that 11001100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-52

- d. (3 pts)

Given that 10001111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-113

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit mango guava grape apple
```

a. (3 pts) What is the value of `argc` in this case? **5**

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) e->data **int**
- b. (3 pts) e->next **Node ***
- c. (3 pts) argv[0] **char ***
- d. (3 pts) &e **Node ****
- e. (3 pts) c **double ***
- f. (3 pts) e->next->next **Node ***
- g. (3 pts) b **char**
- h. (3 pts) argc **int**
- i. (3 pts) argv[1][2] **char**
- j. (3 pts) *d **int**
- k. (3 pts) &y **double ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #144 Page: 2 Name: _____

144

1. a. (2 pts) Convert 0001 0111 0011 1110 from binary to base 16 **173e**

b. (2 pts) Convert 1001 1000 from binary to base 10 **152**

c. (2 pts) Convert 249 from base 10 to binary **1111 1001**

d. (2 pts) Convert 72 from base 8 to binary **111 010**

e. (2 pts) Convert 43 from decimal to base 2 **0010 1011**

f. (2 pts) Convert 1100 0011 1101 1011 from binary to base 16 **c3db**

g. (2 pts) Convert 1d2d from base 16 to binary **0001 1101 0010 1101**

h. (2 pts) Convert 1010 1111 0010 1000 from binary to hexadecimal **af28**

i. (2 pts) Convert 147 from base 10 to binary **1001 0011**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -12, what is this number's binary representation in 8-bit two's complement?

11110100

c. (3 pts)

Given that 10010110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-106

d. (3 pts)

Given that 11100000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-32

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

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

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

```
./runIt kiwi lime grape date
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

d. (3 pts) What is the value of `argv[0][1]`? /

4. Given the following declarations:

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

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

- a. (3 pts) `*b` **double**
- b. (3 pts) `argc` **int**
- c. (3 pts) `&z` **char ***
- d. (3 pts) `argv[1][2]` **char**
- e. (3 pts) `argv[0]` **char ***
- f. (3 pts) `c->next->next` **Node ***
- g. (3 pts) `c->next` **Node ***
- h. (3 pts) `y` **Node**
- i. (3 pts) `&a` **int ****
- j. (3 pts) `c->data` **int**
- k. (3 pts) `c` **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #145 Page: 1 Name: _____

145

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #145 Page: 2 Name: _____

145

-
1. a. (2 pts) Convert 010 111 000 from binary to base 8 **270**
- b. (2 pts) Convert 0101 0010 0101 0101 from binary to hexadecimal **5255**
- c. (2 pts) Convert 1110 0111 from base 2 to decimal **231**
- d. (2 pts) Convert 751d from base 16 to base 2 **0111 0101 0001 1101**
- e. (2 pts) Convert 1110 1101 0000 from binary to hexadecimal **ed0**
- f. (2 pts) Convert 5db8 from hexadecimal to base 2 **0101 1101 1011 1000**
- g. (2 pts) Convert 61 from octal to binary **110 001**
- h. (2 pts) Convert 0011 0110 from binary to decimal **54**
- i. (2 pts) Convert 010 111 011 from binary to base 8 **273**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -47, what is this number's binary representation in 8-bit two's complement?

11010001

c. (3 pts)

Given that 10000000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-128

d. (3 pts)

Given that 11111001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-7

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple kiwi fig
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][4]`? e

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `x->next->next` **Node ***
- b. (3 pts) `argv[1][2]` **char**
- c. (3 pts) `&s` **double ***
- d. (3 pts) `x->data` **int**
- e. (3 pts) `&y` **double ****
- f. (3 pts) `x->next` **Node ***
- g. (3 pts) `argv[0]` **char ***
- h. (3 pts) `argc` **int**
- i. (3 pts) `x` **Node ***
- j. (3 pts) `q` **int**
- k. (3 pts) `*y` **double**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #146 Page: 1 Name: _____

146

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #146 Page: 2 Name: _____

146

1. a. (2 pts) Convert 43 from base 8 to binary **100 011**

- b. (2 pts) Convert 23db from base 16 to base 2 **0010 0011 1101 1011**

- c. (2 pts) Convert 0001 1110 0001 1011 from base 2 to hexadecimal **1e1b**

- d. (2 pts) Convert 1101 0001 0101 1111 from binary to hexadecimal **d15f**

- e. (2 pts) Convert 64ee from hexadecimal to binary **0110 0100 1110 1110**

- f. (2 pts) Convert 100 100 000 from base 2 to octal **440**

- g. (2 pts) Convert c94b from base 16 to binary **1100 1001 0100 1011**

- h. (2 pts) Convert 1001 1111 from base 2 to decimal **159**

- i. (2 pts) Convert 255 from base 10 to binary **1111 1111**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -71, what is this number's binary representation in 8-bit two's complement?

10111001

c. (3 pts)

Given that 11000001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-63

d. (3 pts)

Given that 10101101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-83

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

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

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

```
./runIt date lime
```

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

b. (3 pts) What is the value of `argv[0][5]`? I

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

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

4. Given the following declarations:

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

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

- a. (3 pts) h->data int
- b. (3 pts) &d Node *
- c. (3 pts) g double *
- d. (3 pts) h->next->next Node *
- e. (3 pts) d Node
- f. (3 pts) argc int
- g. (3 pts) *p int
- h. (3 pts) argv[1][2] char
- i. (3 pts) argv[0] char *
- j. (3 pts) h->next Node *
- k. (3 pts) &g double **

6

Exam #146 Page: 6 Name: _____

146

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #147 Page: 1 Name: _____

147

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #147 Page: 2 Name: _____

147

1. a. (2 pts) Convert 1101 0010 from binary to base 10 **210**

- b. (2 pts) Convert 1101 1110 0000 0100 from base 2 to hexadecimal **de04**

- c. (2 pts) Convert 11 from decimal to binary **1011**

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

- e. (2 pts) Convert 0100 1000 from binary to base 10 **72**

- f. (2 pts) Convert 12 from base 8 to binary **001 010**

- g. (2 pts) Convert 733c from hexadecimal to binary **0111 0011 0011 1100**

- h. (2 pts) Convert 0010 0111 0111 1000 from base 2 to hexadecimal **2778**

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

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -106, what is this number's binary representation in 8-bit two's complement?

10010110

c. (3 pts)

Given that 10101011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-85

d. (3 pts)

Given that 11000110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-58

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

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

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

```
./runIt date lime grape fig
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `&f` **Node ****
- b. (3 pts) `&z` **int ***
- c. (3 pts) `g` **char ***
- d. (3 pts) `f->data` **int**
- e. (3 pts) `argc` **int**
- f. (3 pts) `c` **char**
- g. (3 pts) `f->next->next` **Node ***
- h. (3 pts) `f->next` **Node ***
- i. (3 pts) `*g` **char**
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `argv[0]` **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #148 Page: 1 Name: _____

148

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #148 Page: 2 Name: _____

148

1. a. (2 pts) Convert 0010 1010 from binary to base 10 **42**

b. (2 pts) Convert 80d0 from base 16 to base 2 **1000 0000 1101 0000**

c. (2 pts) Convert b15d from base 16 to base 2 **1011 0001 0101 1101**

d. (2 pts) Convert 24 from base 10 to base 2 **0001 1000**

e. (2 pts) Convert b8b2 from hexadecimal to binary **1011 1000 1011 0010**

f. (2 pts) Convert 43 from decimal to binary **0010 1011**

g. (2 pts) Convert 61 from base 8 to base 2 **110 001**

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

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

3

Exam #148 Page: 3 Name: _____

148

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

- b. (3 pts)

Given the decimal number -22, what is this number's binary representation in 8-bit two's complement?

11101010

- c. (3 pts)

Given that 10111111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-65

- d. (3 pts)

Given that 11000110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-58

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lemon apple kiwi cherry
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][4]`? n

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

d. (3 pts) What is the value of `argv[0][4]`? n

4. Given the following declarations:

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

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

- a. (3 pts) `t` **char**
- b. (3 pts) `argc` **int**
- c. (3 pts) `y->next->next` **Node ***
- d. (3 pts) `*z` **char**
- e. (3 pts) `y->next` **Node ***
- f. (3 pts) `argv[1][2]` **char**
- g. (3 pts) `y` **Node ***
- h. (3 pts) `y->data` **int**
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `&x` **double ****
- k. (3 pts) `&q` **int ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #149 Page: 1 Name: _____

149

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #149 Page: 2 Name: _____

149

1. a. (2 pts) Convert 0110 1111 1001 1011 from binary to base 16 **6f9b**

- b. (2 pts) Convert 001 110 101 from binary to octal **165**

- c. (2 pts) Convert 9f3d from hexadecimal to base 2 **1001 1111 0011 1101**

- d. (2 pts) Convert 1010 0100 from base 2 to decimal **164**

- e. (2 pts) Convert 46 from octal to binary **100 110**

- f. (2 pts) Convert 197 from decimal to base 2 **1100 0101**

- g. (2 pts) Convert 011 011 101 from binary to base 8 **335**

- h. (2 pts) Convert 0101 0101 from binary to decimal **85**

- i. (2 pts) Convert 100 001 100 from binary to base 8 **414**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -56, what is this number's binary representation in 8-bit two's complement?

11001000

c. (3 pts)

Given that 10101001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-87

d. (3 pts)

Given that 11011111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-33

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi mango fig
```

a. (3 pts) What is the value of `argc` in this case? 4

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

c. (3 pts) What is the value of `argv[2][4]`? o

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

4. Given the following declarations:

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

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

- a. (3 pts) h **Node**
- b. (3 pts) s->data **int**
- c. (3 pts) &q **char ***
- d. (3 pts) r **int ***
- e. (3 pts) s->next->next **Node ***
- f. (3 pts) *r **int**
- g. (3 pts) s->next **Node ***
- h. (3 pts) argv[1][2] **char**
- i. (3 pts) argc **int**
- j. (3 pts) &w **char ****
- k. (3 pts) argv[0] **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #150 Page: 1 Name: _____

150

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #150 Page: 2 Name: _____

150

1. a. (2 pts) Convert a0fd from base 16 to binary **1010 0000 1111 1101**

b. (2 pts) Convert 3 from base 8 to base 2 **011**

c. (2 pts) Convert 1101 0101 from binary to base 10 **213**

d. (2 pts) Convert 0001 0000 1000 from base 2 to hexadecimal **108**

e. (2 pts) Convert 111 100 100 from binary to octal **744**

f. (2 pts) Convert 247 from base 10 to binary **1111 0111**

g. (2 pts) Convert 34 from base 8 to base 2 **011 100**

h. (2 pts) Convert 190 from decimal to binary **1011 1110**

i. (2 pts) Convert 0010 0111 from base 2 to decimal **39**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -81, what is this number's binary representation in 8-bit two's complement?

10101111

c. (3 pts)

Given that 11101010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-22

d. (3 pts)

Given that 10010011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-109

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit mango grape
```

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

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

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

d. (3 pts) What is the value of `argv[1][4]`? o

4. Given the following declarations:

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

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

- a. (3 pts) `&c` **double ****
- b. (3 pts) `argc` **int**
- c. (3 pts) `*c` **double**
- d. (3 pts) `d->next` **Node ***
- e. (3 pts) `argv[0]` **char ***
- f. (3 pts) `&z` **Node ***
- g. (3 pts) `d->next->next` **Node ***
- h. (3 pts) `c` **double ***
- i. (3 pts) `d->data` **int**
- j. (3 pts) `a` **int**
- k. (3 pts) `argv[1][2]` **char**

6

Exam #150 Page: 6 Name: _____

150

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #151 Page: 1 Name: _____

151

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #151 Page: 2 Name: _____

151

1. a. (2 pts) Convert 71 from base 8 to binary **111 001**

- b. (2 pts) Convert 198 from base 10 to binary **1100 0110**

- c. (2 pts) Convert 60 from base 8 to base 2 **110 000**

- d. (2 pts) Convert 140 from base 10 to base 2 **1000 1100**

- e. (2 pts) Convert 1101 0101 from binary to base 10 **213**

- f. (2 pts) Convert 9171 from base 16 to binary **1001 0001 0111 0001**

- g. (2 pts) Convert 6 from octal to base 2 **110**

- h. (2 pts) Convert 0100 0110 0011 1001 from base 2 to hexadecimal **4639**

- i. (2 pts) Convert 111 100 100 from binary to base 8 **744**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -115, what is this number's binary representation in 8-bit two's complement?

10001101

c. (3 pts)

Given that 11010101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-43

d. (3 pts)

Given that 10101100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-84

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

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

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

```
./runIt mango apple grape fig
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) t **double**
- b. (3 pts) argv[1][2] **char**
- c. (3 pts) b->data **int**
- d. (3 pts) *a **int**
- e. (3 pts) c **char ***
- f. (3 pts) &b **Node ****
- g. (3 pts) argc **int**
- h. (3 pts) argv[0] **char ***
- i. (3 pts) b->next **Node ***
- j. (3 pts) &t **double ***
- k. (3 pts) b->next->next **Node ***

6

Exam #151 Page: 6 Name: _____

151

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #152 Page: 1 Name: _____

152

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #152 Page: 2 Name: _____

152

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

- b. (2 pts) Convert 011 010 010 from base 2 to base 8 **322**

- c. (2 pts) Convert 011 010 001 from binary to octal **321**

- d. (2 pts) Convert 010 010 001 from base 2 to base 8 **221**

- e. (2 pts) Convert 0100 0101 1101 1111 from binary to hexadecimal **45df**

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

- g. (2 pts) Convert 108 from base 10 to binary **0110 1100**

- h. (2 pts) Convert 1110 1100 from base 2 to decimal **236**

- i. (2 pts) Convert 111 001 000 from base 2 to octal **710**

3

Exam #152 Page: 3 Name: _____

152

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

- b. (3 pts)

Given the decimal number -32, what is this number's binary representation in 8-bit two's complement?

11100000

- c. (3 pts)

Given that 11101000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-24

- d. (3 pts)

Given that 10101011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-85

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lime cherry kiwi fig
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][0]`? l

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

d. (3 pts) What is the value of `argv[2][4]`? r

4. Given the following declarations:

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

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

- a. (3 pts) `s->next` **Node ***
- b. (3 pts) `r` **int ***
- c. (3 pts) `argv[1][2]` **char**
- d. (3 pts) `s->next->next` **Node ***
- e. (3 pts) `&f` **double ***
- f. (3 pts) `s->data` **int**
- g. (3 pts) `f` **double**
- h. (3 pts) `argc` **int**
- i. (3 pts) `&r` **int ****
- j. (3 pts) `argv[0]` **char ***
- k. (3 pts) `*q` **double**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #153 Page: 1 Name: _____

153

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #153 Page: 2 Name: _____

153

-
1. a. (2 pts) Convert 131 from base 10 to base 2 **1000 0011**
- b. (2 pts) Convert 0010 0011 from binary to decimal **35**
- c. (2 pts) Convert 0101 0110 1011 1111 from base 2 to hexadecimal **56bf**
- d. (2 pts) Convert 110 101 000 from base 2 to octal **650**
- e. (2 pts) Convert 12 from octal to base 2 **001 010**
- f. (2 pts) Convert 0010 1100 1001 1111 from binary to hexadecimal **2c9f**
- g. (2 pts) Convert 22 from decimal to base 2 **0001 0110**
- h. (2 pts) Convert 0111 0100 from base 2 to decimal **116**
- i. (2 pts) Convert 1010 1110 from binary to base 10 **174**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -66, what is this number's binary representation in 8-bit two's complement?

10111110

c. (3 pts)

Given that 11010010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-46

d. (3 pts)

Given that 11000101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-59

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple grape cherry
```

a. (3 pts) What is the value of `argc` in this case? 4

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

c. (3 pts) What is the value of `argv[1][0]`? a

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

4. Given the following declarations:

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

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

- a. (3 pts) `&q` `char **`
- b. (3 pts) `g->next->next` `Node *`
- c. (3 pts) `argv[0]` `char *`
- d. (3 pts) `g->data` `int`
- e. (3 pts) `argc` `int`
- f. (3 pts) `e` `double`
- g. (3 pts) `*p` `double`
- h. (3 pts) `&f` `char *`
- i. (3 pts) `g` `Node *`
- j. (3 pts) `argv[1][2]` `char`
- k. (3 pts) `g->next` `Node *`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #154 Page: 1 Name: _____

154

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #154 Page: 2 Name: _____

154

-
1. a. (2 pts) Convert 101 101 000 from binary to base 8 **550**
- b. (2 pts) Convert 111 101 010 from binary to base 8 **752**
- c. (2 pts) Convert 141 from base 10 to base 2 **1000 1101**
- d. (2 pts) Convert 0011 0000 from base 2 to base 10 **48**
- e. (2 pts) Convert 011 111 110 from binary to base 8 **376**
- f. (2 pts) Convert 0101 1111 from binary to base 10 **95**
- g. (2 pts) Convert 110 001 from base 2 to octal **61**
- h. (2 pts) Convert 221 from decimal to binary **1101 1101**
- i. (2 pts) Convert 010 100 000 from base 2 to octal **240**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -91, what is this number's binary representation in 8-bit two's complement?

10100101

c. (3 pts)

Given that 10010100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-108

d. (3 pts)

Given that 11111000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-8

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

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

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

```
./runIt grape lime
```

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

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `argv[0]` `char *`
- b. (3 pts) `g->next` `Node *`
- c. (3 pts) `&e` `int *`
- d. (3 pts) `q` `char *`
- e. (3 pts) `g->next->next` `Node *`
- f. (3 pts) `argv[1][2]` `char`
- g. (3 pts) `f` `char`
- h. (3 pts) `*p` `int`
- i. (3 pts) `&h` `double **`
- j. (3 pts) `argc` `int`
- k. (3 pts) `g->data` `int`

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #155 Page: 1 Name: _____

155

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. a. (2 pts) Convert f959 from base 16 to binary 1111 1001 0101 1001
- b. (2 pts) Convert af4a from base 16 to binary 1010 1111 0100 1010
- c. (2 pts) Convert 011 110 101 from base 2 to octal 365
- d. (2 pts) Convert 1011 1100 from base 2 to decimal 188
- e. (2 pts) Convert 98 from decimal to binary 0110 0010
- f. (2 pts) Convert 111 110 001 from base 2 to octal 761
- g. (2 pts) Convert 110 000 101 from binary to octal 605
- h. (2 pts) Convert 0110 0100 1111 1011 from binary to hexadecimal 64fb
- i. (2 pts) Convert 0001 1010 from binary to base 10 26

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -125, what is this number's binary representation in 8-bit two's complement?

10000011

c. (3 pts)

Given that 11111110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-2

d. (3 pts)

Given that 10010010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-110

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi date apple guava
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[0][5]`? I

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `&y` **char ****
- b. (3 pts) `argv[1][2]` **char**
- c. (3 pts) `r` **Node**
- d. (3 pts) `x->next` **Node ***
- e. (3 pts) `&q` **int ***
- f. (3 pts) `x->data` **int**
- g. (3 pts) `argc` **int**
- h. (3 pts) `x->next->next` **Node ***
- i. (3 pts) `*y` **char**
- j. (3 pts) `argv[0]` **char ***
- k. (3 pts) `x` **Node ***

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #156 Page: 1 Name: _____

156

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #156 Page: 2 Name: _____

156

-
1. a. (2 pts) Convert 51b6 from hexadecimal to binary 0101 0001 1011 0110
- b. (2 pts) Convert 82 from base 10 to binary 0101 0010
- c. (2 pts) Convert 0010 0000 from binary to base 10 32
- d. (2 pts) Convert 36 from octal to binary 011 110
- e. (2 pts) Convert 1101 0011 0000 1101 from base 2 to base 16 d30d
- f. (2 pts) Convert 1111 1010 0011 0101 from base 2 to hexadecimal fa35
- g. (2 pts) Convert 20 from base 10 to base 2 0001 0100
- h. (2 pts) Convert 1011 from binary to decimal 11
- i. (2 pts) Convert 3 from base 8 to base 2 011

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -41, what is this number's binary representation in 8-bit two's complement?

11010111

c. (3 pts)

Given that 10010001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-111

d. (3 pts)

Given that 11010010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-46

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lime fig lemon mango
```

a. (3 pts) What is the value of `argc` in this case? **5**

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

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

d. (3 pts) What is the value of `argv[0][4]`? **n**

4. Given the following declarations:

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

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

- a. (3 pts) `g` **double ***
- b. (3 pts) `&c` **double ***
- c. (3 pts) `h->next` **Node ***
- d. (3 pts) `argc` **int**
- e. (3 pts) `&h` **Node ****
- f. (3 pts) `*h` **Node**
- g. (3 pts) `argv[1][2]` **char**
- h. (3 pts) `h->data` **int**
- i. (3 pts) `h->next->next` **Node ***
- j. (3 pts) `c` **double**
- k. (3 pts) `argv[0]` **char ***

6

Exam #156 Page: 6 Name: _____

156

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. a. (2 pts) Convert 100 101 101 from base 2 to octal 455
- b. (2 pts) Convert c3f from base 16 to base 2 1100 0011 1111
- c. (2 pts) Convert 1110 from base 2 to base 10 14
- d. (2 pts) Convert 418 from hexadecimal to base 2 0100 0001 1000
- e. (2 pts) Convert 101 101 100 from base 2 to base 8 554
- f. (2 pts) Convert 9412 from base 16 to binary 1001 0100 0001 0010
- g. (2 pts) Convert be65 from hexadecimal to binary 1011 1110 0110 0101
- h. (2 pts) Convert 147 from base 10 to binary 1001 0011
- i. (2 pts) Convert 110 101 101 from binary to octal 655

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -76, what is this number's binary representation in 8-bit two's complement?

10110100

c. (3 pts)

Given that 11111100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-4

d. (3 pts)

Given that 10101011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-85

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

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

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

```
./runIt guava date
```

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

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `c` `double *`
- b. (3 pts) `argv[0]` `char *`
- c. (3 pts) `b` `char`
- d. (3 pts) `argv[1][2]` `char`
- e. (3 pts) `&c` `double **`
- f. (3 pts) `d->next` `Node *`
- g. (3 pts) `d->next->next` `Node *`
- h. (3 pts) `*c` `double`
- i. (3 pts) `argc` `int`
- j. (3 pts) `&y` `double *`
- k. (3 pts) `d->data` `int`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #158 Page: 1 Name: _____

158

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #158 Page: 2 Name: _____

158

1. a. (2 pts) Convert 1100 0111 1111 0111 from binary to base 16 **c7f7**

b. (2 pts) Convert 1101 1101 1100 0100 from binary to hexadecimal **ddc4**

c. (2 pts) Convert 68 from base 10 to binary **0100 0100**

d. (2 pts) Convert 0110 0000 0101 1001 from base 2 to base 16 **6059**

e. (2 pts) Convert 1100 from base 2 to base 10 **12**

f. (2 pts) Convert c67c from hexadecimal to base 2 **1100 0110 0111 1100**

g. (2 pts) Convert 192 from base 10 to base 2 **1100 0000**

h. (2 pts) Convert 1111 1100 0000 1100 from binary to hexadecimal **fc0c**

i. (2 pts) Convert 011 110 001 from base 2 to octal **361**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -100, what is this number's binary representation in 8-bit two's complement?

10011100

c. (3 pts)

Given that 10111101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-67

d. (3 pts)

Given that 11011110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-34

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

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

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

```
./runIt apple grape
```

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

b. (3 pts) What is the value of `argv[2][4]`? e

c. (3 pts) What is the value of `argv[0][1]`? /

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

4. Given the following declarations:

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

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

- a. (3 pts) `argc` **int**
- b. (3 pts) `&b` **int ***
- c. (3 pts) `*g` **char**
- d. (3 pts) `&f` **int ****
- e. (3 pts) `argv[1][2]` **char**
- f. (3 pts) `g` **char ***
- g. (3 pts) `d->next` **Node ***
- h. (3 pts) `argv[0]` **char ***
- i. (3 pts) `d->next->next` **Node ***
- j. (3 pts) `d->data` **int**
- k. (3 pts) `z` **Node**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. a. (2 pts) Convert 1100 from binary to base 10 12

b. (2 pts) Convert 45 from octal to base 2 100 101

c. (2 pts) Convert 3280 from hexadecimal to binary 0011 0010 1000 0000

d. (2 pts) Convert 73 from octal to base 2 111 011

e. (2 pts) Convert efc1 from base 16 to binary 1110 1111 1100 0001

f. (2 pts) Convert 0110 0000 0101 1000 from base 2 to base 16 6058

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

h. (2 pts) Convert 40 from octal to binary 100 000

i. (2 pts) Convert 010 000 101 from binary to base 8 205

-
2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -96, what is this number's binary representation in 8-bit two's complement?

