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 8415 from hexadecimal to base 2 1000 0100 0001 0101

   b. (2 pts) Convert 57 from octal to base 2 101 111

   c. (2 pts) Convert 0100 1001 0100 1010 from base 2 to hexadecimal 494a

   d. (2 pts) Convert 75 from octal to base 2 111 101

   e. (2 pts) Convert 1110 0101 from binary to decimal 229

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

   g. (2 pts) Convert 6086 from hexadecimal to base 2 0110 0000 1000 0110

   h. (2 pts) Convert 129 from decimal to base 2 1000 0001

   i. (2 pts) Convert 2fc6 from base 16 to base 2 0010 1111 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 -91, what is this number's binary representation in 8-bit two's complement?

   10100101

   b. (3 pts)

   Given the decimal number -67, what is this number's binary representation in 8-bit two's complement?

   10111101

   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 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 lime lemon
```

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

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

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

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 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) &c `double **`

b. (3 pts) argv[0] `char *`

c. (3 pts) *e `char`

d. (3 pts) b->next->next `Node *`

e. (3 pts) z `int`

f. (3 pts) b->next `Node *`

g. (3 pts) argc `int`

h. (3 pts) b->data `int`

i. (3 pts) argv[1][2] `char`

j. (3 pts) d `int *`

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 adresses 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 `#include`s 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)



    }
  }
}
```