

CSE 333 Midterm Exam 5/10/13

Name _____

There are 5 questions worth a total of 100 points. Please budget your time so you get to all of the questions. Keep your answers brief and to the point.

The exam is closed book, closed notes, closed electronics, closed telepathy, open mind.

If you don't remember the exact syntax for something, make the best attempt you can. We will make allowances when grading.

Don't be alarmed if there seems to be more space than is needed for your answers – we tried to include more than enough blank space.

Relax, you are here to learn.

Please wait to turn the page until everyone is told to begin.

Score _____ / 100

1. _____ / 18

2. _____ / 20

3. _____ / 22

4. _____ / 22

5. _____ / 18

CSE 333 Midterm Exam 5/10/13

Question 1. (18 points) Consider these two C files:

a.c

```
void f(int p);  
  
int main() {  
    f(17);  
    return 0;  
}
```

b.c

```
void f(char *p) {  
    *p = 'x';  
}
```

(a) Why is the program made from a.c and b.c incorrect? What would you expect to happen if it is executed?

(b) Will `gcc -Wall -c a.c` and `gcc -Wall -c b.c` give an error or will they successfully produce a.o and b.o without complaint?

(c) Will `gcc -Wall a.c b.c` give an error or will it successfully produce a.out without complaint?

(d) How would you use standard C coding practices (using an extra file) to avoid the problems with this program (or at least detect them properly)? Give the contents of that extra file below and explain what modifications should be made to a.c and/or b.c, if any.

CSE 333 Midterm Exam 5/10/13

Question 2. (20 points) Consider the following rather twisted C program, which does compile and execute with no warnings or errors:

```
#include <stdio.h>

int *conundrum(int *a, int b, int **c) {
    **c = b + *a;
    *c = a;
    **c = *a * b;
    // HERE
    return a;
}

int main() {
    int p = 2, q = 3, r = 7;
    int *d = &r;
    d = conundrum(&p, q, &d);
    printf("%d %d %d %d\n", p, q, r, *d);
    return 0;
}
```

(a) Draw a boxes 'n arrows diagram showing state of memory when control reaches the comment containing `HERE`, right before executing the `return` statement in function `conundrum`. Your diagram should have two boxes showing the stack frames for functions `main` and `conundrum`. The stack frames should include values of integer variables and an arrow from each pointer to the location that it references. Then answer part (b) at the bottom of the page.

(b) What output does this program produce when it is executed?

CSE 333 Midterm Exam 5/10/13

Question 3. (22 points) The nodes in a linked list of C strings can be defined as follows:

```
typedef struct snode {
    char * str;          // this node's heap-allocated string
    struct snode * next; // next node in the list or NULL
} Snode;
```

Complete the definition of function `Clone` below so that it returns (a pointer to) an exact duplicate of the list that is its argument, including duplicates of all the nodes and strings in the original list. You may use `strcpy` instead of `strncpy`, and may assume that all strings in the original list are properly `\0`-terminated. You may assume that `malloc` will always successfully allocate data when it is called. Also assume that all necessary library header files have already been `#included`. Hint: `strcpy(dst, src)` copies the string `src` to the string `dst`.

```
// return a clone of the linked list with first node lst
// (which may be NULL)
Snode * Clone(Snode * lst) {
```

```
}
```

(additional room on the next page if needed)

CSE 333 Midterm Exam 5/10/13

Question 3. (cont.) Additional room for your answer if you need it.

CSE 333 Midterm Exam 5/10/13

Question 4. (22 points) For this problem consider the C++ class `Vec` on the next page. This class is supposed to implement a simple vector of integers. All of the code is written in the class declaration instead of in separate `.h` and `.cc` files to get it to fit on one page for the exam.

The class has a constructor, copy constructor, assignment, methods to set and get individual elements from the vector, and a destructor. The assignment and copy constructor operations are supposed to make a complete copy (clone) of their argument.

The class compiles without any errors or warnings, but when it is used in a program it generally segfaults, and, even if it doesn't crash, valgrind reports all sorts of memory management problems.

Mark the code on the following page to identify the problems, and write in corrections so the class will work as intended. Keep your notes brief and to the point.

You should ignore possible problems with invalid index values or length arguments – i.e., assume the length provided to the constructor is positive and that the index arguments to `get` and `set` are within bounds (i.e., $0 \leq \text{index} < \text{len_}$). You can also assume that heap allocation (`new`) always succeeds. (There are enough other problems in the code without worrying about these possibilities. ☺)

You can use the space below if you need extra room to write explanations or corrections, but *please* help the graders by making it easy to read your changes and figure out where they fit.

CSE 333 Midterm Exam 5/10/13

Question 4. (cont.) Find the bugs in the C++ code below, give a *very* brief description of the problems, and correct the code so it works properly.

```
class Vec { // a vector of integers
public:

    // initialize new Vec with n elements all set to 0. Assume n>0.
    Vec(int n) {
        v_ = new int[n];
        len_ = n;
        for (int i=0; i<len_; i++) v_[i] = 0;
    }

    // copy constructor - initialize *this to be a clone of other
    Vec(const Vec &other) {
        v_ = other.v_;
        len_ = other.len_;
    }

    // destructor - free resources
    ~Vec() { delete v_; }

    // replace the contents of *this with a clone of rhs
    Vec &operator=(const Vec &rhs) {
        delete v_;
        v_ = rhs.v_;
        len_ = rhs.len_;
        return *this;
    }

    // get/set functions. Assume that 0<=index<len_ (i.e., for this
    // question don't worry about index out of bounds problems)
    int get(int index) const { return v_[index]; }
    void set(int index, int n) { v_[index] = n; }

private:
    int* v_; // array of int values allocated on the heap
    int len_; // number of ints in v_
};
```

CSE 333 Midterm Exam 5/10/13

Question 5. (18 points) A few short questions to wrap up. For each question circle the correct choice. You do not need to justify your answers.

(a) If `a` is declared as `char a[5]`, then `a[3] == *(a+3)` is always true. (circle)

True False

(b) If `a` is declared as `int32_t a[5]`, then `a[3] == *(a+12)` is always true. (Recall that a `int32_t` value occupies 4 bytes.) (circle)

True False

(c) If we execute program `xyzzzy` using the command line `./xyzzzy one two`, then the value of `argv[0]` in the main program will be the string `one`. (circle)

True False

(d) If on our Linux system a program contains a pointer variable declared as `int *p` and in the debugger we see that the value of `p` is `0x7fffffff318`, we can conclude that: (circle the best choice)

(i) `p` refers to a variable allocated on the heap

(ii) `p` refers to a local variable in some function's stack frame

(iii) `p` refers to constant data or to a location in the program's x86 machine code

(iv) `p` is NULL

(v) we cannot conclude anything about the data referenced by `p`

(e) The system-level `read` function normally returns the number of bytes read. If it returns the value `-1` (error), then the program should not attempt any further I/O operations on that stream because it would be a fatal error to do so. (circle)

True False

(f) The `stat` system function returns information about a file. That information includes a field `st_mode` that, among other things, describes the type of the file. All of these files are classified as "regular files" as opposed to some other type: `ascii` text, Word `.docx` files, `jpeg` picture files, compiler produced `.o` files, and executable files like `a.out`. (circle)

True False