10100000

b. (3 pts)

Given the decimal number -7, what is this number's binary representation in 8-bit two's complement?

1111001

c. (3 pts)

Given that 10100111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-89

d. (3 pts)

Given that 11111000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-8

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit lemon mango
```

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

b. (3 pts) What is the value of `argv[2][4]`? o

c. (3 pts) What is the value of `argv[1][0]`? l

d. (3 pts) What is the value of `argv[0][6]`? t

4. Given the following declarations:

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

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

- a. (3 pts) `b` **int ***
- b. (3 pts) `&x` **int ***
- c. (3 pts) `argc` **int**
- d. (3 pts) `z->next` **Node ***
- e. (3 pts) `y` **char**
- f. (3 pts) `argv[0]` **char ***
- g. (3 pts) `*a` **double**
- h. (3 pts) `z->next->next` **Node ***
- i. (3 pts) `argv[1][2]` **char**
- j. (3 pts) `z->data` **int**
- k. (3 pts) `&z` **Node ****

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #160 Page: 1 Name: _____

160

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #160 Page: 2 Name: _____

160

1. a. (2 pts) Convert 0001 from binary to decimal **1**

- b. (2 pts) Convert 0111 1111 from base 2 to base 10 **127**

- c. (2 pts) Convert 100 100 111 from binary to octal **447**

- d. (2 pts) Convert 251d from base 16 to base 2 **0010 0101 0001 1101**

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

- f. (2 pts) Convert 47 from base 8 to base 2 **100 111**

- g. (2 pts) Convert 76 from octal to base 2 **111 110**

- h. (2 pts) Convert 0010 0000 from binary to decimal **32**

- i. (2 pts) Convert 7811 from hexadecimal to binary **0111 1000 0001 0001**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -1, what is this number's binary representation in 8-bit two's complement?

11111111

c. (3 pts)

Given that 10000100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-124

d. (3 pts)

Given that 10100110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-90

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple lemon banana cherry
```

a. (3 pts) What is the value of `argc` in this case? 5

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

c. (3 pts) What is the value of `argv[0][4]`? n

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

4. Given the following declarations:

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

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

- a. (3 pts) `h->next->next` **Node ***
- b. (3 pts) `&c` **double ***
- c. (3 pts) `e` **char**
- d. (3 pts) `g` **double ***
- e. (3 pts) `&f` **int ****
- f. (3 pts) `argc` **int**
- g. (3 pts) `h->data` **int**
- h. (3 pts) `argv[0]` **char ***
- i. (3 pts) `h->next` **Node ***
- j. (3 pts) `*f` **int**
- k. (3 pts) `argv[1][2]` **char**

6

Exam #160 Page: 6 Name: _____

160

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #161 Page: 1 Name: _____

161

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #161 Page: 2 Name: _____

161

1. a. (2 pts) Convert 0100 0101 1111 1100 from base 2 to hexadecimal **45fc**

- b. (2 pts) Convert 0011 1001 1111 1011 from base 2 to hexadecimal **39fb**

- c. (2 pts) Convert 1000 0001 1100 1110 from base 2 to hexadecimal **81ce**

- d. (2 pts) Convert b107 from base 16 to binary **1011 0001 0000 0111**

- e. (2 pts) Convert 1010 1100 from base 2 to decimal **172**

- f. (2 pts) Convert 0011 0111 0010 0011 from base 2 to base 16 **3723**

- g. (2 pts) Convert 1010 0100 from base 2 to base 10 **164**

- h. (2 pts) Convert 1010 0111 1110 1010 from base 2 to hexadecimal **a7ea**

- i. (2 pts) Convert 4228 from base 16 to base 2 **0100 0010 0010 1000**

3

Exam #161 Page: 3 Name: _____

161

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -35, what is this number's binary representation in 8-bit two's complement?

11011101

- c. (3 pts)

Given that 11101110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-18

- d. (3 pts)

Given that 10111111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-65

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

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

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

```
./runIt lemon kiwi grape
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][1]`? /

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

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

4. Given the following declarations:

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

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

- a. (3 pts) b->next **Node ***
- b. (3 pts) &x **Node ***
- c. (3 pts) argc **int**
- d. (3 pts) *c **int**
- e. (3 pts) b->data **int**
- f. (3 pts) y **int**
- g. (3 pts) b **Node ***
- h. (3 pts) &d **double ****
- i. (3 pts) argv[1][2] **char**
- j. (3 pts) argv[0] **char ***
- k. (3 pts) b->next->next **Node ***

6

Exam #161 Page: 6 Name: _____

161

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #162 Page: 1 Name: _____

162

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #162 Page: 2 Name: _____

162

-
1. a. (2 pts) Convert 119 from decimal to base 2 **0111 0111**
- b. (2 pts) Convert b81 from hexadecimal to base 2 **1011 1000 0001**
- c. (2 pts) Convert 184 from base 10 to base 2 **1011 1000**
- d. (2 pts) Convert 1101 0100 1000 from base 2 to base 16 **d48**
- e. (2 pts) Convert 3 from base 10 to binary **0011**
- f. (2 pts) Convert 011 010 011 from binary to octal **323**
- g. (2 pts) Convert 51 from octal to base 2 **101 001**
- h. (2 pts) Convert 0001 0000 1101 1001 from binary to hexadecimal **10d9**
- i. (2 pts) Convert e3e3 from hexadecimal to binary **1110 0011 1110 0011**

3

Exam #162 Page: 3 Name: _____

162

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -60, what is this number's binary representation in 8-bit two's complement?

11000100

- c. (3 pts)

Given that 10101111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-81

- d. (3 pts)

Given that 11110011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-13

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit mango guava
```

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

b. (3 pts) What is the value of `argv[2][4]`? **a**

c. (3 pts) What is the value of `argv[0][5]`? **I**

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

4. Given the following declarations:

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

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

- a. (3 pts) c->next->next **Node ***
- b. (3 pts) &b **char ***
- c. (3 pts) c->data **int**
- d. (3 pts) argv[1][2] **char**
- e. (3 pts) c->next **Node ***
- f. (3 pts) argv[0] **char ***
- g. (3 pts) *d **double**
- h. (3 pts) argc **int**
- i. (3 pts) a **int**
- j. (3 pts) &c **Node ****
- k. (3 pts) c **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #163 Page: 1 Name: _____

163

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #163 Page: 2 Name: _____

163

1. a. (2 pts) Convert 57 from octal to binary **101 111**

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

- c. (2 pts) Convert 101 001 100 from base 2 to octal **514**

- d. (2 pts) Convert 46 from base 8 to binary **100 110**

- e. (2 pts) Convert e65a from base 16 to binary **1110 0110 0101 1010**

- f. (2 pts) Convert 110 from binary to octal **6**

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

- h. (2 pts) Convert 100 110 001 from binary to octal **461**

- i. (2 pts) Convert 1010 1101 1111 1010 from base 2 to base 16 **adfa**

3

Exam #163 Page: 3 Name: _____

163

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -102, what is this number's binary representation in 8-bit two's complement?

10011010

- c. (3 pts)

Given that 10001100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-116

- d. (3 pts)

Given that 11010111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-41

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi date mango lemon
```

a. (3 pts) What is the value of `argc` in this case? 5

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `r->next->next` **Node ***
- b. (3 pts) `f` **double**
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `r->next` **Node ***
- e. (3 pts) `&r` **Node ****
- f. (3 pts) `r->data` **int**
- g. (3 pts) `argv[1][2]` **char**
- h. (3 pts) `&g` **Node ***
- i. (3 pts) `s` **char ***
- j. (3 pts) `argc` **int**
- k. (3 pts) `*r` **Node**

6

Exam #163 Page: 6 Name: _____

163

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #164 Page: 1 Name: _____

164

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #164 Page: 2 Name: _____

164

1. a. (2 pts) Convert 5 from octal to binary **101**

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

- c. (2 pts) Convert 0100 1011 from base 2 to base 10 **75**

- d. (2 pts) Convert 84 from base 10 to base 2 **0101 0100**

- e. (2 pts) Convert 010 101 101 from base 2 to octal **255**

- f. (2 pts) Convert 4b9 from base 16 to binary **0100 1011 1001**

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

- h. (2 pts) Convert 3efb from hexadecimal to binary **0011 1110 1111 1011**

- i. (2 pts) Convert 1010 0000 0110 1110 from binary to hexadecimal **a06e**

3

Exam #164 Page: 3 Name: _____

164

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -11, what is this number's binary representation in 8-bit two's complement?

11110101

- c. (3 pts)

Given that 10101101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-83

- d. (3 pts)

Given that 10001100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-116

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit cherry mango lime banana
```

a. (3 pts) What is the value of `argc` in this case? **5**

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

c. (3 pts) What is the value of `argv[2][4]`? **o**

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

4. Given the following declarations:

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

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

- a. (3 pts) `argv[0]` `char *`
- b. (3 pts) `x` `int`
- c. (3 pts) `argv[1][2]` `char`
- d. (3 pts) `argc` `int`
- e. (3 pts) `*d` `Node`
- f. (3 pts) `&z` `Node *`
- g. (3 pts) `d->next` `Node *`
- h. (3 pts) `c` `double *`
- i. (3 pts) `&c` `double **`
- j. (3 pts) `d->next->next` `Node *`
- k. (3 pts) `d->data` `int`

6

Exam #164 Page: 6 Name: _____

164

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #165 Page: 1 Name: _____

165

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #165 Page: 2 Name: _____

165

-
1. a. (2 pts) Convert 89 from decimal to binary **0101 1001**
- b. (2 pts) Convert 001 000 101 from base 2 to octal **105**
- c. (2 pts) Convert 16 from base 8 to base 2 **001 110**
- d. (2 pts) Convert 1110 0000 from base 2 to decimal **224**
- e. (2 pts) Convert 0011 1010 from base 2 to decimal **58**
- f. (2 pts) Convert 158 from base 10 to base 2 **1001 1110**
- g. (2 pts) Convert 75 from base 10 to base 2 **0100 1011**
- h. (2 pts) Convert 1100 0110 1010 1011 from binary to hexadecimal **c6ab**
- i. (2 pts) Convert 6a85 from base 16 to binary **0110 1010 1000 0101**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -45, what is this number's binary representation in 8-bit two's complement?

11010011

c. (3 pts)

Given that 10011000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-104

d. (3 pts)

Given that 10100101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-91

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit grape date cherry
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][6]`? t

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

d. (3 pts) What is the value of `argv[1][0]`? g

4. Given the following declarations:

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

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

- a. (3 pts) `argv[1][2]` **char**
- b. (3 pts) `z->data` **int**
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `*b` **char**
- e. (3 pts) `&t` **Node ***
- f. (3 pts) `&b` **char ****
- g. (3 pts) `z->next` **Node ***
- h. (3 pts) `argc` **int**
- i. (3 pts) `z->next->next` **Node ***
- j. (3 pts) `w` **double**
- k. (3 pts) `y` **int ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #166 Page: 1 Name: _____

166

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #166 Page: 2 Name: _____

166

-
1. a. (2 pts) Convert 100 010 101 from base 2 to base 8 **425**
- b. (2 pts) Convert 1111 0100 from base 2 to base 10 **244**
- c. (2 pts) Convert 111 from base 10 to binary **0110 1111**
- d. (2 pts) Convert 0011 1100 1111 0001 from base 2 to base 16 **3cf1**
- e. (2 pts) Convert 144 from base 10 to binary **1001 0000**
- f. (2 pts) Convert d100 from hexadecimal to binary **1101 0001 0000 0000**
- g. (2 pts) Convert 78 from decimal to binary **0100 1110**
- h. (2 pts) Convert 001 011 111 from base 2 to octal **137**
- i. (2 pts) Convert 12 from decimal to binary **1100**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -70, what is this number's binary representation in 8-bit two's complement?

10111010

c. (3 pts)

Given that 11011001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-39

d. (3 pts)

Given that 11111100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-4

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple cherry
```

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

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

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

d. (3 pts) What is the value of `argv[1][0]`? a

4. Given the following declarations:

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

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

- a. (3 pts) z->next->next **Node ***
- b. (3 pts) &s **double ***
- c. (3 pts) z->data **int**
- d. (3 pts) argv[0] **char ***
- e. (3 pts) z->next **Node ***
- f. (3 pts) b **char ***
- g. (3 pts) *b **char**
- h. (3 pts) s **double**
- i. (3 pts) argc **int**
- j. (3 pts) argv[1][2] **char**
- k. (3 pts) &y **double ****

6

Exam #166 Page: 6 Name: _____

166

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #167 Page: 1 Name: _____

167

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #167 Page: 2 Name: _____

167

1. a. (2 pts) Convert cfba from base 16 to base 2 **1100 1111 1011 1010**

- b. (2 pts) Convert 174 from decimal to binary **1010 1110**

- c. (2 pts) Convert 5d8f from hexadecimal to binary **0101 1101 1000 1111**

- d. (2 pts) Convert 200 from base 10 to binary **1100 1000**

- e. (2 pts) Convert 0111 0011 1000 1000 from base 2 to base 16 **7388**

- f. (2 pts) Convert 106 from decimal to binary **0110 1010**

- g. (2 pts) Convert 248 from base 10 to binary **1111 1000**

- h. (2 pts) Convert 1011 0111 from base 2 to decimal **183**

- i. (2 pts) Convert d657 from base 16 to base 2 **1101 0110 0101 0111**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -104, what is this number's binary representation in 8-bit two's complement?

10011000

c. (3 pts)

Given that 11000011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-61

d. (3 pts)

Given that 11110010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-14

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple date cherry lemon
```

a. (3 pts) What is the value of `argc` in this case? 5

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

c. (3 pts) What is the value of `argv[1][4]`? e

d. (3 pts) What is the value of `argv[0][5]`? I

4. Given the following declarations:

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

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

- a. (3 pts) `argc` **int**
- b. (3 pts) `&x` **Node ****
- c. (3 pts) `r` **Node**
- d. (3 pts) `&s` **char ***
- e. (3 pts) `x->data` **int**
- f. (3 pts) `x->next->next` **Node ***
- g. (3 pts) `x` **Node ***
- h. (3 pts) `x->next` **Node ***
- i. (3 pts) `*t` **double**
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `argv[0]` **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #168 Page: 1 Name: _____

168

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #168 Page: 2 Name: _____

168

1. a. (2 pts) Convert 2817 from base 16 to base 2 **0010 1000 0001 0111**

b. (2 pts) Convert 010 100 010 from binary to base 8 **242**

c. (2 pts) Convert 0010 1111 0001 from base 2 to base 16 **2f1**

d. (2 pts) Convert 100 001 000 from binary to octal **410**

e. (2 pts) Convert 111 001 000 from base 2 to octal **710**

f. (2 pts) Convert 011 011 000 from binary to octal **330**

g. (2 pts) Convert 0100 1001 1100 1111 from base 2 to base 16 **49cf**

h. (2 pts) Convert 5dbc from base 16 to binary **0101 1101 1011 1100**

i. (2 pts) Convert c8ca from hexadecimal to base 2 **1100 1000 1100 1010**

3

Exam #168 Page: 3 Name: _____

168

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -21, what is this number's binary representation in 8-bit two's complement?

11101011

- c. (3 pts)

Given that 11010110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-42

- d. (3 pts)

Given that 11110001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-15

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit grape mango date apple
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[0][1]`? /

c. (3 pts) What is the value of `argv[1][0]`? g

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

4. Given the following declarations:

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

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

- a. (3 pts) `&d` **Node ***
- b. (3 pts) `*f` **double**
- c. (3 pts) `argc` **int**
- d. (3 pts) `h->next` **Node ***
- e. (3 pts) `h->next->next` **Node ***
- f. (3 pts) `g` **int ***
- g. (3 pts) `argv[1][2]` **char**
- h. (3 pts) `h->data` **int**
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `&g` **int ****
- k. (3 pts) `c` **int**

6

Exam #168 Page: 6 Name: _____

168

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #169 Page: 1 Name: _____

169

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #169 Page: 2 Name: _____

169

1. a. (2 pts) Convert 33 from base 8 to binary **011 011**

- b. (2 pts) Convert b41 from base 16 to binary **1011 0100 0001**

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

- d. (2 pts) Convert 100 000 from binary to base 8 **40**

- e. (2 pts) Convert 199 from base 10 to base 2 **1100 0111**

- f. (2 pts) Convert 1 from base 8 to binary **001**

- g. (2 pts) Convert 1111 0011 1100 0000 from binary to hexadecimal **f3c0**

- h. (2 pts) Convert 111 001 010 from binary to octal **712**

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

3

Exam #169 Page: 3 Name: _____

169

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -55, what is this number's binary representation in 8-bit two's complement?

11001001

- c. (3 pts)

Given that 11000001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-63

- d. (3 pts)

Given that 10001011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-117

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi fig banana
```

a. (3 pts) What is the value of `argc` in this case? 4

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `&p` **int ***
- b. (3 pts) `*t` **int**
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `s->data` **int**
- e. (3 pts) `s->next` **Node ***
- f. (3 pts) `s` **Node ***
- g. (3 pts) `s->next->next` **Node ***
- h. (3 pts) `h` **Node**
- i. (3 pts) `argv[1][2]` **char**
- j. (3 pts) `&x` **char ****
- k. (3 pts) `argc` **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #170 Page: 1 Name: _____

170

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #170 Page: 2 Name: _____

170

-
1. a. (2 pts) Convert 1001 1110 0101 1000 from binary to hexadecimal **9e58**
- b. (2 pts) Convert 220 from base 10 to base 2 **1101 1100**
- c. (2 pts) Convert 11 from octal to base 2 **001 001**
- d. (2 pts) Convert 0110 1100 from base 2 to decimal **108**
- e. (2 pts) Convert 1d6a from base 16 to base 2 **0001 1101 0110 1010**
- f. (2 pts) Convert 56 from base 10 to base 2 **0011 1000**
- g. (2 pts) Convert 245 from decimal to base 2 **1111 0101**
- h. (2 pts) Convert 010 011 100 from base 2 to octal **234**
- i. (2 pts) Convert 349c from hexadecimal to base 2 **0011 0100 1001 1100**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -80, what is this number's binary representation in 8-bit two's complement?

10110000

c. (3 pts)

Given that 10000010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-126

d. (3 pts)

Given that 10111110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-66

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit fig cherry
```

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

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

c. (3 pts) What is the value of `argv[0][1]`? /

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

4. Given the following declarations:

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

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

- a. (3 pts) `*t` **double**
- b. (3 pts) `s->next` **Node ***
- c. (3 pts) `x` **char ***
- d. (3 pts) `s->next->next` **Node ***
- e. (3 pts) `argv[1][2]` **char**
- f. (3 pts) `argv[0]` **char ***
- g. (3 pts) `&t` **double ****
- h. (3 pts) `&h` **Node ***
- i. (3 pts) `s->data` **int**
- j. (3 pts) `p` **double**
- k. (3 pts) `argc` **int**

6

Exam #170 Page: 6 Name: _____

170

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #171 Page: 2 Name: _____

171

1. a. (2 pts) Convert 111 000 110 from base 2 to base 8 **706**

b. (2 pts) Convert 96f0 from hexadecimal to binary **1001 0110 1111 0000**

c. (2 pts) Convert 1510 from base 16 to base 2 **0001 0101 0001 0000**

d. (2 pts) Convert 76 from octal to base 2 **111 110**

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

f. (2 pts) Convert 1101 0010 from binary to decimal **210**

g. (2 pts) Convert 1001 1111 1101 1111 from base 2 to base 16 **9fdf**

h. (2 pts) Convert 1101 0110 from base 2 to decimal **214**

i. (2 pts) Convert 254 from base 10 to binary **1111 1110**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -114, what is this number's binary representation in 8-bit two's complement?

10001110

c. (3 pts)

Given that 11101100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-20

d. (3 pts)

Given that 11011000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-40

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit kiwi date lemon apple
```

a. (3 pts) What is the value of `argc` in this case? 5

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

c. (3 pts) What is the value of `argv[1][0]`? k

d. (3 pts) What is the value of `argv[0][1]`? /

4. Given the following declarations:

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

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

- a. (3 pts) `*r` **Node**
- b. (3 pts) `argc` **int**
- c. (3 pts) `r` **Node ***
- d. (3 pts) `&s` **char ****
- e. (3 pts) `r->data` **int**
- f. (3 pts) `argv[0]` **char ***
- g. (3 pts) `h` **char**
- h. (3 pts) `r->next->next` **Node ***
- i. (3 pts) `argv[1][2]` **char**
- j. (3 pts) `&h` **char ***
- k. (3 pts) `r->next` **Node ***

6

Exam #171 Page: 6 Name: _____

171

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #172 Page: 1 Name: _____

172

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #172 Page: 2 Name: _____

172

-
1. a. (2 pts) Convert 0011 1011 from binary to base 10 **59**
- b. (2 pts) Convert 57 from decimal to binary **0011 1001**
- c. (2 pts) Convert 1011 1010 0111 0010 from binary to hexadecimal **ba72**
- d. (2 pts) Convert b417 from hexadecimal to binary **1011 0100 0001 0111**
- e. (2 pts) Convert 712e from base 16 to base 2 **0111 0001 0010 1110**
- f. (2 pts) Convert 1101 0011 1010 0001 from binary to hexadecimal **d3a1**
- g. (2 pts) Convert 74 from base 8 to binary **111 100**
- h. (2 pts) Convert 0111 1100 0111 1110 from base 2 to hexadecimal **7c7e**
- i. (2 pts) Convert 241 from decimal to base 2 **1111 0001**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -30, what is this number's binary representation in 8-bit two's complement?

11100010

c. (3 pts)

Given that 10000000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-128

d. (3 pts)

Given that 11010111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-41

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple grape date lime
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[0][5]`? I

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `d->next->next` **Node ***
- b. (3 pts) `argc` **int**
- c. (3 pts) `argv[1][2]` **char**
- d. (3 pts) `z` **Node**
- e. (3 pts) `b` **int ***
- f. (3 pts) `&d` **Node ****
- g. (3 pts) `d->next` **Node ***
- h. (3 pts) `d->data` **int**
- i. (3 pts) `&a` **char ***
- j. (3 pts) `argv[0]` **char ***
- k. (3 pts) `*c` **double**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #173 Page: 1 Name: _____

173

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

-
1. a. (2 pts) Convert 8073 from base 16 to base 2 1000 0000 0111 0011
- b. (2 pts) Convert 74 from octal to base 2 111 100
- c. (2 pts) Convert a853 from base 16 to binary 1010 1000 0101 0011
- d. (2 pts) Convert 4001 from hexadecimal to binary 0100 0000 0000 0001
- e. (2 pts) Convert 25 from base 8 to base 2 010 101
- f. (2 pts) Convert 0110 1101 0111 1101 from binary to hexadecimal 6d7d
- g. (2 pts) Convert 46 from octal to base 2 100 110
- h. (2 pts) Convert 001 000 from base 2 to octal 10
- i. (2 pts) Convert bb3e from hexadecimal to binary 1011 1011 0011 1110

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -65, what is this number's binary representation in 8-bit two's complement?

10111111

c. (3 pts)

Given that 11101010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-22

d. (3 pts)

Given that 11110001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-15

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

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

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

```
./runIt fig banana grape
```

a. (3 pts) What is the value of `argc` in this case? 4

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) `argv[0]` **char ***
- b. (3 pts) `&y` **Node ****
- c. (3 pts) `y->data` **int**
- d. (3 pts) `t` **int**
- e. (3 pts) `y->next` **Node ***
- f. (3 pts) `*b` **char**
- g. (3 pts) `argc` **int**
- h. (3 pts) `&w` **double ***
- i. (3 pts) `b` **char ***
- j. (3 pts) `y->next->next` **Node ***
- k. (3 pts) `argv[1][2]` **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #174 Page: 2 Name: _____

174

1. a. (2 pts) Convert 177 from decimal to binary **1011 0001**

- b. (2 pts) Convert 1100 0101 0110 1010 from base 2 to hexadecimal **c56a**

- c. (2 pts) Convert 110 111 101 from base 2 to octal **675**

- d. (2 pts) Convert 1001 1100 0100 0011 from binary to hexadecimal **9c43**

- e. (2 pts) Convert 101 010 101 from base 2 to base 8 **525**

- f. (2 pts) Convert 1001 1111 1110 0111 from base 2 to base 16 **9fe7**

- g. (2 pts) Convert 47 from base 8 to binary **100 111**

- h. (2 pts) Convert 0110 1101 from binary to decimal **109**

- i. (2 pts) Convert 5cf9 from hexadecimal to base 2 **0101 1100 1111 1001**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -89, what is this number's binary representation in 8-bit two's complement?

10100111

c. (3 pts)

Given that 10101011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-85

d. (3 pts)

Given that 10100100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-92

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit apple grape
```

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

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

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

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

4. Given the following declarations:

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

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

- a. (3 pts) argv[1][2] **char**
- b. (3 pts) p->data **int**
- c. (3 pts) s **char ***
- d. (3 pts) g **int**
- e. (3 pts) argv[0] **char ***
- f. (3 pts) &f **double ***
- g. (3 pts) &r **int ****
- h. (3 pts) p->next->next **Node ***
- i. (3 pts) argc **int**
- j. (3 pts) *s **char**
- k. (3 pts) p->next **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #175 Page: 1 Name: _____

175

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

2

Exam #175 Page: 2 Name: _____

175

-
1. a. (2 pts) Convert 1111 0110 1011 0101 from binary to hexadecimal **f6b5**
- b. (2 pts) Convert 0111 1111 from base 2 to base 10 **127**
- c. (2 pts) Convert 110 011 001 from base 2 to octal **631**
- d. (2 pts) Convert 12 from octal to base 2 **001 010**
- e. (2 pts) Convert 1000 1101 from base 2 to decimal **141**
- f. (2 pts) Convert 39c4 from hexadecimal to base 2 **0011 1001 1100 0100**
- g. (2 pts) Convert 21 from octal to binary **010 001**
- h. (2 pts) Convert f4cd from base 16 to binary **1111 0100 1100 1101**
- i. (2 pts) Convert 39 from base 10 to binary **0010 0111**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -124, what is this number's binary representation in 8-bit two's complement?

10000100

c. (3 pts)

Given that 10010101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-107

d. (3 pts)

Given that 10111110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-66

3. Assume the `main` function in the program `runit.cpp` starts with:

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

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

```
./runit grape cherry banana lime
```

a. (3 pts) What is the value of `argc` in this case? **5**

b. (3 pts) What is the value of `argv[1][4]`? **e**

c. (3 pts) What is the value of `argv[2][4]`? **r**

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

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int b;  
    double c;  
    Node d;  
    char e;  
    int *f;  
    double *g;  
    Node *h;  
    char *p;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argv[1][2]` **char**
- b. (3 pts) `h->next` **Node ***
- c. (3 pts) `&b` **int ***
- d. (3 pts) `b` **int**
- e. (3 pts) `h->data` **int**
- f. (3 pts) `&f` **int ****
- g. (3 pts) `argc` **int**
- h. (3 pts) `h->next->next` **Node ***
- i. (3 pts) `*f` **int**
- j. (3 pts) `argv[0]` **char ***
- k. (3 pts) `g` **double ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #176 Page: 1 Name: _____

176

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #176 Page: 2 Name: _____

176

-
1. a. (2 pts) Convert 1011 0011 0010 1000 from base 2 to hexadecimal **b328**
- b. (2 pts) Convert 67 from octal to base 2 **110 111**
- c. (2 pts) Convert 55 from octal to base 2 **101 101**
- d. (2 pts) Convert 011 001 100 from base 2 to base 8 **314**
- e. (2 pts) Convert 100 101 001 from base 2 to base 8 **451**
- f. (2 pts) Convert 255 from decimal to binary **1111 1111**
- g. (2 pts) Convert 26 from octal to base 2 **010 110**
- h. (2 pts) Convert 1010 0101 0011 0100 from binary to base 16 **a534**
- i. (2 pts) Convert 1101 0110 1010 0000 from base 2 to hexadecimal **d6a0**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -90, what is this number's binary representation in 8-bit two's complement?

10100110

c. (3 pts)

Given that 11011111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-33

d. (3 pts)

Given that 10001110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-114

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit date lime banana
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][0]`? d

c. (3 pts) What is the value of `argv[0][0]`? .

d. (3 pts) What is the value of `argv[2][1]`? i

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node p;  
    int q;  
    double r;  
    char s;  
    Node *t;  
    int *w;  
    double *x;  
    char *y;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&w` `int **`
- b. (3 pts) `argv[1][2]` `char`
- c. (3 pts) `t->next->next` `Node *`
- d. (3 pts) `argv[0]` `char *`
- e. (3 pts) `&s` `char *`
- f. (3 pts) `argc` `int`
- g. (3 pts) `y` `char *`
- h. (3 pts) `*t` `Node`
- i. (3 pts) `s` `char`
- j. (3 pts) `t->next` `Node *`
- k. (3 pts) `t->data` `int`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

1. a. (2 pts) Convert 1111 1000 from base 2 to decimal **248**

b. (2 pts) Convert 151 from decimal to binary **1001 0111**

c. (2 pts) Convert a3c8 from hexadecimal to base 2 **1010 0011 1100 1000**

d. (2 pts) Convert 111 100 100 from binary to octal **744**

e. (2 pts) Convert 120 from base 10 to binary **0111 1000**

f. (2 pts) Convert 153 from decimal to binary **1001 1001**

g. (2 pts) Convert 1 from base 8 to base 2 **001**

h. (2 pts) Convert 0010 1100 from base 2 to base 10 **44**

i. (2 pts) Convert 1010 0000 1011 0111 from binary to base 16 **a0b7**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -125, what is this number's binary representation in 8-bit two's complement?

10000011

c. (3 pts)

Given that 11001010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-54

d. (3 pts)

Given that 10101000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-88

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana date
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[0][4]`? n

c. (3 pts) What is the value of `argv[1][2]`? n

d. (3 pts) What is the value of `argv[2][1]`? a

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double e;  
    Node f;  
    int g;  
    char h;  
    double *p;  
    Node *q;  
    int *r;  
    char *s;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) argc **int**
- b. (3 pts) *r **int**
- c. (3 pts) &s **char ****
- d. (3 pts) q->next **Node ***
- e. (3 pts) q->data **int**
- f. (3 pts) &g **int ***
- g. (3 pts) argv[0] **char ***
- h. (3 pts) q->next->next **Node ***
- i. (3 pts) argv[1][2] **char**
- j. (3 pts) f **Node**
- k. (3 pts) r **int ***

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #178 Page: 1 Name: _____

178

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #178 Page: 2 Name: _____

178

1. a. (2 pts) Convert 12 from octal to base 2 **001 010**

- b. (2 pts) Convert 0110 1000 from base 2 to base 10 **104**

- c. (2 pts) Convert 110 110 100 from binary to octal **664**

- d. (2 pts) Convert 0100 1110 from base 2 to decimal **78**

- e. (2 pts) Convert ce59 from base 16 to binary **1100 1110 0101 1001**

- f. (2 pts) Convert 1100 1011 1011 1110 from base 2 to base 16 **cbbe**

- g. (2 pts) Convert 1 from octal to binary **001**

- h. (2 pts) Convert 1001 0101 from base 2 to base 10 **149**

- i. (2 pts) Convert 0100 0010 0111 0001 from base 2 to base 16 **4271**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -21, what is this number's binary representation in 8-bit two's complement?

11101011

c. (3 pts)

Given that 10001011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-117

d. (3 pts)

Given that 11011011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-37

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime banana
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[2][4]`? n

c. (3 pts) What is the value of `argv[0][3]`? u

d. (3 pts) What is the value of `argv[1][0]`? l

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node f;  
    double g;  
    int h;  
    char p;  
    Node *q;  
    double *r;  
    int *s;  
    char *t;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `r` `double *`
- b. (3 pts) `argc` `int`
- c. (3 pts) `q->data` `int`
- d. (3 pts) `h` `int`
- e. (3 pts) `argv[0]` `char *`
- f. (3 pts) `*r` `double`
- g. (3 pts) `q->next->next` `Node *`
- h. (3 pts) `q->next` `Node *`
- i. (3 pts) `&r` `double **`
- j. (3 pts) `argv[1][2]` `char`
- k. (3 pts) `&g` `double *`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #179 Page: 2 Name: _____

179

1. a. (2 pts) Convert 110 from base 10 to base 2 **0110 1110**

- b. (2 pts) Convert 0010 0011 0001 1101 from base 2 to hexadecimal **231d**

- c. (2 pts) Convert 62 from base 8 to binary **110 010**

- d. (2 pts) Convert 1101 1010 from base 2 to base 10 **218**

- e. (2 pts) Convert 1011 0001 1010 0101 from base 2 to base 16 **b1a5**

- f. (2 pts) Convert 659a from base 16 to binary **0110 0101 1001 1010**

- g. (2 pts) Convert 54 from base 8 to base 2 **101 100**

- h. (2 pts) Convert 0001 1101 1000 0011 from binary to base 16 **1d83**

- i. (2 pts) Convert 3 from octal to binary **011**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -56, what is this number's binary representation in 8-bit two's complement?

11001000

c. (3 pts)

Given that 11110101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-11

d. (3 pts)

Given that 11100111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-25

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit apple lemon kiwi mango
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[0][0]`? .

c. (3 pts) What is the value of `argv[2][2]`? m

d. (3 pts) What is the value of `argv[1][0]`? a

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double c;  
    int d;  
    Node e;  
    char f;  
    double *g;  
    int *h;  
    Node *p;  
    char *q;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) p->next **Node ***
- b. (3 pts) *q **char**
- c. (3 pts) &q **char ****
- d. (3 pts) argv[1][2] **char**
- e. (3 pts) &c **double ***
- f. (3 pts) p->next->next **Node ***
- g. (3 pts) argv[0] **char ***
- h. (3 pts) c **double**
- i. (3 pts) g **double ***
- j. (3 pts) p->data **int**
- k. (3 pts) argc **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #180 Page: 1 Name: _____

180

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #180 Page: 2 Name: _____

180

-
1. a. (2 pts) Convert 198 from decimal to base 2 **1100 0110**
- b. (2 pts) Convert c5e9 from base 16 to binary **1100 0101 1110 1001**
- c. (2 pts) Convert 011 011 010 from binary to base 8 **332**
- d. (2 pts) Convert 9622 from hexadecimal to base 2 **1001 0110 0010 0010**
- e. (2 pts) Convert 0010 0010 from base 2 to decimal **34**
- f. (2 pts) Convert 0110 0110 from binary to decimal **102**
- g. (2 pts) Convert 3 from decimal to binary **0011**
- h. (2 pts) Convert 110 000 111 from binary to base 8 **607**
- i. (2 pts) Convert 77 from base 8 to binary **111 111**

3

Exam #180 Page: 3 Name: _____

180

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -100, what is this number's binary representation in 8-bit two's complement?

10011100

- c. (3 pts)

Given that 10001000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-120

- d. (3 pts)

Given that 11110100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-12

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry fig lime
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][4]`? n

c. (3 pts) What is the value of `argv[1][4]`? r

d. (3 pts) What is the value of `argv[2][1]`? i

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int t;  
    Node w;  
    double x;  
    char y;  
    int *z;  
    Node *a;  
    double *b;  
    char *c;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argc` **int**
- b. (3 pts) `a->next->next` **Node ***
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `&t` **int ***
- e. (3 pts) `a->next` **Node ***
- f. (3 pts) `*b` **double**
- g. (3 pts) `c` **char ***
- h. (3 pts) `argv[1][2]` **char**
- i. (3 pts) `a->data` **int**
- j. (3 pts) `&b` **double ****
- k. (3 pts) `t` **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #181 Page: 2 Name: _____

181

-
1. a. (2 pts) Convert 1011 1000 0100 from base 2 to hexadecimal **b84**
- b. (2 pts) Convert 1000 0000 0001 0010 from binary to base 16 **8012**
- c. (2 pts) Convert 0101 1011 0100 1010 from base 2 to hexadecimal **5b4a**
- d. (2 pts) Convert 220c from hexadecimal to binary **0010 0010 0000 1100**
- e. (2 pts) Convert 5 from base 10 to binary **0101**
- f. (2 pts) Convert 001 from binary to octal **1**
- g. (2 pts) Convert 172 from base 10 to binary **1010 1100**
- h. (2 pts) Convert 0100 1011 from binary to base 10 **75**
- i. (2 pts) Convert 1100 1001 0001 0011 from binary to base 16 **c913**

3

Exam #181 Page: 3 Name: _____

181

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -6, what is this number's binary representation in 8-bit two's complement?

1111010

- c. (3 pts)

Given that 11110011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-13

- d. (3 pts)

Given that 10001110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-114

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit grape date cherry
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][4]`? e

c. (3 pts) What is the value of `argv[0][2]`? r

d. (3 pts) What is the value of `argv[2][2]`? t

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node b;  
    int c;  
    double d;  
    char e;  
    Node *f;  
    int *g;  
    double *h;  
    char *p;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argc` **int**
- b. (3 pts) `d` **double**
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `&e` **char ***
- e. (3 pts) `h` **double ***
- f. (3 pts) `f->next` **Node ***
- g. (3 pts) `argv[1][2]` **char**
- h. (3 pts) `f->next->next` **Node ***
- i. (3 pts) `f->data` **int**
- j. (3 pts) `*p` **char**
- k. (3 pts) `&f` **Node ****

6

Exam #181 Page: 6 Name: _____

181

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #182 Page: 1 Name: _____

182

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #182 Page: 2 Name: _____

182

1. a. (2 pts) Convert 3ce6 from hexadecimal to base 2 **0011 1100 1110 0110**

b. (2 pts) Convert 5197 from base 16 to binary **0101 0001 1001 0111**

c. (2 pts) Convert 1001 0001 from base 2 to base 10 **145**

d. (2 pts) Convert 0111 1110 0100 1110 from binary to base 16 **7e4e**

e. (2 pts) Convert 5b87 from base 16 to binary **0101 1011 1000 0111**

f. (2 pts) Convert 0011 0011 from base 2 to base 10 **51**

g. (2 pts) Convert 53 from octal to base 2 **101 011**

h. (2 pts) Convert 1011 0100 from base 2 to base 10 **180**

i. (2 pts) Convert 32 from base 8 to binary **011 010**

3

Exam #182 Page: 3 Name: _____

182

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -31, what is this number's binary representation in 8-bit two's complement?

11100001

- c. (3 pts)

Given that 10110100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-76

- d. (3 pts)

Given that 11000001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-63

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime mango
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[2][4]`? o

c. (3 pts) What is the value of `argv[0][6]`? t

d. (3 pts) What is the value of `argv[1][1]`? i

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node b;  
    double c;  
    int d;  
    char e;  
    Node *f;  
    double *g;  
    int *h;  
    char *p;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argv[1][2]` **char**
- b. (3 pts) `f->next->next` **Node ***
- c. (3 pts) `&h` **int ****
- d. (3 pts) `f->next` **Node ***
- e. (3 pts) `e` **char**
- f. (3 pts) `f->data` **int**
- g. (3 pts) `g` **double ***
- h. (3 pts) `argc` **int**
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `&c` **double ***
- k. (3 pts) `*p` **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #183 Page: 2 Name: _____

183

-
1. a. (2 pts) Convert 40 from octal to base 2 **100 000**
- b. (2 pts) Convert 010 111 from base 2 to base 8 **27**
- c. (2 pts) Convert 0111 1111 from binary to base 10 **127**
- d. (2 pts) Convert 2 from octal to binary **010**
- e. (2 pts) Convert 17 from base 8 to base 2 **001 111**
- f. (2 pts) Convert 110 011 010 from base 2 to base 8 **632**
- g. (2 pts) Convert 89 from base 10 to binary **0101 1001**
- h. (2 pts) Convert 0011 1100 0100 0101 from base 2 to base 16 **3c45**
- i. (2 pts) Convert 0011 0100 1110 0101 from binary to base 16 **34e5**

3

Exam #183 Page: 3 Name: _____

183

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -65, what is this number's binary representation in 8-bit two's complement?

1011111

- c. (3 pts)

Given that 10011110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-98

- d. (3 pts)

Given that 11011011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-37

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava lime cherry grape
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][1]`? u

c. (3 pts) What is the value of `argv[2][2]`? m

d. (3 pts) What is the value of `argv[0][1]`? /

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int y;  
    double z;  
    Node a;  
    char b;  
    int *c;  
    double *d;  
    Node *e;  
    char *f;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `e->next->next` **Node ***
- b. (3 pts) `z` **double**
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `e->data` **int**
- e. (3 pts) `e->next` **Node ***
- f. (3 pts) `&c` **int ****
- g. (3 pts) `*c` **int**
- h. (3 pts) `&z` **double ***
- i. (3 pts) `argc` **int**
- j. (3 pts) `c` **int ***
- k. (3 pts) `argv[1][2]` **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #184 Page: 1 Name: _____

184

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #184 Page: 2 Name: _____

184

-
1. a. (2 pts) Convert 110 110 100 from binary to base 8 **664**
- b. (2 pts) Convert 1010 1110 1000 1100 from base 2 to hexadecimal **ae8c**
- c. (2 pts) Convert 36 from base 10 to binary **0010 0100**
- d. (2 pts) Convert 197 from decimal to binary **1100 0101**
- e. (2 pts) Convert 1010 1111 from base 2 to decimal **175**
- f. (2 pts) Convert 1100 1110 0101 1111 from base 2 to base 16 **ce5f**
- g. (2 pts) Convert 170 from decimal to base 2 **1010 1010**
- h. (2 pts) Convert 1110 0010 from base 2 to base 10 **226**
- i. (2 pts) Convert 0010 0111 0101 1000 from base 2 to hexadecimal **2758**

3

Exam #184 Page: 3 Name: _____

184

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -110, what is this number's binary representation in 8-bit two's complement?

10010010

- c. (3 pts)

Given that 10110010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-78

- d. (3 pts)

Given that 11011010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-38

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit mango guava date
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[2][4]`? a

c. (3 pts) What is the value of `argv[1][4]`? o

d. (3 pts) What is the value of `argv[0][2]`? r

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int p;  
    Node q;  
    double r;  
    char s;  
    int *t;  
    Node *w;  
    double *x;  
    char *y;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&p` **int ***
- b. (3 pts) `argc` **int**
- c. (3 pts) `w->data` **int**
- d. (3 pts) `&y` **char ****
- e. (3 pts) `*t` **int**
- f. (3 pts) `argv[0]` **char ***
- g. (3 pts) `w->next` **Node ***
- h. (3 pts) `w->next->next` **Node ***
- i. (3 pts) `q` **Node**
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `x` **double ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #185 Page: 1 Name: _____

185

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

1. a. (2 pts) Convert 31 from decimal to base 2 0001 1111
- b. (2 pts) Convert 32 from octal to base 2 011 010
- c. (2 pts) Convert 0001 0010 from base 2 to decimal 18
- d. (2 pts) Convert 0101 0001 from binary to base 10 81
- e. (2 pts) Convert 9296 from base 16 to base 2 1001 0010 1001 0110
- f. (2 pts) Convert 683b from hexadecimal to binary 0110 1000 0011 1011
- g. (2 pts) Convert 84 from base 10 to base 2 0101 0100
- h. (2 pts) Convert 0110 1010 from binary to base 10 106
- i. (2 pts) Convert f170 from hexadecimal to base 2 1111 0001 0111 0000

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -16, what is this number's binary representation in 8-bit two's complement?

11110000

c. (3 pts)

Given that 10011100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-100

d. (3 pts)

Given that 11110100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-12

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit fig lemon mango
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[2][1]`? e

c. (3 pts) What is the value of `argv[0][5]`? I

d. (3 pts) What is the value of `argv[1][2]`? g

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int f;  
    Node g;  
    double h;  
    char p;  
    int *q;  
    Node *r;  
    double *s;  
    char *t;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) r->next->next **Node ***
- b. (3 pts) p **char**
- c. (3 pts) &p **char ***
- d. (3 pts) r->next **Node ***
- e. (3 pts) *r **Node**
- f. (3 pts) r **Node ***
- g. (3 pts) &r **Node ****
- h. (3 pts) argv[0] **char ***
- i. (3 pts) argv[1][2] **char**
- j. (3 pts) r->data **int**
- k. (3 pts) argc **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #186 Page: 1 Name: _____

186

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #186 Page: 2 Name: _____

186

-
1. a. (2 pts) Convert 0101 0000 from base 2 to decimal **80**
- b. (2 pts) Convert 001 110 100 from binary to octal **164**
- c. (2 pts) Convert 0100 1001 0010 1010 from base 2 to hexadecimal **492a**
- d. (2 pts) Convert 1010 1101 1111 0111 from binary to base 16 **adf7**
- e. (2 pts) Convert 1110 1000 1011 0100 from base 2 to base 16 **e8b4**
- f. (2 pts) Convert 46 from octal to base 2 **100 110**
- g. (2 pts) Convert 86 from decimal to binary **0101 0110**
- h. (2 pts) Convert d356 from hexadecimal to base 2 **1101 0011 0101 0110**
- i. (2 pts) Convert 1001 0011 0010 1010 from base 2 to hexadecimal **932a**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -41, what is this number's binary representation in 8-bit two's complement?

11010111

c. (3 pts)

Given that 11011101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-35

d. (3 pts)

Given that 10100111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-89

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit apple fig
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[0][6]`? t

c. (3 pts) What is the value of `argv[2][1]`? i

d. (3 pts) What is the value of `argv[1][3]`? l

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double x;  
    Node y;  
    int z;  
    char a;  
    double *b;  
    Node *c;  
    int *d;  
    char *e;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&d` **int ****
- b. (3 pts) `argv[1][2]` **char**
- c. (3 pts) `*c` **Node**
- d. (3 pts) `c->next` **Node ***
- e. (3 pts) `c->data` **int**
- f. (3 pts) `argc` **int**
- g. (3 pts) `b` **double ***
- h. (3 pts) `x` **double**
- i. (3 pts) `&z` **int ***
- j. (3 pts) `c->next->next` **Node ***
- k. (3 pts) `argv[0]` **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

1. a. (2 pts) Convert 9543 from base 16 to base 2 **1001 0101 0100 0011**

b. (2 pts) Convert 1111 0100 from base 2 to base 10 **244**

c. (2 pts) Convert 55 from decimal to binary **0011 0111**

d. (2 pts) Convert 57 from decimal to base 2 **0011 1001**

e. (2 pts) Convert 110 010 111 from base 2 to octal **627**

f. (2 pts) Convert 0011 0100 1000 0010 from base 2 to base 16 **3482**

g. (2 pts) Convert 0 from base 10 to base 2 **0000**

h. (2 pts) Convert 0101 1011 0000 0110 from base 2 to base 16 **5b06**

i. (2 pts) Convert 5d41 from hexadecimal to base 2 **0101 1101 0100 0001**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -75, what is this number's binary representation in 8-bit two's complement?

10110101

c. (3 pts)

Given that 11000111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-57

d. (3 pts)

Given that 11000001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-63

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit date mango grape apple
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][2]`? t

c. (3 pts) What is the value of `argv[2][1]`? a

d. (3 pts) What is the value of `argv[0][4]`? n

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double s;  
    int t;  
    Node w;  
    char x;  
    double *y;  
    int *z;  
    Node *a;  
    char *b;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) a->next->next **Node ***
- b. (3 pts) argv[1][2] **char**
- c. (3 pts) *a **Node**
- d. (3 pts) &t **int ***
- e. (3 pts) a->next **Node ***
- f. (3 pts) x **char**
- g. (3 pts) &y **double ****
- h. (3 pts) argc **int**
- i. (3 pts) b **char ***
- j. (3 pts) argv[0] **char ***
- k. (3 pts) a->data **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #188 Page: 2 Name: _____

188

-
1. a. (2 pts) Convert ed9f from base 16 to base 2 **1110 1101 1001 1111**
- b. (2 pts) Convert 100 101 110 from base 2 to octal **456**
- c. (2 pts) Convert dc6c from base 16 to base 2 **1101 1100 0110 1100**
- d. (2 pts) Convert 111 101 010 from base 2 to octal **752**
- e. (2 pts) Convert 0011 1100 0111 1000 from base 2 to base 16 **3c78**
- f. (2 pts) Convert 53 from base 10 to base 2 **0011 0101**
- g. (2 pts) Convert 24 from octal to binary **010 100**
- h. (2 pts) Convert 0001 from base 2 to decimal **1**
- i. (2 pts) Convert 4fb5 from base 16 to binary **0100 1111 1011 0101**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -119, what is this number's binary representation in 8-bit two's complement?

10001001

c. (3 pts)

Given that 11011011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-37

d. (3 pts)

Given that 11000000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-64

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana guava kiwi
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][6]`? t

c. (3 pts) What is the value of `argv[1][4]`? n

d. (3 pts) What is the value of `argv[2][0]`? g

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node e;  
    int f;  
    double g;  
    char h;  
    Node *p;  
    int *q;  
    double *r;  
    char *s;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `r` `double *`
- b. (3 pts) `p->data` `int`
- c. (3 pts) `p->next->next` `Node *`
- d. (3 pts) `argv[1][2]` `char`
- e. (3 pts) `argv[0]` `char *`
- f. (3 pts) `p->next` `Node *`
- g. (3 pts) `g` `double`
- h. (3 pts) `*r` `double`
- i. (3 pts) `&s` `char **`
- j. (3 pts) `&f` `int *`
- k. (3 pts) `argc` `int`

6

Exam #188 Page: 6 Name: _____

188

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #189 Page: 2 Name: _____

189

1. a. (2 pts) Convert 001 100 100 from base 2 to base 8 **144**

- b. (2 pts) Convert 5158 from base 16 to base 2 **0101 0001 0101 1000**

- c. (2 pts) Convert 1100 1010 from binary to base 10 **202**

- d. (2 pts) Convert 100 000 010 from base 2 to base 8 **402**

- e. (2 pts) Convert 111 111 from base 2 to base 8 **77**

- f. (2 pts) Convert 1100 1111 from base 2 to base 10 **207**

- g. (2 pts) Convert 77 from base 8 to binary **111 111**

- h. (2 pts) Convert 8928 from base 16 to binary **1000 1001 0010 1000**

- i. (2 pts) Convert 0001 1001 1100 1100 from binary to base 16 **19cc**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -26, what is this number's binary representation in 8-bit two's complement?

11100110

c. (3 pts)

Given that 11000101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-59

d. (3 pts)

Given that 11011001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-39

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime kiwi fig
```

a. (3 pts) What is the value of `argc` in this case? **4**

b. (3 pts) What is the value of `argv[0][4]`? **n**

c. (3 pts) What is the value of `argv[1][1]`? **i**

d. (3 pts) What is the value of `argv[2][0]`? **k**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node b;  
    int c;  
    double d;  
    char e;  
    Node *f;  
    int *g;  
    double *h;  
    char *p;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `f->next` **Node ***
- b. (3 pts) `argv[0]` **char ***
- c. (3 pts) `&g` **int ****
- d. (3 pts) `argc` **int**
- e. (3 pts) `argv[1][2]` **char**
- f. (3 pts) `*p` **char**
- g. (3 pts) `&b` **Node ***
- h. (3 pts) `b` **Node**
- i. (3 pts) `f->next->next` **Node ***
- j. (3 pts) `g` **int ***
- k. (3 pts) `f->data` **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #190 Page: 1 Name: _____

190

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #190 Page: 2 Name: _____

190

1. a. (2 pts) Convert 30 from octal to binary **011 000**

- b. (2 pts) Convert 34 from decimal to binary **0010 0010**

- c. (2 pts) Convert 0 from octal to base 2 **000**

- d. (2 pts) Convert 1101 1101 from binary to decimal **221**

- e. (2 pts) Convert 35 from base 8 to base 2 **011 101**

- f. (2 pts) Convert 2 from decimal to binary **0010**

- g. (2 pts) Convert 1111 1110 1010 0111 from binary to hexadecimal **fea7**

- h. (2 pts) Convert 1111 0010 0001 0111 from binary to base 16 **f217**

- i. (2 pts) Convert 1011 1011 1000 0111 from base 2 to hexadecimal **bb87**

3

Exam #190 Page: 3 Name: _____

190

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -50, what is this number's binary representation in 8-bit two's complement?

11001110

- c. (3 pts)

Given that 10000110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-122

- d. (3 pts)

Given that 10001101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-115

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit kiwi cherry
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[0][4]`? n

c. (3 pts) What is the value of `argv[2][4]`? r

d. (3 pts) What is the value of `argv[1][3]`? i

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node s;  
    double t;  
    int w;  
    char x;  
    Node *y;  
    double *z;  
    int *a;  
    char *b;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `y` **Node ***
- b. (3 pts) `y->next->next` **Node ***
- c. (3 pts) `*b` **char**
- d. (3 pts) `argv[1][2]` **char**
- e. (3 pts) `argc` **int**
- f. (3 pts) `&x` **char ***
- g. (3 pts) `t` **double**
- h. (3 pts) `&b` **char ****
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `y->data` **int**
- k. (3 pts) `y->next` **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #191 Page: 1 Name: _____

191

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #191 Page: 2 Name: _____

191

1. a. (2 pts) Convert 1010 1000 from base 2 to decimal **168**

b. (2 pts) Convert 67 from octal to base 2 **110 111**

c. (2 pts) Convert 73 from base 8 to base 2 **111 011**

d. (2 pts) Convert 32 from octal to base 2 **011 010**

e. (2 pts) Convert 89 from base 10 to binary **0101 1001**

f. (2 pts) Convert 1001 1011 from base 2 to base 10 **155**

g. (2 pts) Convert 52 from base 8 to binary **101 010**

h. (2 pts) Convert 011 110 011 from binary to octal **363**

i. (2 pts) Convert 41 from octal to base 2 **100 001**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -85, what is this number's binary representation in 8-bit two's complement?

10101011

c. (3 pts)

Given that 11110001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-15

d. (3 pts)

Given that 10100110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-90

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango kiwi banana date
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][4]`? o

c. (3 pts) What is the value of `argv[2][1]`? i

d. (3 pts) What is the value of `argv[0][6]`? t

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double p;  
    int q;  
    Node r;  
    char s;  
    double *t;  
    int *w;  
    Node *x;  
    char *y;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `y` `char *`
- b. (3 pts) `&w` `int **`
- c. (3 pts) `x->next` `Node *`
- d. (3 pts) `x->data` `int`
- e. (3 pts) `argc` `int`
- f. (3 pts) `argv[0]` `char *`
- g. (3 pts) `x->next->next` `Node *`
- h. (3 pts) `argv[1][2]` `char`
- i. (3 pts) `*t` `double`
- j. (3 pts) `p` `double`
- k. (3 pts) `&r` `Node *`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #192 Page: 1 Name: _____

192

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #192 Page: 2 Name: _____

192

1. a. (2 pts) Convert 157 from base 10 to base 2 **1001 1101**

- b. (2 pts) Convert 1100 0100 from base 2 to base 10 **196**

- c. (2 pts) Convert 010 011 111 from base 2 to octal **237**

- d. (2 pts) Convert 101 000 100 from binary to base 8 **504**

- e. (2 pts) Convert 3311 from base 16 to base 2 **0011 0011 0001 0001**

- f. (2 pts) Convert d8e3 from base 16 to binary **1101 1000 1110 0011**

- g. (2 pts) Convert 56 from base 10 to binary **0011 1000**

- h. (2 pts) Convert 1645 from base 16 to base 2 **0001 0110 0100 0101**

- i. (2 pts) Convert 1011 1010 from binary to decimal **186**

3

Exam #192 Page: 3 Name: _____

192

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -79, what is this number's binary representation in 8-bit two's complement?

10110001

- c. (3 pts)

Given that 11001110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-50

- d. (3 pts)

Given that 11010100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-44

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit apple cherry lemon
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][2]`? r

c. (3 pts) What is the value of `argv[1][0]`? a

d. (3 pts) What is the value of `argv[2][5]`? y

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node e;  
    int f;  
    double g;  
    char h;  
    Node *p;  
    int *q;  
    double *r;  
    char *s;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `s` **char ***
- b. (3 pts) `p->data` **int**
- c. (3 pts) `&q` **int ****
- d. (3 pts) `argv[1][2]` **char**
- e. (3 pts) `p->next` **Node ***
- f. (3 pts) `e` **Node**
- g. (3 pts) `p->next->next` **Node ***
- h. (3 pts) `&f` **int ***
- i. (3 pts) `argc` **int**
- j. (3 pts) `argv[0]` **char ***
- k. (3 pts) `*s` **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #193 Page: 2 Name: _____

193

-
1. a. (2 pts) Convert 1110 0001 1110 0101 from binary to base 16 **e1e5**
- b. (2 pts) Convert 127 from base 10 to binary **0111 1111**
- c. (2 pts) Convert 0011 1101 from binary to decimal **61**
- d. (2 pts) Convert 0010 1110 0100 1101 from binary to hexadecimal **2e4d**
- e. (2 pts) Convert 0001 0110 0101 1101 from base 2 to hexadecimal **165d**
- f. (2 pts) Convert 114 from base 10 to binary **0111 0010**
- g. (2 pts) Convert 226 from decimal to binary **1110 0010**
- h. (2 pts) Convert 9df5 from base 16 to base 2 **1001 1101 1111 0101**
- i. (2 pts) Convert 1000 0101 from base 2 to base 10 **133**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -113, what is this number's binary representation in 8-bit two's complement?

10001111

c. (3 pts)

Given that 10111000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-72

d. (3 pts)

Given that 11101110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-18

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime mango
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[2][3]`? g

c. (3 pts) What is the value of `argv[0][2]`? r

d. (3 pts) What is the value of `argv[1][2]`? m

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node a;  
    double b;  
    int c;  
    char d;  
    Node *e;  
    double *f;  
    int *g;  
    char *h;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) e->data **int**
- b. (3 pts) argv[0] **char ***
- c. (3 pts) e->next->next **Node ***
- d. (3 pts) &b **double ***
- e. (3 pts) &h **char ****
- f. (3 pts) f **double ***
- g. (3 pts) c **int**
- h. (3 pts) argv[1][2] **char**
- i. (3 pts) *f **double**
- j. (3 pts) argc **int**
- k. (3 pts) e->next **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #194 Page: 1 Name: _____

194

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #194 Page: 2 Name: _____

194

1. a. (2 pts) Convert 1347 from base 16 to binary **0001 0011 0100 0111**

b. (2 pts) Convert 0101 0000 from binary to base 10 **80**

c. (2 pts) Convert 0111 0100 0011 1001 from binary to hexadecimal **7439**

d. (2 pts) Convert 1000 1010 from binary to base 10 **138**

e. (2 pts) Convert 33 from base 8 to binary **011 011**

f. (2 pts) Convert 51 from octal to base 2 **101 001**

g. (2 pts) Convert 1110 0100 from base 2 to base 10 **228**

h. (2 pts) Convert 0110 1110 0100 from binary to base 16 **6e4**

i. (2 pts) Convert 0010 0110 from binary to decimal **38**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -10, what is this number's binary representation in 8-bit two's complement?

11110110

c. (3 pts)

Given that 11111001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-7

d. (3 pts)

Given that 10100010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-94

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit mango guava
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[0][0]`? .

c. (3 pts) What is the value of `argv[2][2]`? a

d. (3 pts) What is the value of `argv[1][2]`? n

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double r;  
    Node s;  
    int t;  
    char w;  
    double *x;  
    Node *y;  
    int *z;  
    char *a;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) y->next->next **Node ***
- b. (3 pts) w **char**
- c. (3 pts) argv[0] **char ***
- d. (3 pts) argv[1][2] **char**
- e. (3 pts) &w **char ***
- f. (3 pts) argc **int**
- g. (3 pts) *x **double**
- h. (3 pts) y **Node ***
- i. (3 pts) y->next **Node ***
- j. (3 pts) &y **Node ****
- k. (3 pts) y->data **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #195 Page: 1 Name: _____

195

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

1. a. (2 pts) Convert 26 from base 8 to binary 010 110
- b. (2 pts) Convert 1010 1100 0010 from binary to hexadecimal ac2
- c. (2 pts) Convert 98 from decimal to binary 0110 0010
- d. (2 pts) Convert 5 from octal to binary 101
- e. (2 pts) Convert 0100 1111 from base 2 to decimal 79
- f. (2 pts) Convert 0011 1111 0000 0110 from binary to hexadecimal 3f06
- g. (2 pts) Convert 1000 1110 from binary to decimal 142
- h. (2 pts) Convert 100 011 101 from binary to base 8 435
- i. (2 pts) Convert 240 from base 10 to base 2 1111 0000

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -45, what is this number's binary representation in 8-bit two's complement?

11010011

c. (3 pts)

Given that 11100011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-29

d. (3 pts)

Given that 10111011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-69

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava date cherry mango
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][4]`? a

c. (3 pts) What is the value of `argv[2][3]`? e

d. (3 pts) What is the value of `argv[0][3]`? u

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double h;  
    int p;  
    Node q;  
    char r;  
    double *s;  
    int *t;  
    Node *w;  
    char *x;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&x` **char ****
- b. (3 pts) `w->data` **int**
- c. (3 pts) `&q` **Node ***
- d. (3 pts) `w->next->next` **Node ***
- e. (3 pts) `w->next` **Node ***
- f. (3 pts) `argc` **int**
- g. (3 pts) `q` **Node**
- h. (3 pts) `argv[0]` **char ***
- i. (3 pts) `argv[1][2]` **char**
- j. (3 pts) `*w` **Node**
- k. (3 pts) `s` **double ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #196 Page: 1 Name: _____

196

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #196 Page: 2 Name: _____

196

1. a. (2 pts) Convert 54 from base 8 to binary **101 100**

- b. (2 pts) Convert ad8e from hexadecimal to binary **1010 1101 1000 1110**

- c. (2 pts) Convert 77b from hexadecimal to base 2 **0111 0111 1011**

- d. (2 pts) Convert d20c from hexadecimal to binary **1101 0010 0000 1100**

- e. (2 pts) Convert c03f from base 16 to binary **1100 0000 0011 1111**

- f. (2 pts) Convert 010 000 000 from base 2 to octal **200**

- g. (2 pts) Convert 1110 0000 0010 0001 from base 2 to base 16 **e021**

- h. (2 pts) Convert 3506 from hexadecimal to base 2 **0011 0101 0000 0110**

- i. (2 pts) Convert 111 000 110 from binary to octal **706**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -89, what is this number's binary representation in 8-bit two's complement?

10100111

c. (3 pts)

Given that 11110111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-9

d. (3 pts)

Given that 10111010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-70

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit banana date guava
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][0]`? .

c. (3 pts) What is the value of `argv[1][4]`? n

d. (3 pts) What is the value of `argv[2][3]`? e

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node a;  
    int b;  
    double c;  
    char d;  
    Node *e;  
    int *f;  
    double *g;  
    char *h;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) e->next->next **Node ***
- b. (3 pts) e->data **int**
- c. (3 pts) &c **double ***
- d. (3 pts) argv[1][2] **char**
- e. (3 pts) g **double ***
- f. (3 pts) &g **double ****
- g. (3 pts) *f **int**
- h. (3 pts) argc **int**
- i. (3 pts) b **int**
- j. (3 pts) e->next **Node ***
- k. (3 pts) argv[0] **char ***

6

Exam #196 Page: 6 Name: _____

196

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #197 Page: 2 Name: _____

197

1. a. (2 pts) Convert 245 from base 10 to base 2 **1111 0101**

- b. (2 pts) Convert 0110 0111 1011 0111 from base 2 to base 16 **67b7**

- c. (2 pts) Convert 1111 0101 from binary to base 10 **245**

- d. (2 pts) Convert 5df6 from hexadecimal to base 2 **0101 1101 1111 0110**

- e. (2 pts) Convert 50 from base 8 to binary **101 000**

- f. (2 pts) Convert 1101 1010 from base 2 to decimal **218**

- g. (2 pts) Convert 1000 1010 0001 0001 from binary to base 16 **8a11**

- h. (2 pts) Convert 1011 1100 1011 0111 from base 2 to base 16 **bcb7**

- i. (2 pts) Convert 1010 1101 from base 2 to base 10 **173**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -123, what is this number's binary representation in 8-bit two's complement?

10000101

c. (3 pts)

Given that 11100001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-31

d. (3 pts)

Given that 11010100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-44

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime apple
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[1][2]`? m

c. (3 pts) What is the value of `argv[0][5]`? I

d. (3 pts) What is the value of `argv[2][0]`? a

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double x;  
    Node y;  
    int z;  
    char a;  
    double *b;  
    Node *c;  
    int *d;  
    char *e;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) *e **char**
- b. (3 pts) c->next **Node ***
- c. (3 pts) c->data **int**
- d. (3 pts) argc **int**
- e. (3 pts) c **Node ***
- f. (3 pts) &b **double ****
- g. (3 pts) argv[0] **char ***
- h. (3 pts) &y **Node ***
- i. (3 pts) argv[1][2] **char**
- j. (3 pts) c->next->next **Node ***
- k. (3 pts) a **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #198 Page: 2 Name: _____

198

1. a. (2 pts) Convert 0010 0110 from binary to decimal **38**

b. (2 pts) Convert 16 from base 8 to binary **001 110**

c. (2 pts) Convert 12 from octal to base 2 **001 010**

d. (2 pts) Convert 1011 1010 0011 0111 from binary to hexadecimal **ba37**

e. (2 pts) Convert 111 110 011 from base 2 to octal **763**

f. (2 pts) Convert 12 from decimal to base 2 **1100**

g. (2 pts) Convert 1000 1100 0011 1111 from base 2 to base 16 **8c3f**

h. (2 pts) Convert 001 001 011 from binary to base 8 **113**

i. (2 pts) Convert 010 011 110 from binary to octal **236**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -20, what is this number's binary representation in 8-bit two's complement?

11101100

c. (3 pts)

Given that 10100010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-94

d. (3 pts)

Given that 10000111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-121

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava cherry
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[0][4]`? n

c. (3 pts) What is the value of `argv[2][4]`? r

d. (3 pts) What is the value of `argv[1][0]`? g

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node g;  
    double h;  
    int p;  
    char q;  
    Node *r;  
    double *s;  
    int *t;  
    char *w;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argc` **int**
- b. (3 pts) `r->next->next` **Node ***
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `r->next` **Node ***
- e. (3 pts) `&t` **int ****
- f. (3 pts) `r` **Node ***
- g. (3 pts) `r->data` **int**
- h. (3 pts) `*t` **int**
- i. (3 pts) `&g` **Node ***
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `h` **double**

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #199 Page: 2 Name: _____

199

1. a. (2 pts) Convert 6ba4 from hexadecimal to binary **0110 1011 1010 0100**

- b. (2 pts) Convert 111 100 110 from binary to base 8 **746**

- c. (2 pts) Convert 6 from base 8 to base 2 **110**

- d. (2 pts) Convert 21 from octal to base 2 **010 001**

- e. (2 pts) Convert 220 from base 10 to base 2 **1101 1100**

- f. (2 pts) Convert 1010 0110 from binary to base 10 **166**

- g. (2 pts) Convert 0011 0110 0011 0000 from binary to base 16 **3630**

- h. (2 pts) Convert 101 011 010 from binary to base 8 **532**

- i. (2 pts) Convert 0001 1001 from base 2 to base 10 **25**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -54, what is this number's binary representation in 8-bit two's complement?

11001010

c. (3 pts)

Given that 10001101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-115

d. (3 pts)

Given that 10010011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-109

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime grape fig apple
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][2]`? m

c. (3 pts) What is the value of `argv[2][3]`? p

d. (3 pts) What is the value of `argv[0][6]`? t

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int d;  
    double e;  
    Node f;  
    char g;  
    int *h;  
    double *p;  
    Node *q;  
    char *r;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) q->next **Node ***
- b. (3 pts) argv[1][2] **char**
- c. (3 pts) *h **int**
- d. (3 pts) argc **int**
- e. (3 pts) q->next->next **Node ***
- f. (3 pts) &g **char ***
- g. (3 pts) argv[0] **char ***
- h. (3 pts) g **char**
- i. (3 pts) r **char ***
- j. (3 pts) q->data **int**
- k. (3 pts) &h **int ****

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #200 Page: 1 Name: _____

200

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #200 Page: 2 Name: _____

200

1. a. (2 pts) Convert c400 from hexadecimal to binary **1100 0100 0000 0000**

b. (2 pts) Convert 1001 0110 0011 0010 from binary to hexadecimal **9632**

c. (2 pts) Convert 1011 1110 1111 1101 from binary to hexadecimal **befd**

d. (2 pts) Convert 011 from binary to octal **3**

e. (2 pts) Convert 010 011 010 from base 2 to octal **232**

f. (2 pts) Convert 1010 0111 1100 1010 from binary to base 16 **a7ca**

g. (2 pts) Convert 41 from octal to binary **100 001**

h. (2 pts) Convert 53c8 from hexadecimal to base 2 **0101 0011 1100 1000**

i. (2 pts) Convert 1011 from binary to decimal **11**

3

Exam #200 Page: 3 Name: _____

200

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -99, what is this number's binary representation in 8-bit two's complement?

10011101

c. (3 pts)

Given that 10100000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-96

d. (3 pts)

Given that 10100110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-90

4

Exam #200 Page: 4 Name: _____

200

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit apple banana lemon
```

a. (3 pts) What is the value of `argc` in this case? **4**

b. (3 pts) What is the value of `argv[1][4]`? **e**

c. (3 pts) What is the value of `argv[0][1]`? **/**

d. (3 pts) What is the value of `argv[2][2]`? **n**

5

Exam #200 Page: 5 Name: _____

200

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int w;  
    Node x;  
    double y;  
    char z;  
    int *a;  
    Node *b;  
    double *c;  
    char *d;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argv[0]` **char ***
- b. (3 pts) `*d` **char**
- c. (3 pts) `b->data` **int**
- d. (3 pts) `y` **double**
- e. (3 pts) `&y` **double ***
- f. (3 pts) `b->next` **Node ***
- g. (3 pts) `c` **double ***
- h. (3 pts) `argv[1][2]` **char**
- i. (3 pts) `argc` **int**
- j. (3 pts) `&d` **char ****
- k. (3 pts) `b->next->next` **Node ***

6

Exam #200 Page: 6 Name: _____

200

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #201 Page: 1 Name: _____

201

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #201 Page: 2 Name: _____

201

1. a. (2 pts) Convert 010 001 from binary to octal **21**

- b. (2 pts) Convert 0101 0000 from binary to base 10 **80**

- c. (2 pts) Convert 1010 1100 1101 1101 from base 2 to hexadecimal **acdd**

- d. (2 pts) Convert 1000 1101 from base 2 to decimal **141**

- e. (2 pts) Convert 48 from base 10 to base 2 **0011 0000**

- f. (2 pts) Convert 0100 0001 1010 0111 from base 2 to hexadecimal **41a7**

- g. (2 pts) Convert 0011 0001 from base 2 to decimal **49**

- h. (2 pts) Convert 110 110 110 from binary to base 8 **666**

- i. (2 pts) Convert 110 101 011 from base 2 to base 8 **653**

3

Exam #201 Page: 3 Name: _____

201

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -5, what is this number's binary representation in 8-bit two's complement?

1111011

- c. (3 pts)

Given that 10001010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-118

- d. (3 pts)

Given that 10111010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-70

4

Exam #201 Page: 4 Name: _____

201

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit date banana mango
```

a. (3 pts) What is the value of `argc` in this case? **4**

b. (3 pts) What is the value of `argv[0][2]`? **r**

c. (3 pts) What is the value of `argv[2][0]`? **b**

d. (3 pts) What is the value of `argv[1][0]`? **d**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int r;  
    Node s;  
    double t;  
    char w;  
    int *x;  
    Node *y;  
    double *z;  
    char *a;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argv[1][2]` **char**
- b. (3 pts) `&y` **Node ****
- c. (3 pts) `y` **Node ***
- d. (3 pts) `argv[0]` **char ***
- e. (3 pts) `y->next->next` **Node ***
- f. (3 pts) `s` **Node**
- g. (3 pts) `argc` **int**
- h. (3 pts) `y->next` **Node ***
- i. (3 pts) `y->data` **int**
- j. (3 pts) `*x` **int**
- k. (3 pts) `&t` **double ***

6

Exam #201 Page: 6 Name: _____

201

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

7

Exam #201 Page: 7 Name: _____

201

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #202 Page: 1 Name: _____

202

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #202 Page: 2 Name: _____

202

1. a. (2 pts) Convert 0011 1010 0100 0010 from base 2 to base 16 **3a42**

b. (2 pts) Convert 001 000 011 from base 2 to octal **103**

c. (2 pts) Convert 227 from base 10 to base 2 **1110 0011**

d. (2 pts) Convert 233 from decimal to binary **1110 1001**

e. (2 pts) Convert 1000 0110 from base 2 to base 10 **134**

f. (2 pts) Convert 011 101 000 from base 2 to base 8 **350**

g. (2 pts) Convert 0011 0100 0000 0010 from binary to hexadecimal **3402**

h. (2 pts) Convert 010 001 000 from base 2 to base 8 **210**

i. (2 pts) Convert 011 101 111 from binary to octal **357**

3

Exam #202 Page: 3 Name: _____

202

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -30, what is this number's binary representation in 8-bit two's complement?

11100010

- c. (3 pts)

Given that 11001011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-53

- d. (3 pts)

Given that 11101101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-19

4

Exam #202 Page: 4 Name: _____

202

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit guava cherry
```

a. (3 pts) What is the value of `argc` in this case? **3**

b. (3 pts) What is the value of `argv[2][4]`? **r**

c. (3 pts) What is the value of `argv[1][4]`? **a**

d. (3 pts) What is the value of `argv[0][5]`? **I**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double s;  
    Node t;  
    int w;  
    char x;  
    double *y;  
    Node *z;  
    int *a;  
    char *b;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `y` **double ***
- b. (3 pts) `z->data` **int**
- c. (3 pts) `w` **int**
- d. (3 pts) `argv[1][2]` **char**
- e. (3 pts) `argc` **int**
- f. (3 pts) `z->next->next` **Node ***
- g. (3 pts) `&a` **int ****
- h. (3 pts) `z->next` **Node ***
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `&s` **double ***
- k. (3 pts) `*y` **double**

6

Exam #202 Page: 6 Name: _____

202

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

7

Exam #202 Page: 7 Name: _____

202

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #203 Page: 1 Name: _____

203

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #203 Page: 2 Name: _____

203

1. a. (2 pts) Convert 0111 1111 from binary to decimal **127**

- b. (2 pts) Convert dc09 from hexadecimal to base 2 **1101 1100 0000 1001**

- c. (2 pts) Convert 110 100 010 from binary to octal **642**

- d. (2 pts) Convert 117 from decimal to base 2 **0111 0101**

- e. (2 pts) Convert 0110 1010 0010 0001 from base 2 to base 16 **6a21**

- f. (2 pts) Convert 3 from octal to base 2 **011**

- g. (2 pts) Convert 1101 1101 1111 0011 from base 2 to hexadecimal **ddf3**

- h. (2 pts) Convert cc17 from base 16 to base 2 **1100 1100 0001 0111**

- i. (2 pts) Convert 010 000 011 from base 2 to base 8 **203**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -64, what is this number's binary representation in 8-bit two's complement?

11000000

c. (3 pts)

Given that 10110110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-74

d. (3 pts)

Given that 10000111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-121

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit cherry lime apple mango
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][5]`? y

c. (3 pts) What is the value of `argv[2][1]`? i

d. (3 pts) What is the value of `argv[0][1]`? /

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int z;  
    double a;  
    Node b;  
    char c;  
    int *d;  
    double *e;  
    Node *f;  
    char *g;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) f->data **int**
- b. (3 pts) &c **char ***
- c. (3 pts) f->next->next **Node ***
- d. (3 pts) z **int**
- e. (3 pts) *e **double**
- f. (3 pts) argc **int**
- g. (3 pts) f->next **Node ***
- h. (3 pts) &d **int ****
- i. (3 pts) argv[0] **char ***
- j. (3 pts) g **char ***
- k. (3 pts) argv[1][2] **char**

6

Exam #203 Page: 6 Name: _____

203

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #204 Page: 1 Name: _____

204

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #204 Page: 2 Name: _____

204

1. a. (2 pts) Convert 1101 0111 from base 2 to decimal **215**

- b. (2 pts) Convert 011 111 101 from binary to octal **375**

- c. (2 pts) Convert 0111 0110 from base 2 to base 10 **118**

- d. (2 pts) Convert 001 100 010 from base 2 to base 8 **142**

- e. (2 pts) Convert 1101 1010 from base 2 to decimal **218**

- f. (2 pts) Convert 1111 from binary to base 10 **15**

- g. (2 pts) Convert 0010 1111 from binary to base 10 **47**

- h. (2 pts) Convert 7289 from base 16 to binary **0111 0010 1000 1001**

- i. (2 pts) Convert 001 101 000 from binary to octal **150**

3

Exam #204 Page: 3 Name: _____

204

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -108, what is this number's binary representation in 8-bit two's complement?

10010100

- c. (3 pts)

Given that 11001001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-55

- d. (3 pts)

Given that 10000110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-122

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lemon apple kiwi
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][1]`? e

c. (3 pts) What is the value of `argv[0][4]`? n

d. (3 pts) What is the value of `argv[2][0]`? a

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node q;  
    int r;  
    double s;  
    char t;  
    Node *w;  
    int *x;  
    double *y;  
    char *z;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&z` `char **`
- b. (3 pts) `argc` `int`
- c. (3 pts) `*x` `int`
- d. (3 pts) `w->next->next` `Node *`
- e. (3 pts) `argv[0]` `char *`
- f. (3 pts) `w->data` `int`
- g. (3 pts) `t` `char`
- h. (3 pts) `&t` `char *`
- i. (3 pts) `w->next` `Node *`
- j. (3 pts) `argv[1][2]` `char`
- k. (3 pts) `x` `int *`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #205 Page: 1 Name: _____

205

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #205 Page: 2 Name: _____

205

1. a. (2 pts) Convert 0001 1100 0101 1101 from binary to hexadecimal **1c5d**

- b. (2 pts) Convert 38fd from hexadecimal to base 2 **0011 1000 1111 1101**

- c. (2 pts) Convert 31 from base 8 to base 2 **011 001**

- d. (2 pts) Convert 1011 1101 0100 0111 from base 2 to hexadecimal **bd47**

- e. (2 pts) Convert 189 from base 10 to base 2 **1011 1101**

- f. (2 pts) Convert 169 from decimal to base 2 **1010 1001**

- g. (2 pts) Convert 1101 1001 from base 2 to decimal **217**

- h. (2 pts) Convert 111 110 100 from binary to octal **764**

- i. (2 pts) Convert 77 from octal to base 2 **111 111**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -15, what is this number's binary representation in 8-bit two's complement?

11110001

c. (3 pts)

Given that 10110011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-77

d. (3 pts)

Given that 10011111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-97

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date lemon lime
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][6]`? t

c. (3 pts) What is the value of `argv[2][3]`? o

d. (3 pts) What is the value of `argv[1][1]`? a

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int g;  
    Node h;  
    double p;  
    char q;  
    int *r;  
    Node *s;  
    double *t;  
    char *w;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argv[0]` **char ***
- b. (3 pts) `p` **double**
- c. (3 pts) `s->next->next` **Node ***
- d. (3 pts) `r` **int ***
- e. (3 pts) `&s` **Node ****
- f. (3 pts) `&p` **double ***
- g. (3 pts) `argc` **int**
- h. (3 pts) `s->next` **Node ***
- i. (3 pts) `argv[1][2]` **char**
- j. (3 pts) `s->data` **int**
- k. (3 pts) `*t` **double**

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #206 Page: 1 Name: _____

206

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #206 Page: 2 Name: _____

206

1. a. (2 pts) Convert 77 from base 10 to base 2 **0100 1101**

- b. (2 pts) Convert 10 from base 10 to binary **1010**

- c. (2 pts) Convert 154 from decimal to binary **1001 1010**

- d. (2 pts) Convert 0001 1001 from base 2 to base 10 **25**

- e. (2 pts) Convert 1403 from base 16 to base 2 **0001 0100 0000 0011**

- f. (2 pts) Convert db84 from base 16 to binary **1101 1011 1000 0100**

- g. (2 pts) Convert 1101 1011 from binary to base 10 **219**

- h. (2 pts) Convert 011 000 110 from base 2 to base 8 **306**

- i. (2 pts) Convert 1001 1111 from binary to decimal **159**

3

Exam #206 Page: 3 Name: _____

206

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -39, what is this number's binary representation in 8-bit two's complement?

11011001

- c. (3 pts)

Given that 11110101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-11

- d. (3 pts)

Given that 11010011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-45

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit mango date
```

a. (3 pts) What is the value of `argc` in this case? **3**

b. (3 pts) What is the value of `argv[2][1]`? **a**

c. (3 pts) What is the value of `argv[0][5]`? **I**

d. (3 pts) What is the value of `argv[1][4]`? **o**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double h;  
    Node p;  
    int q;  
    char r;  
    double *s;  
    Node *t;  
    int *w;  
    char *x;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) t->data **int**
- b. (3 pts) &x **char ****
- c. (3 pts) t->next->next **Node ***
- d. (3 pts) argc **int**
- e. (3 pts) t->next **Node ***
- f. (3 pts) r **char**
- g. (3 pts) argv[1][2] **char**
- h. (3 pts) s **double ***
- i. (3 pts) &p **Node ***
- j. (3 pts) *w **int**
- k. (3 pts) argv[0] **char ***

6

Exam #206 Page: 6 Name: _____

206

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #207 Page: 1 Name: _____

207

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #207 Page: 2 Name: _____

207

1. a. (2 pts) Convert 44 from base 8 to base 2 **100 100**

- b. (2 pts) Convert 61 from base 8 to base 2 **110 001**

- c. (2 pts) Convert 1000 1000 from binary to decimal **136**

- d. (2 pts) Convert 51 from base 8 to binary **101 001**

- e. (2 pts) Convert 75 from base 8 to binary **111 101**

- f. (2 pts) Convert 117 from decimal to base 2 **0111 0101**

- g. (2 pts) Convert 1000 0101 from base 2 to base 10 **133**

- h. (2 pts) Convert ead9 from base 16 to base 2 **1110 1010 1101 1001**

- i. (2 pts) Convert 011 010 011 from base 2 to base 8 **323**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -74, what is this number's binary representation in 8-bit two's complement?

10110110

c. (3 pts)

Given that 11011111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-33

d. (3 pts)

Given that 11101101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-19

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit date lime kiwi cherry
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][1]`? i

c. (3 pts) What is the value of `argv[1][2]`? t

d. (3 pts) What is the value of `argv[0][4]`? n

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double e;  
    int f;  
    Node g;  
    char h;  
    double *p;  
    int *q;  
    Node *r;  
    char *s;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) &q **int ****
- b. (3 pts) &e **double ***
- c. (3 pts) *s **char**
- d. (3 pts) r->next->next **Node ***
- e. (3 pts) f **int**
- f. (3 pts) r->next **Node ***
- g. (3 pts) argv[1][2] **char**
- h. (3 pts) argc **int**
- i. (3 pts) s **char ***
- j. (3 pts) argv[0] **char ***
- k. (3 pts) r->data **int**

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #208 Page: 1 Name: _____

208

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #208 Page: 2 Name: _____

208

-
1. a. (2 pts) Convert 0100 1111 0001 0001 from binary to hexadecimal **4f11**

- b. (2 pts) Convert 10 from octal to binary **001 000**

- c. (2 pts) Convert 34 from octal to base 2 **011 100**

- d. (2 pts) Convert 227 from decimal to base 2 **1110 0011**

- e. (2 pts) Convert 1111 1110 0101 1011 from base 2 to base 16 **fe5b**

- f. (2 pts) Convert 0011 1011 from binary to base 10 **59**

- g. (2 pts) Convert 1001 1001 from base 2 to decimal **153**

- h. (2 pts) Convert 100 110 110 from base 2 to octal **466**

- i. (2 pts) Convert 25 from base 10 to base 2 **0001 1001**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -40, what is this number's binary representation in 8-bit two's complement?

11011000

c. (3 pts)

Given that 10101001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-87

d. (3 pts)

Given that 10111101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-67

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit guava mango date
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[2][2]`? n

c. (3 pts) What is the value of `argv[1][4]`? a

d. (3 pts) What is the value of `argv[0][3]`? u

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int s;  
    Node t;  
    double w;  
    char x;  
    int *y;  
    Node *z;  
    double *a;  
    char *b;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&b` `char **`
- b. (3 pts) `argc` `int`
- c. (3 pts) `z->next` `Node *`
- d. (3 pts) `y` `int *`
- e. (3 pts) `*b` `char`
- f. (3 pts) `argv[1][2]` `char`
- g. (3 pts) `&x` `char *`
- h. (3 pts) `z->data` `int`
- i. (3 pts) `z->next->next` `Node *`
- j. (3 pts) `s` `int`
- k. (3 pts) `argv[0]` `char *`

6

Exam #208 Page: 6 Name: _____

208

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #209 Page: 1 Name: _____

209

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #209 Page: 2 Name: _____

209

1. a. (2 pts) Convert 1001 0011 from binary to base 10 **147**

b. (2 pts) Convert 110 111 001 from binary to base 8 **671**

c. (2 pts) Convert 27 from base 8 to base 2 **010 111**

d. (2 pts) Convert 0110 1111 from binary to decimal **111**

e. (2 pts) Convert 70 from base 8 to binary **111 000**

f. (2 pts) Convert 110 101 001 from binary to octal **651**

g. (2 pts) Convert 67 from decimal to binary **0100 0011**

h. (2 pts) Convert 001 000 101 from base 2 to octal **105**

i. (2 pts) Convert 227 from decimal to binary **1110 0011**

3

Exam #209 Page: 3 Name: _____

209

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -74, what is this number's binary representation in 8-bit two's complement?

10110110

- c. (3 pts)

Given that 10010011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-109

- d. (3 pts)

Given that 11010111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-41

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit kiwi fig
```

a. (3 pts) What is the value of `argc` in this case? **3**

b. (3 pts) What is the value of `argv[2][0]`? **f**

c. (3 pts) What is the value of `argv[0][5]`? **I**

d. (3 pts) What is the value of `argv[1][2]`? **w**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node h;  
    double p;  
    int q;  
    char r;  
    Node *s;  
    double *t;  
    int *w;  
    char *x;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `x` `char *`
- b. (3 pts) `argv[1][2]` `char`
- c. (3 pts) `&t` `double **`
- d. (3 pts) `argv[0]` `char *`
- e. (3 pts) `s->next->next` `Node *`
- f. (3 pts) `q` `int`
- g. (3 pts) `s->next` `Node *`
- h. (3 pts) `argc` `int`
- i. (3 pts) `*t` `double`
- j. (3 pts) `&r` `char *`
- k. (3 pts) `s->data` `int`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #210 Page: 1 Name: _____

210

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #210 Page: 2 Name: _____

210

1. a. (2 pts) Convert 110 001 010 from base 2 to octal **612**

- b. (2 pts) Convert 53 from base 8 to binary **101 011**

- c. (2 pts) Convert 9633 from base 16 to base 2 **1001 0110 0011 0011**

- d. (2 pts) Convert 203 from decimal to base 2 **1100 1011**

- e. (2 pts) Convert 001 101 111 from base 2 to octal **157**

- f. (2 pts) Convert 75a from base 16 to base 2 **0111 0101 1010**

- g. (2 pts) Convert 0100 0101 from base 2 to base 10 **69**

- h. (2 pts) Convert 1000 1011 from binary to decimal **139**

- i. (2 pts) Convert 133 from base 10 to base 2 **1000 0101**

3

Exam #210 Page: 3 Name: _____

210

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -99, what is this number's binary representation in 8-bit two's complement?

10011101

- c. (3 pts)

Given that 11010100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-44

- d. (3 pts)

Given that 10001010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-118

4

Exam #210 Page: 4 Name: _____

210

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit banana fig
```

a. (3 pts) What is the value of `argc` in this case? **3**

b. (3 pts) What is the value of `argv[2][1]`? **i**

c. (3 pts) What is the value of `argv[1][0]`? **b**

d. (3 pts) What is the value of `argv[0][2]`? **r**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double a;  
    Node b;  
    int c;  
    char d;  
    double *e;  
    Node *f;  
    int *g;  
    char *h;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&g` **int ****
- b. (3 pts) `f->next` **Node ***
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `argc` **int**
- e. (3 pts) `g` **int ***
- f. (3 pts) `*e` **double**
- g. (3 pts) `&b` **Node ***
- h. (3 pts) `argv[1][2]` **char**
- i. (3 pts) `d` **char**
- j. (3 pts) `f->data` **int**
- k. (3 pts) `f->next->next` **Node ***

6

Exam #210 Page: 6 Name: _____

210

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #211 Page: 1 Name: _____

211

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #211 Page: 2 Name: _____

211

1. a. (2 pts) Convert 10 from decimal to binary **1010**

- b. (2 pts) Convert 104 from base 10 to base 2 **0110 1000**

- c. (2 pts) Convert 1000 0100 from binary to base 10 **132**

- d. (2 pts) Convert 87 from decimal to binary **0101 0111**

- e. (2 pts) Convert 27 from base 10 to binary **0001 1011**

- f. (2 pts) Convert 161 from base 10 to base 2 **1010 0001**

- g. (2 pts) Convert 1110 1111 from binary to base 10 **239**

- h. (2 pts) Convert 138f from hexadecimal to binary **0001 0011 1000 1111**

- i. (2 pts) Convert 4f6c from hexadecimal to binary **0100 1111 0110 1100**

3

Exam #211 Page: 3 Name: _____

211

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -6, what is this number's binary representation in 8-bit two's complement?

1111010

- c. (3 pts)

Given that 10111111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-65

- d. (3 pts)

Given that 10100100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-92

4

Exam #211 Page: 4 Name: _____

211

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit guava fig kiwi cherry
```

a. (3 pts) What is the value of `argc` in this case? **5**

b. (3 pts) What is the value of `argv[1][0]`? **g**

c. (3 pts) What is the value of `argv[2][1]`? **i**

d. (3 pts) What is the value of `argv[0][6]`? **t**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int x;  
    double y;  
    Node z;  
    char a;  
    int *b;  
    double *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `d->next->next` **Node ***
- b. (3 pts) `&x` **int ***
- c. (3 pts) `y` **double**
- d. (3 pts) `c` **double ***
- e. (3 pts) `argv[0]` **char ***
- f. (3 pts) `argc` **int**
- g. (3 pts) `d->data` **int**
- h. (3 pts) `*d` **Node**
- i. (3 pts) `argv[1][2]` **char**
- j. (3 pts) `d->next` **Node ***
- k. (3 pts) `&b` **int ****

6

Exam #211 Page: 6 Name: _____

211

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #212 Page: 1 Name: _____

212

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #212 Page: 2 Name: _____

212

1. a. (2 pts) Convert 98 from base 10 to binary **0110 0010**

b. (2 pts) Convert 1011 from base 2 to decimal **11**

c. (2 pts) Convert 41 from base 10 to binary **0010 1001**

d. (2 pts) Convert 100 110 from base 2 to octal **46**

e. (2 pts) Convert 1000 1011 1000 1001 from base 2 to hexadecimal **8b89**

f. (2 pts) Convert 1010 0010 1000 1000 from binary to base 16 **a288**

g. (2 pts) Convert 65 from base 10 to base 2 **0100 0001**

h. (2 pts) Convert 101 110 100 from binary to base 8 **564**

i. (2 pts) Convert 0100 0001 from base 2 to base 10 **65**

3

Exam #212 Page: 3 Name: _____

212

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -50, what is this number's binary representation in 8-bit two's complement?

11001110

- c. (3 pts)

Given that 11010010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-46

- d. (3 pts)

Given that 10100011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-93

4

Exam #212 Page: 4 Name: _____

212

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit cherry banana lemon
```

a. (3 pts) What is the value of `argc` in this case? **4**

b. (3 pts) What is the value of `argv[1][2]`? **e**

c. (3 pts) What is the value of `argv[0][6]`? **t**

d. (3 pts) What is the value of `argv[2][4]`? **n**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node h;  
    int p;  
    double q;  
    char r;  
    Node *s;  
    int *t;  
    double *w;  
    char *x;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) s **Node ***
- b. (3 pts) argv[1][2] **char**
- c. (3 pts) s->data **int**
- d. (3 pts) argc **int**
- e. (3 pts) p **int**
- f. (3 pts) s->next->next **Node ***
- g. (3 pts) argv[0] **char ***
- h. (3 pts) &x **char ****
- i. (3 pts) &h **Node ***
- j. (3 pts) *t **int**
- k. (3 pts) s->next **Node ***

6

Exam #212 Page: 6 Name: _____

212

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #213 Page: 1 Name: _____

213

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #213 Page: 2 Name: _____

213

1. a. (2 pts) Convert 51 from octal to binary **101 001**

- b. (2 pts) Convert 197 from decimal to base 2 **1100 0101**

- c. (2 pts) Convert 101 110 from binary to octal **56**

- d. (2 pts) Convert 1001 1111 0101 0011 from base 2 to base 16 **9f53**

- e. (2 pts) Convert 011 011 101 from base 2 to octal **335**

- f. (2 pts) Convert 17 from octal to base 2 **001 111**

- g. (2 pts) Convert 1110 1011 0000 0111 from binary to base 16 **eb07**

- h. (2 pts) Convert 0100 0001 from binary to base 10 **65**

- i. (2 pts) Convert 11 from decimal to binary **1011**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -84, what is this number's binary representation in 8-bit two's complement?

10101100

c. (3 pts)

Given that 10111100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-68

d. (3 pts)

Given that 10010001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-111

4

Exam #213 Page: 4 Name: _____

213

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit fig grape
```

a. (3 pts) What is the value of `argc` in this case? **3**

b. (3 pts) What is the value of `argv[0][5]`? **I**

c. (3 pts) What is the value of `argv[2][0]`? **g**

d. (3 pts) What is the value of `argv[1][2]`? **g**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node e;  
    double f;  
    int g;  
    char h;  
    Node *p;  
    double *q;  
    int *r;  
    char *s;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) p->next **Node ***
- b. (3 pts) &h **char ***
- c. (3 pts) p->data **int**
- d. (3 pts) argc **int**
- e. (3 pts) p->next->next **Node ***
- f. (3 pts) argv[0] **char ***
- g. (3 pts) &q **double ****
- h. (3 pts) argv[1][2] **char**
- i. (3 pts) s **char ***
- j. (3 pts) *s **char**
- k. (3 pts) h **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #214 Page: 1 Name: _____

214

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #214 Page: 2 Name: _____

214

1. a. (2 pts) Convert d8d0 from base 16 to base 2 **1101 1000 1101 0000**

b. (2 pts) Convert 96b0 from hexadecimal to binary **1001 0110 1011 0000**

c. (2 pts) Convert 0100 1101 1011 0101 from binary to base 16 **4db5**

d. (2 pts) Convert 1111 1011 from base 2 to base 10 **251**

e. (2 pts) Convert 1100 0100 from base 2 to decimal **196**

f. (2 pts) Convert 33 from base 8 to binary **011 011**

g. (2 pts) Convert 237 from base 10 to binary **1110 1101**

h. (2 pts) Convert aaa0 from base 16 to binary **1010 1010 1010 0000**

i. (2 pts) Convert 1010 1101 from base 2 to base 10 **173**

3

Exam #214 Page: 3 Name: _____

214

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -109, what is this number's binary representation in 8-bit two's complement?

10010011

- c. (3 pts)

Given that 11111101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-3

- d. (3 pts)

Given that 11110000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-16

4

Exam #214 Page: 4 Name: _____

214

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit date lime guava kiwi
```

a. (3 pts) What is the value of `argc` in this case? **5**

b. (3 pts) What is the value of `argv[0][5]`? **I**

c. (3 pts) What is the value of `argv[2][1]`? **i**

d. (3 pts) What is the value of `argv[1][2]`? **t**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double e;  
    int f;  
    Node g;  
    char h;  
    double *p;  
    int *q;  
    Node *r;  
    char *s;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argc` **int**
- b. (3 pts) `e` **double**
- c. (3 pts) `r->next` **Node ***
- d. (3 pts) `*r` **Node**
- e. (3 pts) `argv[1][2]` **char**
- f. (3 pts) `argv[0]` **char ***
- g. (3 pts) `r->data` **int**
- h. (3 pts) `&g` **Node ***
- i. (3 pts) `&s` **char ****
- j. (3 pts) `r->next->next` **Node ***
- k. (3 pts) `r` **Node ***

6

Exam #214 Page: 6 Name: _____

214

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #215 Page: 1 Name: _____

215

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #215 Page: 2 Name: _____

215

1. a. (2 pts) Convert 111 011 from base 2 to octal **73**

- b. (2 pts) Convert 0101 0000 1101 1001 from base 2 to base 16 **50d9**

- c. (2 pts) Convert 0011 1011 1001 0101 from binary to hexadecimal **3b95**

- d. (2 pts) Convert 41 from base 8 to binary **100 001**

- e. (2 pts) Convert a83e from base 16 to base 2 **1010 1000 0011 1110**

- f. (2 pts) Convert 1000 1010 1011 from base 2 to base 16 **8ab**

- g. (2 pts) Convert 151 from decimal to binary **1001 0111**

- h. (2 pts) Convert 0011 0010 0101 0000 from binary to hexadecimal **3250**

- i. (2 pts) Convert 119 from decimal to binary **0111 0111**

3

Exam #215 Page: 3 Name: _____

215

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -15, what is this number's binary representation in 8-bit two's complement?

11110001

- c. (3 pts)

Given that 11101000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-24

- d. (3 pts)

Given that 10001001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-119

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit mango banana guava apple
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][2]`? n

c. (3 pts) What is the value of `argv[1][0]`? m

d. (3 pts) What is the value of `argv[0][2]`? r

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double r;  
    int s;  
    Node t;  
    char w;  
    double *x;  
    int *y;  
    Node *z;  
    char *a;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `y` `int *`
- b. (3 pts) `z->data` `int`
- c. (3 pts) `argv[1][2]` `char`
- d. (3 pts) `argc` `int`
- e. (3 pts) `t` `Node`
- f. (3 pts) `&s` `int *`
- g. (3 pts) `z->next` `Node *`
- h. (3 pts) `argv[0]` `char *`
- i. (3 pts) `z->next->next` `Node *`
- j. (3 pts) `&y` `int **`
- k. (3 pts) `*x` `double`

6

Exam #215 Page: 6 Name: _____

215

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #216 Page: 1 Name: _____

216

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #216 Page: 2 Name: _____

216

1. a. (2 pts) Convert 0111 1011 00 from base 2 to octal **354**

- b. (2 pts) Convert 243 from base 10 to base 2 **1111 0011**

- c. (2 pts) Convert e0f7 from hexadecimal to base 2 **1110 0000 1111 0111**

- d. (2 pts) Convert 4311 from base 16 to binary **0100 0011 0001 0001**

- e. (2 pts) Convert 6 from base 8 to binary **110**

- f. (2 pts) Convert 9 from base 10 to base 2 **1001**

- g. (2 pts) Convert 1110 1000 1101 1001 from base 2 to hexadecimal **e8d9**

- h. (2 pts) Convert 110 110 001 from binary to base 8 **661**

- i. (2 pts) Convert 106 from base 10 to base 2 **0110 1010**

3

Exam #216 Page: 3 Name: _____

216

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -60, what is this number's binary representation in 8-bit two's complement?

11000100

- c. (3 pts)

Given that 11111011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-5

- d. (3 pts)

Given that 10001001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-119

4

Exam #216 Page: 4 Name: _____

216

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit banana lemon grape
```

a. (3 pts) What is the value of `argc` in this case? **4**

b. (3 pts) What is the value of `argv[1][5]`? **a**

c. (3 pts) What is the value of `argv[0][4]`? **n**

d. (3 pts) What is the value of `argv[2][0]`? **l**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&d` **Node ***
- b. (3 pts) `f` **double**
- c. (3 pts) `*r` **char**
- d. (3 pts) `&h` **Node ****
- e. (3 pts) `h->next->next` **Node ***
- f. (3 pts) `h->next` **Node ***
- g. (3 pts) `h->data` **int**
- h. (3 pts) `argc` **int**
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `r` **char ***

6

Exam #216 Page: 6 Name: _____

216

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

7

Exam #216 Page: 7 Name: _____

216

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #217 Page: 1 Name: _____

217

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #217 Page: 2 Name: _____

217

1. a. (2 pts) Convert baeb from base 16 to binary **1011 1010 1110 1011**

- b. (2 pts) Convert 53 from base 8 to binary **101 011**

- c. (2 pts) Convert ced7 from hexadecimal to base 2 **1100 1110 1101 0111**

- d. (2 pts) Convert cefb from base 16 to base 2 **1100 1110 1111 1011**

- e. (2 pts) Convert fc02 from base 16 to base 2 **1111 1100 0000 0010**

- f. (2 pts) Convert 1010 0011 from binary to base 10 **163**

- g. (2 pts) Convert 44 from octal to binary **100 100**

- h. (2 pts) Convert 6072 from hexadecimal to base 2 **0110 0000 0111 0010**

- i. (2 pts) Convert 0011 0100 from binary to decimal **52**

3

Exam #217 Page: 3 Name: _____

217

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- c. (3 pts)

Given that 11100110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-26

- d. (3 pts)

Given that 11101101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-19

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava lemon
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[0][3]`? u

c. (3 pts) What is the value of `argv[1][1]`? u

d. (3 pts) What is the value of `argv[2][0]`? l

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double a;  
    Node b;  
    int c;  
    char d;  
    double *e;  
    Node *f;  
    int *g;  
    char *h;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `f->next` **Node ***
- b. (3 pts) `argv[1][2]` **char**
- c. (3 pts) `argv[0]` **char ***
- d. (3 pts) `argc` **int**
- e. (3 pts) `g` **int ***
- f. (3 pts) `a` **double**
- g. (3 pts) `&g` **int ****
- h. (3 pts) `&a` **double ***
- i. (3 pts) `f->next->next` **Node ***
- j. (3 pts) `f->data` **int**
- k. (3 pts) `*f` **Node**

6

Exam #217 Page: 6 Name: _____

217

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #218 Page: 1 Name: _____

218

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #218 Page: 2 Name: _____

218

1. a. (2 pts) Convert 1110 1100 from binary to base 10 **236**

b. (2 pts) Convert 0111 1111 0101 0011 from binary to base 16 **7f53**

c. (2 pts) Convert 001 010 from binary to octal **12**

d. (2 pts) Convert 0010 1011 0011 1101 from base 2 to hexadecimal **2b3d**

e. (2 pts) Convert 82 from base 10 to base 2 **0101 0010**

f. (2 pts) Convert 1101 0110 0100 0010 from binary to hexadecimal **d642**

g. (2 pts) Convert 1001 0100 1111 1000 from base 2 to base 16 **94f8**

h. (2 pts) Convert c961 from base 16 to base 2 **1100 1001 0110 0001**

i. (2 pts) Convert d60e from base 16 to base 2 **1101 0110 0000 1110**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -119, what is this number's binary representation in 8-bit two's complement?

10001001

c. (3 pts)

Given that 10100111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-89

d. (3 pts)

Given that 11010110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-42

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit cherry grape lemon fig
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][2]`? a

c. (3 pts) What is the value of `argv[0][5]`? I

d. (3 pts) What is the value of `argv[1][0]`? c

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int b;  
    double c;  
    Node d;  
    char e;  
    int *f;  
    double *g;  
    Node *h;  
    char *p;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `g` `double *`
- b. (3 pts) `&d` `Node *`
- c. (3 pts) `c` `double`
- d. (3 pts) `argc` `int`
- e. (3 pts) `h->next->next` `Node *`
- f. (3 pts) `argv[1][2]` `char`
- g. (3 pts) `h->data` `int`
- h. (3 pts) `*f` `int`
- i. (3 pts) `h->next` `Node *`
- j. (3 pts) `argv[0]` `char *`
- k. (3 pts) `&f` `int **`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #219 Page: 1 Name: _____

219

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #219 Page: 2 Name: _____

219

1. a. (2 pts) Convert 0011 0001 0010 1100 from binary to base 16 **312c**

- b. (2 pts) Convert 0011 1001 from base 2 to decimal **57**

- c. (2 pts) Convert 74 from base 8 to binary **111 100**

- d. (2 pts) Convert 55 from base 8 to base 2 **101 101**

- e. (2 pts) Convert 356b from base 16 to base 2 **0011 0101 0110 1011**

- f. (2 pts) Convert 701e from base 16 to binary **0111 0000 0001 1110**

- g. (2 pts) Convert 0011 1110 1110 1001 from binary to hexadecimal **3ee9**

- h. (2 pts) Convert 0101 0001 0001 0010 from base 2 to base 16 **5112**

- i. (2 pts) Convert 1010 0000 from binary to decimal **160**

3

Exam #219 Page: 3 Name: _____

219

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -25, what is this number's binary representation in 8-bit two's complement?

11100111

- c. (3 pts)

Given that 10010001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-111

- d. (3 pts)

Given that 11101111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-17

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit mango grape lemon date
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][3]`? p

c. (3 pts) What is the value of `argv[1][3]`? g

d. (3 pts) What is the value of `argv[0][6]`? t

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int x;  
    double y;  
    Node z;  
    char a;  
    int *b;  
    double *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&y` **double ***
- b. (3 pts) `x` **int**
- c. (3 pts) `d->next->next` **Node ***
- d. (3 pts) `argc` **int**
- e. (3 pts) `argv[0]` **char ***
- f. (3 pts) `*d` **Node**
- g. (3 pts) `&d` **Node ****
- h. (3 pts) `b` **int ***
- i. (3 pts) `d->data` **int**
- j. (3 pts) `argv[1][2]` **char**
- k. (3 pts) `d->next` **Node ***

6

Exam #219 Page: 6 Name: _____

219

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #220 Page: 1 Name: _____

220

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #220 Page: 2 Name: _____

220

1. a. (2 pts) Convert 1000 1001 1000 1001 from binary to base 16 **8989**

b. (2 pts) Convert 1101 1100 0100 1000 from base 2 to base 16 **dc48**

c. (2 pts) Convert 100 110 000 from binary to base 8 **460**

d. (2 pts) Convert 114 from base 10 to base 2 **0111 0010**

e. (2 pts) Convert 165 from base 10 to base 2 **1010 0101**

f. (2 pts) Convert 34 from base 8 to binary **011 100**

g. (2 pts) Convert 1001 0000 from base 2 to base 10 **144**

h. (2 pts) Convert 1111 0111 from base 2 to base 10 **247**

i. (2 pts) Convert 1001 0010 from base 2 to decimal **146**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -69, what is this number's binary representation in 8-bit two's complement?

10111011

c. (3 pts)

Given that 10100100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-92

d. (3 pts)

Given that 11101111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-17

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit date apple mango
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][0]`? d

c. (3 pts) What is the value of `argv[2][4]`? e

d. (3 pts) What is the value of `argv[0][4]`? n

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int h;  
    Node p;  
    double q;  
    char r;  
    int *s;  
    Node *t;  
    double *w;  
    char *x;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) t->next->next **Node ***
- b. (3 pts) argv[1][2] **char**
- c. (3 pts) &t **Node ****
- d. (3 pts) argv[0] **char ***
- e. (3 pts) *t **Node**
- f. (3 pts) t->data **int**
- g. (3 pts) argc **int**
- h. (3 pts) t->next **Node ***
- i. (3 pts) r **char**
- j. (3 pts) x **char ***
- k. (3 pts) &p **Node ***

6

Exam #220 Page: 6 Name: _____

220

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #221 Page: 1 Name: _____

221

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #221 Page: 2 Name: _____

221

1. a. (2 pts) Convert 110 011 100 from binary to octal **634**

b. (2 pts) Convert 1001 0110 from base 2 to decimal **150**

c. (2 pts) Convert 1000 0110 0101 1001 from base 2 to hexadecimal **8659**

d. (2 pts) Convert 1111 1110 from binary to decimal **254**

e. (2 pts) Convert 1000 1001 0010 1111 from base 2 to base 16 **892f**

f. (2 pts) Convert 010 110 from base 2 to base 8 **26**

g. (2 pts) Convert 0011 1010 from binary to base 10 **58**

h. (2 pts) Convert 7f34 from hexadecimal to base 2 **0111 1111 0011 0100**

i. (2 pts) Convert 010 111 001 from base 2 to base 8 **271**

3

Exam #221 Page: 3 Name: _____

221

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -104, what is this number's binary representation in 8-bit two's complement?

10011000

- c. (3 pts)

Given that 10001111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-113

- d. (3 pts)

Given that 10001000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-120

4

Exam #221 Page: 4 Name: _____

221

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape banana
```

a. (3 pts) What is the value of `argc` in this case? **3**

b. (3 pts) What is the value of `argv[0][2]`? **r**

c. (3 pts) What is the value of `argv[2][2]`? **n**

d. (3 pts) What is the value of `argv[1][3]`? **p**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node w;  
    double x;  
    int y;  
    char z;  
    Node *a;  
    double *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) argc **int**
- b. (3 pts) a->next **Node ***
- c. (3 pts) a->data **int**
- d. (3 pts) c **int ***
- e. (3 pts) a->next->next **Node ***
- f. (3 pts) argv[1][2] **char**
- g. (3 pts) &w **Node ***
- h. (3 pts) *d **char**
- i. (3 pts) x **double**
- j. (3 pts) &d **char ****
- k. (3 pts) argv[0] **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #222 Page: 1 Name: _____

222

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #222 Page: 2 Name: _____

222

1. a. (2 pts) Convert 77 from base 8 to binary **111 111**

- b. (2 pts) Convert 011 001 111 from binary to base 8 **317**

- c. (2 pts) Convert 1011 1100 from base 2 to base 10 **188**

- d. (2 pts) Convert 90 from base 10 to binary **0101 1010**

- e. (2 pts) Convert 67 from base 8 to base 2 **110 111**

- f. (2 pts) Convert 001 111 011 from binary to octal **173**

- g. (2 pts) Convert 17 from base 8 to binary **001 111**

- h. (2 pts) Convert e823 from hexadecimal to base 2 **1110 1000 0010 0011**

- i. (2 pts) Convert 254 from base 10 to binary **1111 1110**

3

Exam #222 Page: 3 Name: _____

222

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -128, what is this number's binary representation in 8-bit two's complement?

10000000

- c. (3 pts)

Given that 11010000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-48

- d. (3 pts)

Given that 10111100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-68

4

Exam #222 Page: 4 Name: _____

222

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit fig mango kiwi grape
```

a. (3 pts) What is the value of `argc` in this case? **5**

b. (3 pts) What is the value of `argv[1][0]`? **f**

c. (3 pts) What is the value of `argv[2][4]`? **o**

d. (3 pts) What is the value of `argv[0][2]`? **r**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double x;  
    int y;  
    Node z;  
    char a;  
    double *b;  
    int *c;  
    Node *d;  
    char *e;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) &a **char ***
- b. (3 pts) d->next **Node ***
- c. (3 pts) c **int ***
- d. (3 pts) argv[0] **char ***
- e. (3 pts) *d **Node**
- f. (3 pts) d->data **int**
- g. (3 pts) argc **int**
- h. (3 pts) d->next->next **Node ***
- i. (3 pts) &b **double ****
- j. (3 pts) z **Node**
- k. (3 pts) argv[1][2] **char**

6

Exam #222 Page: 6 Name: _____

222

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

7

Exam #222 Page: 7 Name: _____

222

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #223 Page: 1 Name: _____

223

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #223 Page: 2 Name: _____

223

1. a. (2 pts) Convert 0100 0100 from binary to base 10 **68**

b. (2 pts) Convert 221f from hexadecimal to binary **0010 0010 0001 1111**

c. (2 pts) Convert 1010 1010 from binary to base 10 **170**

d. (2 pts) Convert 230 from base 10 to binary **1110 0110**

e. (2 pts) Convert 110 000 101 from base 2 to octal **605**

f. (2 pts) Convert 110 101 111 from base 2 to octal **657**

g. (2 pts) Convert 71 from octal to base 2 **111 001**

h. (2 pts) Convert 011 011 111 from base 2 to base 8 **337**

i. (2 pts) Convert 1100 1000 from binary to base 10 **200**

3

Exam #223 Page: 3 Name: _____

223

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -35, what is this number's binary representation in 8-bit two's complement?

11011101

- c. (3 pts)

Given that 10111010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-70

- d. (3 pts)

Given that 11010101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-43

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit kiwi lime mango lemon
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][0]`? l

c. (3 pts) What is the value of `argv[1][2]`? w

d. (3 pts) What is the value of `argv[0][0]`? .

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int s;  
    double t;  
    Node w;  
    char x;  
    int *y;  
    double *z;  
    Node *a;  
    char *b;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argv[0]` **char ***
- b. (3 pts) `argv[1][2]` **char**
- c. (3 pts) `argc` **int**
- d. (3 pts) `a->data` **int**
- e. (3 pts) `*y` **int**
- f. (3 pts) `a->next` **Node ***
- g. (3 pts) `&w` **Node ***
- h. (3 pts) `&a` **Node ****
- i. (3 pts) `y` **int ***
- j. (3 pts) `t` **double**
- k. (3 pts) `a->next->next` **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #224 Page: 1 Name: _____

224

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #224 Page: 2 Name: _____

224

-
1. a. (2 pts) Convert 56 from decimal to binary **0011 1000**
- b. (2 pts) Convert 2 from octal to binary **010**
- c. (2 pts) Convert c06 from hexadecimal to base 2 **1100 0000 0110**
- d. (2 pts) Convert 111 111 from base 2 to octal **77**
- e. (2 pts) Convert 100 111 000 from base 2 to base 8 **470**
- f. (2 pts) Convert 20 from base 10 to base 2 **0001 0100**
- g. (2 pts) Convert 0111 0110 0111 0010 from binary to base 16 **7672**
- h. (2 pts) Convert c51 from base 16 to binary **1100 0101 0001**
- i. (2 pts) Convert 77 from octal to base 2 **111 111**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

b. (3 pts)

Given the decimal number -29, what is this number's binary representation in 8-bit two's complement?

11100011

c. (3 pts)

Given that 10010111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-105

d. (3 pts)

Given that 10000011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-125

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit fig lemon cherry
```

a. (3 pts) What is the value of `argc` in this case? **4**

b. (3 pts) What is the value of `argv[1][2]`? **g**

c. (3 pts) What is the value of `argv[0][6]`? **t**

d. (3 pts) What is the value of `argv[2][4]`? **n**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int z;  
    Node a;  
    double b;  
    char c;  
    int *d;  
    Node *e;  
    double *f;  
    char *g;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&g` `char **`
- b. (3 pts) `a` `Node`
- c. (3 pts) `e->next->next` `Node *`
- d. (3 pts) `*g` `char`
- e. (3 pts) `argv[1][2]` `char`
- f. (3 pts) `e->data` `int`
- g. (3 pts) `e->next` `Node *`
- h. (3 pts) `argv[0]` `char *`
- i. (3 pts) `&b` `double *`
- j. (3 pts) `argc` `int`
- k. (3 pts) `d` `int *`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #225 Page: 1 Name: _____

225

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #225 Page: 2 Name: _____

225

1. a. (2 pts) Convert 37 from octal to binary **011 111**

b. (2 pts) Convert 110 001 000 from base 2 to base 8 **610**

c. (2 pts) Convert f9e6 from base 16 to base 2 **1111 1001 1110 0110**

d. (2 pts) Convert 1010 1011 from base 2 to base 10 **171**

e. (2 pts) Convert 127 from base 10 to binary **0111 1111**

f. (2 pts) Convert 1010 1110 from base 2 to decimal **174**

g. (2 pts) Convert 0010 0000 0110 0011 from base 2 to base 16 **2063**

h. (2 pts) Convert 9401 from hexadecimal to binary **1001 0100 0000 0001**

i. (2 pts) Convert 1100 0111 1111 0000 from base 2 to hexadecimal **c7f0**

3

Exam #225 Page: 3 Name: _____

225

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -63, what is this number's binary representation in 8-bit two's complement?

11000001

- c. (3 pts)

Given that 10000001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-127

- d. (3 pts)

Given that 10011101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-99

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit cherry kiwi
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[0][3]`? u

c. (3 pts) What is the value of `argv[2][1]`? i

d. (3 pts) What is the value of `argv[1][0]`? c

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node w;  
    double x;  
    int y;  
    char z;  
    Node *a;  
    double *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `*a` **Node**
- b. (3 pts) `argv[0]` **char ***
- c. (3 pts) `a->next` **Node ***
- d. (3 pts) `a->data` **int**
- e. (3 pts) `&x` **double ***
- f. (3 pts) `argc` **int**
- g. (3 pts) `z` **char**
- h. (3 pts) `&b` **double ****
- i. (3 pts) `d` **char ***
- j. (3 pts) `a->next->next` **Node ***
- k. (3 pts) `argv[1][2]` **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #226 Page: 1 Name: _____

226

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #226 Page: 2 Name: _____

226

1. a. (2 pts) Convert af31 from base 16 to base 2 **1010 1111 0011 0001**

b. (2 pts) Convert 1001 0101 from binary to decimal **149**

c. (2 pts) Convert 0011 0000 from base 2 to decimal **48**

d. (2 pts) Convert 7 from decimal to binary **0111**

e. (2 pts) Convert d5e6 from base 16 to binary **1101 0101 1110 0110**

f. (2 pts) Convert 1110 0000 1100 0110 from binary to base 16 **e0c6**

g. (2 pts) Convert 0010 0010 1001 0001 from binary to hexadecimal **2291**

h. (2 pts) Convert 77 from octal to binary **111 111**

i. (2 pts) Convert 32 from octal to base 2 **011 010**

3

Exam #226 Page: 3 Name: _____

226

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -88, what is this number's binary representation in 8-bit two's complement?

10101000

- c. (3 pts)

Given that 11000011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-61

- d. (3 pts)

Given that 11010000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-48

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit cherry guava
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[1][3]`? r

c. (3 pts) What is the value of `argv[2][4]`? a

d. (3 pts) What is the value of `argv[0][0]`? .

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double w;  
    Node x;  
    int y;  
    char z;  
    double *a;  
    Node *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `c` `int *`
- b. (3 pts) `argv[1][2]` `char`
- c. (3 pts) `w` `double`
- d. (3 pts) `&z` `char *`
- e. (3 pts) `b->data` `int`
- f. (3 pts) `b->next->next` `Node *`
- g. (3 pts) `&c` `int **`
- h. (3 pts) `b->next` `Node *`
- i. (3 pts) `argc` `int`
- j. (3 pts) `*a` `double`
- k. (3 pts) `argv[0]` `char *`

6

Exam #226 Page: 6 Name: _____

226

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #227 Page: 1 Name: _____

227

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #227 Page: 2 Name: _____

227

1. a. (2 pts) Convert 111 101 000 from binary to base 8 **750**

- b. (2 pts) Convert 79 from decimal to base 2 **0100 1111**

- c. (2 pts) Convert 7 from base 8 to binary **111**

- d. (2 pts) Convert 147 from base 10 to base 2 **1001 0011**

- e. (2 pts) Convert 56 from base 8 to binary **101 110**

- f. (2 pts) Convert 7aa2 from hexadecimal to base 2 **0111 1010 1010 0010**

- g. (2 pts) Convert 1100 1100 1000 0001 from base 2 to hexadecimal **cc81**

- h. (2 pts) Convert 100 001 001 from base 2 to octal **411**

- i. (2 pts) Convert 0011 0011 1100 0010 from base 2 to hexadecimal **33c2**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

b. (3 pts)

Given the decimal number -123, what is this number's binary representation in 8-bit two's complement?

10000101

c. (3 pts)

Given that 10101101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-83

d. (3 pts)

Given that 11101010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-22

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lemon fig guava grape
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][1]`? i

c. (3 pts) What is the value of `argv[1][4]`? n

d. (3 pts) What is the value of `argv[0][4]`? n

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int r;  
    double s;  
    Node t;  
    char w;  
    int *x;  
    double *y;  
    Node *z;  
    char *a;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `y` **double ***
- b. (3 pts) `&x` **int ****
- c. (3 pts) `z->next->next` **Node ***
- d. (3 pts) `argv[1][2]` **char**
- e. (3 pts) `argc` **int**
- f. (3 pts) `z->next` **Node ***
- g. (3 pts) `w` **char**
- h. (3 pts) `&t` **Node ***
- i. (3 pts) `*y` **double**
- j. (3 pts) `argv[0]` **char ***
- k. (3 pts) `z->data` **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #228 Page: 1 Name: _____

228

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #228 Page: 2 Name: _____

228

1. a. (2 pts) Convert 010 011 000 from base 2 to base 8 **230**

- b. (2 pts) Convert 1111 0010 from binary to decimal **242**

- c. (2 pts) Convert 110 000 111 from base 2 to base 8 **607**

- d. (2 pts) Convert 010 011 110 from base 2 to octal **236**

- e. (2 pts) Convert 0010 1001 from base 2 to base 10 **41**

- f. (2 pts) Convert 36 from base 8 to base 2 **011 110**

- g. (2 pts) Convert 111 100 from base 2 to octal **74**

- h. (2 pts) Convert 2b12 from base 16 to binary **0010 1011 0001 0010**

- i. (2 pts) Convert 0010 0110 0011 0101 from binary to hexadecimal **2635**

3

Exam #228 Page: 3 Name: _____

228

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -39, what is this number's binary representation in 8-bit two's complement?

11011001

- c. (3 pts)

Given that 11000000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-64

- d. (3 pts)

Given that 11101001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-23

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime apple grape
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][0]`? l

c. (3 pts) What is the value of `argv[0][4]`? n

d. (3 pts) What is the value of `argv[2][2]`? p

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `h->next->next` **Node ***
- b. (3 pts) `h->data` **int**
- c. (3 pts) `h->next` **Node ***
- d. (3 pts) `*h` **Node**
- e. (3 pts) `&r` **char ****
- f. (3 pts) `argv[1][2]` **char**
- g. (3 pts) `&f` **double ***
- h. (3 pts) `r` **char ***
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `f` **double**
- k. (3 pts) `argc` **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #229 Page: 1 Name: _____

229

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #229 Page: 2 Name: _____

229

1. a. (2 pts) Convert 145 from decimal to binary **1001 0001**

- b. (2 pts) Convert 1010 1100 1101 0000 from base 2 to base 16 **acd0**

- c. (2 pts) Convert 1011 0001 0110 1000 from base 2 to hexadecimal **b168**

- d. (2 pts) Convert 1101 1011 0011 1100 from base 2 to hexadecimal **db3c**

- e. (2 pts) Convert 3 from base 8 to binary **011**

- f. (2 pts) Convert 0001 0101 1101 0000 from base 2 to base 16 **15d0**

- g. (2 pts) Convert 110 010 000 from binary to octal **620**

- h. (2 pts) Convert 54 from base 8 to base 2 **101 100**

- i. (2 pts) Convert 74 from base 8 to base 2 **111 100**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

b. (3 pts)

Given the decimal number -73, what is this number's binary representation in 8-bit two's complement?

10110111

c. (3 pts)

Given that 10101011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-85

d. (3 pts)

Given that 10000010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-126

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime date
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[0][1]`? /

c. (3 pts) What is the value of `argv[2][2]`? t

d. (3 pts) What is the value of `argv[1][0]`? l

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node q;  
    double r;  
    int s;  
    char t;  
    Node *w;  
    double *x;  
    int *y;  
    char *z;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `y` `int *`
- b. (3 pts) `*y` `int`
- c. (3 pts) `argc` `int`
- d. (3 pts) `q` `Node`
- e. (3 pts) `&x` `double **`
- f. (3 pts) `w->data` `int`
- g. (3 pts) `argv[1][2]` `char`
- h. (3 pts) `w->next->next` `Node *`
- i. (3 pts) `argv[0]` `char *`
- j. (3 pts) `&r` `double *`
- k. (3 pts) `w->next` `Node *`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #230 Page: 1 Name: _____

230

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #230 Page: 2 Name: _____

230

1. a. (2 pts) Convert 1100 0010 from base 2 to decimal **194**

- b. (2 pts) Convert 7e56 from base 16 to binary **0111 1110 0101 0110**

- c. (2 pts) Convert 1110 0111 from base 2 to base 10 **231**

- d. (2 pts) Convert 0011 0111 from base 2 to decimal **55**

- e. (2 pts) Convert 0110 0011 0001 0011 from base 2 to hexadecimal **6313**

- f. (2 pts) Convert 010 010 000 from binary to octal **220**

- g. (2 pts) Convert 1100 1010 0101 0011 from binary to base 16 **ca53**

- h. (2 pts) Convert 110 111 from binary to octal **67**

- i. (2 pts) Convert 1001 0010 0000 0111 from binary to base 16 **9207**

3

Exam #230 Page: 3 Name: _____

230

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -98, what is this number's binary representation in 8-bit two's complement?

10011110

- c. (3 pts)

Given that 11101100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-20

- d. (3 pts)

Given that 10110110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-74

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon cherry apple lime
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][0]`? c

c. (3 pts) What is the value of `argv[0][5]`? I

d. (3 pts) What is the value of `argv[1][0]`? l

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double q;  
    int r;  
    Node s;  
    char t;  
    double *w;  
    int *x;  
    Node *y;  
    char *z;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) argv[0] **char ***
- b. (3 pts) *x **int**
- c. (3 pts) x **int ***
- d. (3 pts) &z **char ****
- e. (3 pts) argc **int**
- f. (3 pts) argv[1][2] **char**
- g. (3 pts) &q **double ***
- h. (3 pts) s **Node**
- i. (3 pts) y->next **Node ***
- j. (3 pts) y->data **int**
- k. (3 pts) y->next->next **Node ***

6

Exam #230 Page: 6 Name: _____

230

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #231 Page: 1 Name: _____

231

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #231 Page: 2 Name: _____

231

1. a. (2 pts) Convert 78d from base 16 to base 2 **0111 1000 1101**

- b. (2 pts) Convert 001 110 000 from binary to base 8 **160**

- c. (2 pts) Convert 110 101 011 from binary to base 8 **653**

- d. (2 pts) Convert 60 from base 8 to base 2 **110 000**

- e. (2 pts) Convert 21 from base 8 to base 2 **010 001**

- f. (2 pts) Convert 70 from octal to base 2 **111 000**

- g. (2 pts) Convert 0111 0100 0100 0100 from base 2 to hexadecimal **7444**

- h. (2 pts) Convert 163 from decimal to binary **1010 0011**

- i. (2 pts) Convert 5c1e from base 16 to base 2 **0101 1100 0001 1110**

3

Exam #231 Page: 3 Name: _____

231

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -4, what is this number's binary representation in 8-bit two's complement?

11111100

- c. (3 pts)

Given that 11010110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-42

- d. (3 pts)

Given that 11010000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-48

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lemon fig lime guava
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][1]`? i

c. (3 pts) What is the value of `argv[1][0]`? l

d. (3 pts) What is the value of `argv[0][1]`? /

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double g;  
    int h;  
    Node p;  
    char q;  
    double *r;  
    int *s;  
    Node *t;  
    char *w;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) argc **int**
- b. (3 pts) t->data **int**
- c. (3 pts) t->next->next **Node ***
- d. (3 pts) &s **int ****
- e. (3 pts) g **double**
- f. (3 pts) r **double ***
- g. (3 pts) argv[0] **char ***
- h. (3 pts) *w **char**
- i. (3 pts) argv[1][2] **char**
- j. (3 pts) t->next **Node ***
- k. (3 pts) &q **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

7

Exam #231 Page: 7 Name: _____

231

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #232 Page: 1 Name: _____

232

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #232 Page: 2 Name: _____

232

1. a. (2 pts) Convert 0101 1111 1110 1010 from base 2 to base 16 **5fea**

b. (2 pts) Convert 219 from decimal to base 2 **1101 1011**

c. (2 pts) Convert 0111 1011 from base 2 to decimal **123**

d. (2 pts) Convert 7efb from base 16 to base 2 **0111 1110 1111 1011**

e. (2 pts) Convert 1011 0110 1101 1000 from base 2 to hexadecimal **b6d8**

f. (2 pts) Convert e366 from hexadecimal to base 2 **1110 0011 0110 0110**

g. (2 pts) Convert 1100 0101 from base 2 to base 10 **197**

h. (2 pts) Convert 49d3 from hexadecimal to base 2 **0100 1001 1101 0011**

i. (2 pts) Convert 4e92 from hexadecimal to binary **0100 1110 1001 0010**

3

Exam #232 Page: 3 Name: _____

232

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -48, what is this number's binary representation in 8-bit two's complement?

11010000

- c. (3 pts)

Given that 11101010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-22

- d. (3 pts)

Given that 11001111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-49

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit fig cherry lemon
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][1]`? i

c. (3 pts) What is the value of `argv[0][0]`? .

d. (3 pts) What is the value of `argv[2][4]`? r

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node z;  
    int a;  
    double b;  
    char c;  
    Node *d;  
    int *e;  
    double *f;  
    char *g;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argc` **int**
- b. (3 pts) `&d` **Node ****
- c. (3 pts) `d->next` **Node ***
- d. (3 pts) `c` **char**
- e. (3 pts) `argv[0]` **char ***
- f. (3 pts) `d->next->next` **Node ***
- g. (3 pts) `g` **char ***
- h. (3 pts) `d->data` **int**
- i. (3 pts) `&c` **char ***
- j. (3 pts) `*f` **double**
- k. (3 pts) `argv[1][2]` **char**

6

Exam #232 Page: 6 Name: _____

232

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #233 Page: 1 Name: _____

233

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #233 Page: 2 Name: _____

233

1. a. (2 pts) Convert 101 001 001 from base 2 to base 8 **511**

- b. (2 pts) Convert 45 from octal to binary **100 101**

- c. (2 pts) Convert 0110 1000 from binary to decimal **104**

- d. (2 pts) Convert ae5 from base 16 to binary **1010 1110 0101**

- e. (2 pts) Convert 46 from base 8 to base 2 **100 110**

- f. (2 pts) Convert 125 from base 10 to base 2 **0111 1101**

- g. (2 pts) Convert 0110 1111 from binary to base 10 **111**

- h. (2 pts) Convert 110 100 011 from base 2 to base 8 **643**

- i. (2 pts) Convert 0001 1000 1010 1001 from base 2 to base 16 **18a9**

3

Exam #233 Page: 3 Name: _____

233

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -83, what is this number's binary representation in 8-bit two's complement?

10101101

- c. (3 pts)

Given that 11010100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-44

- d. (3 pts)

Given that 11101000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-24

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit apple lemon
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[1][1]`? p

c. (3 pts) What is the value of `argv[0][4]`? n

d. (3 pts) What is the value of `argv[2][1]`? e

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double w;  
    Node x;  
    int y;  
    char z;  
    double *a;  
    Node *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `argv[0]` **char ***
- b. (3 pts) `argv[1][2]` **char**
- c. (3 pts) `&y` **int ***
- d. (3 pts) `b->next` **Node ***
- e. (3 pts) `argc` **int**
- f. (3 pts) `b->data` **int**
- g. (3 pts) `&c` **int ****
- h. (3 pts) `y` **int**
- i. (3 pts) `c` **int ***
- j. (3 pts) `*a` **double**
- k. (3 pts) `b->next->next` **Node ***

-
5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #234 Page: 1 Name: _____

234

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #234 Page: 2 Name: _____

234

1. a. (2 pts) Convert 65 from octal to binary **110 101**

- b. (2 pts) Convert 0110 0110 1111 1001 from base 2 to base 16 **66f9**

- c. (2 pts) Convert 159 from decimal to binary **1001 1111**

- d. (2 pts) Convert 0110 0111 0010 0110 from base 2 to base 16 **6726**

- e. (2 pts) Convert 111 100 000 from base 2 to base 8 **740**

- f. (2 pts) Convert 53 from base 8 to binary **101 011**

- g. (2 pts) Convert 0111 0010 from base 2 to decimal **114**

- h. (2 pts) Convert 001 110 100 from base 2 to base 8 **164**

- i. (2 pts) Convert 1011 1010 0110 0011 from binary to hexadecimal **ba63**

3

Exam #234 Page: 3 Name: _____

234

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -108, what is this number's binary representation in 8-bit two's complement?

10010100

- c. (3 pts)

Given that 10010101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-107

- d. (3 pts)

Given that 10011100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-100

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit grape lime lemon mango
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][0]`? l

c. (3 pts) What is the value of `argv[0][1]`? /

d. (3 pts) What is the value of `argv[1][1]`? r

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int g;  
    double h;  
    Node p;  
    char q;  
    int *r;  
    double *s;  
    Node *t;  
    char *w;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) argv[1][2] **char**
- b. (3 pts) *w **char**
- c. (3 pts) argv[0] **char ***
- d. (3 pts) q **char**
- e. (3 pts) t->next **Node ***
- f. (3 pts) &r **int ****
- g. (3 pts) t->data **int**
- h. (3 pts) &g **int ***
- i. (3 pts) argc **int**
- j. (3 pts) t->next->next **Node ***
- k. (3 pts) s **double ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #235 Page: 1 Name: _____

235

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #235 Page: 2 Name: _____

235

-
1. a. (2 pts) Convert 0001 1011 from base 2 to decimal **27**
- b. (2 pts) Convert 0010 0001 from binary to decimal **33**
- c. (2 pts) Convert 141 from decimal to binary **1000 1101**
- d. (2 pts) Convert f311 from base 16 to binary **1111 0011 0001 0001**
- e. (2 pts) Convert 211 from base 10 to base 2 **1101 0011**
- f. (2 pts) Convert 0100 1001 1000 1010 from binary to hexadecimal **498a**
- g. (2 pts) Convert 0001 1100 from binary to base 10 **28**
- h. (2 pts) Convert 194 from decimal to base 2 **1100 0010**
- i. (2 pts) Convert 1000 0100 0111 1011 from base 2 to base 16 **847b**

3

Exam #235 Page: 3 Name: _____

235

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -14, what is this number's binary representation in 8-bit two's complement?

11110010

- c. (3 pts)

Given that 11111111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-1

- d. (3 pts)

Given that 10110101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-75

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit guava fig mango cherry
```

a. (3 pts) What is the value of `argc` in this case? **5**

b. (3 pts) What is the value of `argv[0][3]`? **u**

c. (3 pts) What is the value of `argv[1][1]`? **u**

d. (3 pts) What is the value of `argv[2][0]`? **f**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double c;  
    int d;  
    Node e;  
    char f;  
    double *g;  
    int *h;  
    Node *p;  
    char *q;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) p->next->next **Node ***
- b. (3 pts) argc **int**
- c. (3 pts) &f **char ***
- d. (3 pts) d **int**
- e. (3 pts) g **double ***
- f. (3 pts) p->next **Node ***
- g. (3 pts) argv[0] **char ***
- h. (3 pts) argv[1][2] **char**
- i. (3 pts) p->data **int**
- j. (3 pts) &p **Node ****
- k. (3 pts) *h **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #236 Page: 1 Name: _____

236

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #236 Page: 2 Name: _____

236

1. a. (2 pts) Convert 0111 0011 from binary to decimal **115**

- b. (2 pts) Convert 110 000 111 from binary to base 8 **607**

- c. (2 pts) Convert 328b from hexadecimal to base 2 **0011 0010 1000 1011**

- d. (2 pts) Convert 101 011 101 from binary to octal **535**

- e. (2 pts) Convert 0100 0100 0000 0101 from base 2 to hexadecimal **4405**

- f. (2 pts) Convert 010 010 101 from binary to octal **225**

- g. (2 pts) Convert 6db9 from base 16 to binary **0110 1101 1011 1001**

- h. (2 pts) Convert 32 from octal to base 2 **011 010**

- i. (2 pts) Convert 118 from decimal to binary **0111 0110**

3

Exam #236 Page: 3 Name: _____

236

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -58, what is this number's binary representation in 8-bit two's complement?

11000110

- c. (3 pts)

Given that 10010011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-109

- d. (3 pts)

Given that 10110101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-75

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lime kiwi date
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][0]`? .

c. (3 pts) What is the value of `argv[1][2]`? m

d. (3 pts) What is the value of `argv[2][2]`? w

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int t;  
    Node w;  
    double x;  
    char y;  
    int *z;  
    Node *a;  
    double *b;  
    char *c;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) c **char ***
- b. (3 pts) &a **Node ****
- c. (3 pts) a->next->next **Node ***
- d. (3 pts) *z **int**
- e. (3 pts) a->next **Node ***
- f. (3 pts) &y **char ***
- g. (3 pts) t **int**
- h. (3 pts) argv[0] **char ***
- i. (3 pts) argc **int**
- j. (3 pts) argv[1][2] **char**
- k. (3 pts) a->data **int**

6

Exam #236 Page: 6 Name: _____

236

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #237 Page: 1 Name: _____

237

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #237 Page: 2 Name: _____

237

-
1. a. (2 pts) Convert 1011 1000 0100 0110 from base 2 to base 16 **b846**

- b. (2 pts) Convert 0111 1110 from binary to decimal **126**

- c. (2 pts) Convert 32 from decimal to base 2 **0010 0000**

- d. (2 pts) Convert 0011 1010 from binary to base 10 **58**

- e. (2 pts) Convert 001 001 110 from base 2 to base 8 **116**

- f. (2 pts) Convert 71 from octal to base 2 **111 001**

- g. (2 pts) Convert 0001 0111 1010 1010 from binary to hexadecimal **17aa**

- h. (2 pts) Convert 111 100 000 from binary to base 8 **740**

- i. (2 pts) Convert 4105 from hexadecimal to base 2 **0100 0001 0000 0101**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

b. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

c. (3 pts)

Given that 11111101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-3

d. (3 pts)

Given that 11001110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-50

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit mango lime
```

a. (3 pts) What is the value of `argc` in this case? **3**

b. (3 pts) What is the value of `argv[2][3]`? **e**

c. (3 pts) What is the value of `argv[0][0]`? **.**

d. (3 pts) What is the value of `argv[1][4]`? **o**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double q;  
    Node r;  
    int s;  
    char t;  
    double *w;  
    Node *x;  
    int *y;  
    char *z;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) x->next->next **Node ***
- b. (3 pts) argc **int**
- c. (3 pts) t **char**
- d. (3 pts) x->data **int**
- e. (3 pts) &s **int ***
- f. (3 pts) argv[0] **char ***
- g. (3 pts) argv[1][2] **char**
- h. (3 pts) &z **char ****
- i. (3 pts) *y **int**
- j. (3 pts) x->next **Node ***
- k. (3 pts) x **Node ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #238 Page: 1 Name: _____

238

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #238 Page: 2 Name: _____

238

1. a. (2 pts) Convert e9a8 from hexadecimal to binary **1110 1001 1010 1000**

- b. (2 pts) Convert 79 from base 10 to base 2 **0100 1111**

- c. (2 pts) Convert 25 from octal to base 2 **010 101**

- d. (2 pts) Convert 150 from decimal to base 2 **1001 0110**

- e. (2 pts) Convert 0111 1101 from base 2 to base 10 **125**

- f. (2 pts) Convert 0001 0111 from base 2 to decimal **23**

- g. (2 pts) Convert 0001 1001 from base 2 to base 10 **25**

- h. (2 pts) Convert 89 from base 10 to base 2 **0101 1001**

- i. (2 pts) Convert e2c0 from hexadecimal to binary **1110 0010 1100 0000**

3

Exam #238 Page: 3 Name: _____

238

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -117, what is this number's binary representation in 8-bit two's complement?

10001011

- c. (3 pts)

Given that 10111110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-66

- d. (3 pts)

Given that 10000010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-126

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit guava apple lemon grape
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][1]`? p

c. (3 pts) What is the value of `argv[1][4]`? a

d. (3 pts) What is the value of `argv[0][1]`? /

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int c;  
    double d;  
    Node e;  
    char f;  
    int *g;  
    double *h;  
    Node *p;  
    char *q;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) p->next->next **Node ***
- b. (3 pts) c **int**
- c. (3 pts) &d **double ***
- d. (3 pts) argv[1][2] **char**
- e. (3 pts) *h **double**
- f. (3 pts) p->data **int**
- g. (3 pts) h **double ***
- h. (3 pts) argc **int**
- i. (3 pts) p->next **Node ***
- j. (3 pts) argv[0] **char ***
- k. (3 pts) &g **int ****

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #239 Page: 1 Name: _____

239

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #239 Page: 2 Name: _____

239

1. a. (2 pts) Convert 13 from octal to binary **001 011**

- b. (2 pts) Convert 9c5 from base 16 to binary **1001 1100 0101**

- c. (2 pts) Convert 44aa from base 16 to base 2 **0100 0100 1010 1010**

- d. (2 pts) Convert 34 from decimal to binary **0010 0010**

- e. (2 pts) Convert 60ba from base 16 to binary **0110 0000 1011 1010**

- f. (2 pts) Convert 1011 0000 from binary to decimal **176**

- g. (2 pts) Convert c3c9 from hexadecimal to binary **1100 0011 1100 1001**

- h. (2 pts) Convert e0e4 from base 16 to base 2 **1110 0000 1110 0100**

- i. (2 pts) Convert 1010 1100 1101 0111 from base 2 to base 16 **acd7**

3

Exam #239 Page: 3 Name: _____

239

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -94, what is this number's binary representation in 8-bit two's complement?

10100010

- b. (3 pts)

Given the decimal number -24, what is this number's binary representation in 8-bit two's complement?

11101000

- c. (3 pts)

Given that 10101000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-88

- d. (3 pts)

Given that 10011011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-101

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime kiwi fig banana
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][3]`? e

c. (3 pts) What is the value of `argv[2][3]`? i

d. (3 pts) What is the value of `argv[0][5]`? I

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int z;  
    double a;  
    Node b;  
    char c;  
    int *d;  
    double *e;  
    Node *f;  
    char *g;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) f->next->next **Node ***
- b. (3 pts) *g **char**
- c. (3 pts) b **Node**
- d. (3 pts) f->next **Node ***
- e. (3 pts) argc **int**
- f. (3 pts) &z **int ***
- g. (3 pts) &f **Node ****
- h. (3 pts) d **int ***
- i. (3 pts) f->data **int**
- j. (3 pts) argv[1][2] **char**
- k. (3 pts) argv[0] **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #240 Page: 1 Name: _____

240

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #240 Page: 2 Name: _____

240

1. a. (2 pts) Convert 72 from base 8 to base 2 **111 010**

- b. (2 pts) Convert 6778 from base 16 to base 2 **0110 0111 0111 1000**

- c. (2 pts) Convert 46 from base 10 to base 2 **0010 1110**

- d. (2 pts) Convert 6106 from base 16 to binary **0110 0001 0000 0110**

- e. (2 pts) Convert 0110 0111 from base 2 to base 10 **103**

- f. (2 pts) Convert 011 101 101 from binary to base 8 **355**

- g. (2 pts) Convert 1101 0111 0110 1000 from binary to hexadecimal **d768**

- h. (2 pts) Convert 100 100 010 from base 2 to octal **442**

- i. (2 pts) Convert 010 111 000 from base 2 to base 8 **270**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -118, what is this number's binary representation in 8-bit two's complement?

10001010

c. (3 pts)

Given that 11110010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-14

d. (3 pts)

Given that 11101100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-20

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit guava kiwi banana
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[1][2]`? a

c. (3 pts) What is the value of `argv[0][2]`? r

d. (3 pts) What is the value of `argv[2][2]`? w

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    Node x;  
    int y;  
    double z;  
    char a;  
    Node *b;  
    int *c;  
    double *d;  
    char *e;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `b->next` `Node *`
- b. (3 pts) `argv[0]` `char *`
- c. (3 pts) `argc` `int`
- d. (3 pts) `b->data` `int`
- e. (3 pts) `c` `int *`
- f. (3 pts) `argv[1][2]` `char`
- g. (3 pts) `y` `int`
- h. (3 pts) `*e` `char`
- i. (3 pts) `b->next->next` `Node *`
- j. (3 pts) `&x` `Node *`
- k. (3 pts) `&b` `Node **`

6

Exam #240 Page: 6 Name: _____

240

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students,int numStudents,StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #241 Page: 1 Name: _____

241

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #241 Page: 2 Name: _____

241

1. a. (2 pts) Convert 0010 1111 from base 2 to decimal **47**

- b. (2 pts) Convert 0010 0001 1010 0001 from binary to hexadecimal **21a1**

- c. (2 pts) Convert 27 from decimal to binary **0001 1011**

- d. (2 pts) Convert ecf0 from base 16 to base 2 **1110 1100 1111 0000**

- e. (2 pts) Convert 4b12 from base 16 to binary **0100 1011 0001 0010**

- f. (2 pts) Convert 4 from octal to base 2 **100**

- g. (2 pts) Convert 1000 0001 0101 1001 from base 2 to base 16 **8159**

- h. (2 pts) Convert 110 001 from base 2 to octal **61**

- i. (2 pts) Convert 0010 0110 from binary to decimal **38**

3

Exam #241 Page: 3 Name: _____

241

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -24, what is this number's binary representation in 8-bit two's complement?

11101000

- c. (3 pts)

Given that 11011101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-35

- d. (3 pts)

Given that 10000101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-123

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit date lime lemon
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][4]`? n

c. (3 pts) What is the value of `argv[1][0]`? d

d. (3 pts) What is the value of `argv[2][2]`? m

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node d;  
    int e;  
    double f;  
    char g;  
    Node *h;  
    int *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `h->next->next` **Node ***
- b. (3 pts) `g` **char**
- c. (3 pts) `h->next` **Node ***
- d. (3 pts) `&g` **char ***
- e. (3 pts) `argv[1][2]` **char**
- f. (3 pts) `argc` **int**
- g. (3 pts) `*h` **Node**
- h. (3 pts) `h->data` **int**
- i. (3 pts) `&q` **double ****
- j. (3 pts) `h` **Node ***
- k. (3 pts) `argv[0]` **char ***

6

Exam #241 Page: 6 Name: _____

241

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #242 Page: 1 Name: _____

242

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #242 Page: 2 Name: _____

242

1. a. (2 pts) Convert 011 000 010 from binary to octal **302**

- b. (2 pts) Convert 74 from base 8 to base 2 **111 100**

- c. (2 pts) Convert 24 from octal to base 2 **010 100**

- d. (2 pts) Convert 0100 1001 0011 0010 from base 2 to base 16 **4932**

- e. (2 pts) Convert 1010 0001 0011 0000 from base 2 to hexadecimal **a130**

- f. (2 pts) Convert 0100 0010 from base 2 to decimal **66**

- g. (2 pts) Convert 1000 0011 from binary to decimal **131**

- h. (2 pts) Convert 81ea from hexadecimal to binary **1000 0001 1110 1010**

- i. (2 pts) Convert 110 010 000 from base 2 to base 8 **620**

3

Exam #242 Page: 3 Name: _____

242

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -49, what is this number's binary representation in 8-bit two's complement?

11001111

- c. (3 pts)

Given that 10011110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-98

- d. (3 pts)

Given that 10111001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-71

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape lime
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[2][0]`? l

c. (3 pts) What is the value of `argv[0][3]`? u

d. (3 pts) What is the value of `argv[1][2]`? a

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    Node d;  
    double e;  
    int f;  
    char g;  
    Node *h;  
    double *p;  
    int *q;  
    char *r;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&e` **double ***
- b. (3 pts) `h->data` **int**
- c. (3 pts) `*h` **Node**
- d. (3 pts) `r` **char ***
- e. (3 pts) `argc` **int**
- f. (3 pts) `h->next->next` **Node ***
- g. (3 pts) `d` **Node**
- h. (3 pts) `&h` **Node ****
- i. (3 pts) `argv[0]` **char ***
- j. (3 pts) `h->next` **Node ***
- k. (3 pts) `argv[1][2]` **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #243 Page: 1 Name: _____

243

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #243 Page: 2 Name: _____

243

1. a. (2 pts) Convert 166 from decimal to base 2 **1010 0110**

- b. (2 pts) Convert 101 011 010 from base 2 to octal **532**

- c. (2 pts) Convert 4020 from base 16 to base 2 **0100 0000 0010 0000**

- d. (2 pts) Convert 65 from base 8 to binary **110 101**

- e. (2 pts) Convert 100 001 000 from base 2 to octal **410**

- f. (2 pts) Convert 110 111 001 from base 2 to octal **671**

- g. (2 pts) Convert 45 from base 10 to binary **0010 1101**

- h. (2 pts) Convert 99a from base 16 to binary **1001 1001 1010**

- i. (2 pts) Convert 44 from octal to binary **100 100**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -84, what is this number's binary representation in 8-bit two's complement?

10101100

c. (3 pts)

Given that 10001000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-120

d. (3 pts)

Given that 11010010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-46

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit fig guava lime banana
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[2][1]`? u

c. (3 pts) What is the value of `argv[1][0]`? f

d. (3 pts) What is the value of `argv[0][0]`? .

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int a;  
    double b;  
    Node c;  
    char d;  
    int *e;  
    double *f;  
    Node *g;  
    char *h;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `&g` **Node ****
- b. (3 pts) `argv[1][2]` **char**
- c. (3 pts) `argc` **int**
- d. (3 pts) `&a` **int ***
- e. (3 pts) `g` **Node ***
- f. (3 pts) `argv[0]` **char ***
- g. (3 pts) `g->next` **Node ***
- h. (3 pts) `g->data` **int**
- i. (3 pts) `*f` **double**
- j. (3 pts) `g->next->next` **Node ***
- k. (3 pts) `c` **Node**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #244 Page: 1 Name: _____

244

**CS16—Midterm Exam
E02, W15, Phill Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each pages is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #244 Page: 2 Name: _____

244

1. a. (2 pts) Convert fe78 from hexadecimal to base 2 **1111 1110 0111 1000**

b. (2 pts) Convert 24 from base 8 to base 2 **010 100**

c. (2 pts) Convert 1110 0101 1000 0001 from binary to hexadecimal **e581**

d. (2 pts) Convert 100 100 001 from binary to octal **441**

e. (2 pts) Convert 244 from base 10 to base 2 **1111 0100**

f. (2 pts) Convert 222 from decimal to binary **1101 1110**

g. (2 pts) Convert 0111 1111 0010 1011 from binary to base 16 **7f2b**

h. (2 pts) Convert 101 100 000 from base 2 to octal **540**

i. (2 pts) Convert 41 from base 8 to base 2 **100 001**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -128, what is this number's binary representation in 8-bit two's complement?

10000000

c. (3 pts)

Given that 10011100 is the 8-bit two's complement representation of a number, what is that number in base ten?

-100

d. (3 pts)

Given that 11010010 is the 8-bit two's complement representation of a number, what is that number in base ten?

-46

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit mango fig kiwi
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[0][6]`? t

c. (3 pts) What is the value of `argv[1][2]`? n

d. (3 pts) What is the value of `argv[2][0]`? f

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int r;  
    Node s;  
    double t;  
    char w;  
    int *x;  
    Node *y;  
    double *z;  
    char *a;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `y` **Node ***
- b. (3 pts) `t` **double**
- c. (3 pts) `y->data` **int**
- d. (3 pts) `argc` **int**
- e. (3 pts) `&r` **int ***
- f. (3 pts) `&y` **Node ****
- g. (3 pts) `argv[0]` **char ***
- h. (3 pts) `y->next->next` **Node ***
- i. (3 pts) `y->next` **Node ***
- j. (3 pts) `*y` **Node**
- k. (3 pts) `argv[1][2]` **char**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #245 Page: 1 Name: _____

245

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #245 Page: 2 Name: _____

245

1. a. (2 pts) Convert 20 from octal to base 2 **010 000**

- b. (2 pts) Convert 10 from decimal to binary **1010**

- c. (2 pts) Convert d362 from base 16 to binary **1101 0011 0110 0010**

- d. (2 pts) Convert 0001 1100 from binary to base 10 **28**

- e. (2 pts) Convert 1101 1000 0100 0000 from base 2 to hexadecimal **d840**

- f. (2 pts) Convert 7801 from hexadecimal to base 2 **0111 1000 0000 0001**

- g. (2 pts) Convert 0010 1001 0001 1011 from base 2 to hexadecimal **291b**

- h. (2 pts) Convert 55 from base 10 to binary **0011 0111**

- i. (2 pts) Convert 010 011 101 from binary to octal **235**

3

Exam #245 Page: 3 Name: _____

245

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -34, what is this number's binary representation in 8-bit two's complement?

11011110

- c. (3 pts)

Given that 10000110 is the 8-bit two's complement representation of a number, what is that number in base ten?

-122

- d. (3 pts)

Given that 11101011 is the 8-bit two's complement representation of a number, what is that number in base ten?

-21

4

Exam #245 Page: 4 Name: _____

245

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit fig cherry banana
```

a. (3 pts) What is the value of `argc` in this case? **4**

b. (3 pts) What is the value of `argv[2][4]`? **r**

c. (3 pts) What is the value of `argv[0][2]`? **r**

d. (3 pts) What is the value of `argv[1][1]`? **i**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int h;  
    Node p;  
    double q;  
    char r;  
    int *s;  
    Node *t;  
    double *w;  
    char *x;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `t->data` **int**
- b. (3 pts) `h` **int**
- c. (3 pts) `argc` **int**
- d. (3 pts) `&x` **char ****
- e. (3 pts) `&r` **char ***
- f. (3 pts) `t->next` **Node ***
- g. (3 pts) `s` **int ***
- h. (3 pts) `*w` **double**
- i. (3 pts) `argv[1][2]` **char**
- j. (3 pts) `argv[0]` **char ***
- k. (3 pts) `t->next->next` **Node ***

6

Exam #245 Page: 6 Name: _____

245

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #246 Page: 1 Name: _____

246

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #246 Page: 2 Name: _____

246

1. a. (2 pts) Convert 0111 0100 1011 1001 from base 2 to hexadecimal **74b9**

- b. (2 pts) Convert 1101 1011 from binary to base 10 **219**

- c. (2 pts) Convert 010 011 from base 2 to octal **23**

- d. (2 pts) Convert 120 from base 10 to binary **0111 1000**

- e. (2 pts) Convert 13 from base 8 to binary **001 011**

- f. (2 pts) Convert 1010 1010 0110 1011 from base 2 to hexadecimal **aa6b**

- g. (2 pts) Convert 0010 1011 0100 1001 from binary to base 16 **2b49**

- h. (2 pts) Convert a0ab from hexadecimal to binary **1010 0000 1010 1011**

- i. (2 pts) Convert 74 from base 8 to base 2 **111 100**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -59, what is this number's binary representation in 8-bit two's complement?

11000101

c. (3 pts)

Given that 11000111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-57

d. (3 pts)

Given that 10011111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-97

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit cherry banana
```

a. (3 pts) What is the value of `argc` in this case? 3

b. (3 pts) What is the value of `argv[1][0]`? c

c. (3 pts) What is the value of `argv[2][2]`? n

d. (3 pts) What is the value of `argv[0][6]`? t

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double z;  
    Node a;  
    int b;  
    char c;  
    double *d;  
    Node *e;  
    int *f;  
    char *g;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `g` `char *`
- b. (3 pts) `argc` `int`
- c. (3 pts) `e->next` `Node *`
- d. (3 pts) `e->next->next` `Node *`
- e. (3 pts) `*f` `int`
- f. (3 pts) `a` `Node`
- g. (3 pts) `argv[1][2]` `char`
- h. (3 pts) `argv[0]` `char *`
- i. (3 pts) `&b` `int *`
- j. (3 pts) `&d` `double **`
- k. (3 pts) `e->data` `int`

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #247 Page: 1 Name: _____

247

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #247 Page: 2 Name: _____

247

1. a. (2 pts) Convert 101 110 011 from binary to octal **563**

- b. (2 pts) Convert 149 from base 10 to binary **1001 0101**

- c. (2 pts) Convert 1111 0111 1010 0001 from base 2 to hexadecimal **f7a1**

- d. (2 pts) Convert 4 from base 10 to base 2 **0100**

- e. (2 pts) Convert 0001 0001 from base 2 to base 10 **17**

- f. (2 pts) Convert 4448 from base 16 to binary **0100 0100 0100 1000**

- g. (2 pts) Convert 1101 0101 0011 1010 from base 2 to base 16 **d53a**

- h. (2 pts) Convert 285c from hexadecimal to binary **0010 1000 0101 1100**

- i. (2 pts) Convert 101 110 101 from binary to octal **565**

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

b. (3 pts)

Given the decimal number -93, what is this number's binary representation in 8-bit two's complement?

10100011

c. (3 pts)

Given that 10110001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-79

d. (3 pts)

Given that 10111000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-72

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
...
```

Further, suppose this program is invoked with the following command line:

```
./runit grape cherry guava lime
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[1][2]`? a

c. (3 pts) What is the value of `argv[2][4]`? r

d. (3 pts) What is the value of `argv[0][3]`? u

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    int w;  
    double x;  
    Node y;  
    char z;  
    int *a;  
    double *b;  
    Node *c;  
    char *d;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) c->next **Node ***
- b. (3 pts) argc **int**
- c. (3 pts) argv[0] **char ***
- d. (3 pts) c->next->next **Node ***
- e. (3 pts) c->data **int**
- f. (3 pts) z **char**
- g. (3 pts) argv[1][2] **char**
- h. (3 pts) &x **double ***
- i. (3 pts) c **Node ***
- j. (3 pts) *d **char**
- k. (3 pts) &c **Node ****

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #248 Page: 1 Name: _____

248

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #248 Page: 2 Name: _____

248

1. a. (2 pts) Convert 100 011 from base 2 to octal **43**

- b. (2 pts) Convert 001 110 001 from base 2 to base 8 **161**

- c. (2 pts) Convert 47 from base 8 to base 2 **100 111**

- d. (2 pts) Convert 110 000 000 from binary to base 8 **600**

- e. (2 pts) Convert 40 from base 8 to binary **100 000**

- f. (2 pts) Convert 0100 0101 from base 2 to decimal **69**

- g. (2 pts) Convert 0010 0110 from binary to base 10 **38**

- h. (2 pts) Convert 110 011 101 from binary to octal **635**

- i. (2 pts) Convert 1010 1101 0010 0000 from base 2 to base 16 **ad20**

3

Exam #248 Page: 3 Name: _____

248

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -9, what is this number's binary representation in 8-bit two's complement?

11110111

- c. (3 pts)

Given that 11000101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-59

- d. (3 pts)

Given that 10110111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-73

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit kiwi mango date fig
```

a. (3 pts) What is the value of `argc` in this case? 5

b. (3 pts) What is the value of `argv[0][4]`? n

c. (3 pts) What is the value of `argv[1][0]`? k

d. (3 pts) What is the value of `argv[2][2]`? n

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[]) {  
    double g;  
    int h;  
    Node p;  
    char q;  
    double *r;  
    int *s;  
    Node *t;  
    char *w;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) argv[1][2] **char**
- b. (3 pts) &h **int ***
- c. (3 pts) t->next **Node ***
- d. (3 pts) *w **char**
- e. (3 pts) argc **int**
- f. (3 pts) q **char**
- g. (3 pts) argv[0] **char ***
- h. (3 pts) r **double ***
- i. (3 pts) &s **int ****
- j. (3 pts) t->next->next **Node ***
- k. (3 pts) t->data **int**

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #249 Page: 1 Name: _____

249

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #249 Page: 2 Name: _____

249

1. a. (2 pts) Convert 56d4 from hexadecimal to base 2 **0101 0110 1101 0100**

- b. (2 pts) Convert f2e7 from base 16 to binary **1111 0010 1110 0111**

- c. (2 pts) Convert 100 010 101 from base 2 to base 8 **425**

- d. (2 pts) Convert 0100 1100 0100 0010 from binary to hexadecimal **4c42**

- e. (2 pts) Convert 0110 0101 from base 2 to base 10 **101**

- f. (2 pts) Convert 110 111 110 from binary to octal **676**

- g. (2 pts) Convert 1101 0000 from base 2 to base 10 **208**

- h. (2 pts) Convert 567e from hexadecimal to binary **0101 0110 0111 1110**

- i. (2 pts) Convert 35 from octal to binary **011 101**

3

Exam #249 Page: 3 Name: _____

249

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -44, what is this number's binary representation in 8-bit two's complement?

11010100

- c. (3 pts)

Given that 10101111 is the 8-bit two's complement representation of a number, what is that number in base ten?

-81

- d. (3 pts)

Given that 11010001 is the 8-bit two's complement representation of a number, what is that number in base ten?

-47

3. Assume the `main` function in the program `runit.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runit lemon mango apple
```

a. (3 pts) What is the value of `argc` in this case? 4

b. (3 pts) What is the value of `argv[2][0]`? m

c. (3 pts) What is the value of `argv[0][0]`? .

d. (3 pts) What is the value of `argv[1][0]`? l

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    int d;  
    Node e;  
    double f;  
    char g;  
    int *h;  
    Node *p;  
    double *q;  
    char *r;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. int, int *, etc.

- a. (3 pts) argv[1][2] **char**
- b. (3 pts) &d **int ***
- c. (3 pts) *h **int**
- d. (3 pts) argc **int**
- e. (3 pts) &r **char ****
- f. (3 pts) p->data **int**
- g. (3 pts) p->next **Node ***
- h. (3 pts) argv[0] **char ***
- i. (3 pts) p->next->next **Node ***
- j. (3 pts) e **Node**
- k. (3 pts) r **char ***

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

1

Exam #250 Page: 1 Name: _____

250

**CS16—Midterm Exam
E02, W15, Phillip Conrad, UC Santa Barbara
Wednesday, 03/09/2015**

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
 - Be sure you turn in every page of this exam.
 - Each exam is numbered (e.g. Exam #137).
 - Each page is numbered (e.g. Page 1, Page 2, etc.)
 - The last page clearly says "End of Exam".
 - This exam is **closed book, closed notes, closed mouth, cell phone off**
 - You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
 - These sheets will be collected with the exam, and might not be returned
 - Please write your name on your notes sheet
-

2

Exam #250 Page: 2 Name: _____

250

1. a. (2 pts) Convert 136 from base 10 to binary **1000 1000**

- b. (2 pts) Convert 196 from decimal to base 2 **1100 0100**

- c. (2 pts) Convert c142 from base 16 to base 2 **1100 0001 0100 0010**

- d. (2 pts) Convert 1010 1000 from binary to decimal **168**

- e. (2 pts) Convert 187 from base 10 to base 2 **1011 1011**

- f. (2 pts) Convert 17 from decimal to base 2 **0001 0001**

- g. (2 pts) Convert 1101 0011 0000 1100 from binary to hexadecimal **d30c**

- h. (2 pts) Convert bf6d from base 16 to binary **1011 1111 0110 1101**

- i. (2 pts) Convert 6 from base 8 to base 2 **110**

3

Exam #250 Page: 3 Name: _____

250

2. For these questions, assume 8-bit two's complement representation of negative integers. The left-most bit is the sign bit, with 1 representing negative numbers, and 0 representing positive numbers.

- a. (3 pts)

Given the decimal number -95, what is this number's binary representation in 8-bit two's complement?

10100001

- b. (3 pts)

Given the decimal number -69, what is this number's binary representation in 8-bit two's complement?

10111011

- c. (3 pts)

Given that 11110000 is the 8-bit two's complement representation of a number, what is that number in base ten?

-16

- d. (3 pts)

Given that 10000101 is the 8-bit two's complement representation of a number, what is that number in base ten?

-123

4

Exam #250 Page: 4 Name: _____

250

3. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[ ]) {  
    ...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana apple
```

a. (3 pts) What is the value of `argc` in this case? **3**

b. (3 pts) What is the value of `argv[0][1]`? **/**

c. (3 pts) What is the value of `argv[2][0]`? **a**

d. (3 pts) What is the value of `argv[1][0]`? **b**

4. Given the following declarations:

```
struct Node {  
    int data;  
    Node *next;  
};  
  
int main(int argc, char *argv[ ]) {  
    double w;  
    Node x;  
    int y;  
    char z;  
    double *a;  
    Node *b;  
    int *c;  
    char *d;  
  
    return 0;  
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

- a. (3 pts) `d` `char *`
- b. (3 pts) `&b` `Node **`
- c. (3 pts) `b->next->next` `Node *`
- d. (3 pts) `b->next` `Node *`
- e. (3 pts) `b->data` `int`
- f. (3 pts) `*a` `double`
- g. (3 pts) `argv[0]` `char *`
- h. (3 pts) `y` `int`
- i. (3 pts) `argc` `int`
- j. (3 pts) `argv[1][2]` `char`
- k. (3 pts) `&y` `int *`

6

Exam #250 Page: 6 Name: _____

250

5. (5 pts) Early in the quarter, we did not use a Makefile in our projects. Later in the quarter, we did. What changed? That is, at a "big picture" level, what is the purpose of using a Makefile in a software development project in C++?

NOTE: Keep your answer short and precise. I am looking for a SHORT answer that addresses the main point—not a "brain dump" with everything you know about Makefiles. You will also not get any credit for answers that happen to be "true statements" about Makefiles but that are not relevant to what the question is asking.

6. (10 pts) Using the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam, write the definition of the function `numHonorsStudents`, as described below.

`numHonorsStudents` takes two parameters:

- `students` is an array of `StudentGpa` structs
- `numStudents` is the number of elements in that array (the occupancy)

`numHonorsStudents` should return a count of the students in the `StudentGpa` array that have GPAs that are greater than or equal to 3.5

Write ONLY the function definition—NOTHING ELSE. No `#includes`, no `using namespace std;`, no main function.

7. Fill in the incomplete parts of the function definitions for `addPermToList` and `honorsStudentList` below. The function definition uses the definitions for the structs `StudentGpa`, `Student` and `StudentList` on the handout you got with this exam. Assume that the necessary `#includes` to pull in those function definitions have already been done.

A few lines of code are missing, as indicated by comments in the code. In each spot, **fill in the missing lines of code**.

The description of what `honorsStudentList` is supposed to do is on your handout.

```
// Fill in parameters // (a) (2 pts)
void addPermToList( ) {

    // Declare a variable s of type pointer to Student, and
    // initialize it to a new Student struct on the heap (b) (2 pts)

    s->perm = perm;
    s->next = NULL;

    if (sList->head == NULL) {
        sList -> head = s;
        // Fill in another line of code here (c) (2 pts)

    } else {
        // Fill in a line of code here @@@ (d) (2 pts)

        sList -> tail = s;
    }
}

void honorsStudentList(StudentGpa *students, int numStudents, StudentList *sList){
    for (int i=0; i<numStudents; i++) {
        if ( students[i].gpa >= 3.5 ) {
            // call addPermToList with appropriate parameters (e) (2 pts)

        }
    }
}
```

