**ECE242, Fall 2002, Section 2, Quiz 1** **Thursday October 3**

**Name:** **Student Number:**

1. [**4 marks:**] Recall that if `a` and `b` are the same type of `struct`, then the assignment `a = b` assigns the value of `b` to `a`. Consider the following code and answer the questions below.

```
int main()
{
    struct {
        int n;
        int *ip;
    } a, b;

    a.n = 7;
    a.ip = &b.n;
    b = a;
    b.n = 3;

    /* comment A */
    return 0;
}
```

At the line `/* comment A */`, what is the value of:

(a) `a.n == 7`

(b) `b.n == 3`

(c) `*a.ip == 3`

(d) `*b.ip == 3`

2. [**3 marks:**] Replace the comment `/* write something recursive here */` by a call to `palindrome` itself, in order to complete the recursive function below.

```
#include <string.h>
typedef enum {FALSE, TRUE} boolean;

/* precondition: substring is an array of characters,
 *               not necessarily terminated by '\0'.
 *               0 <= start and stop < strlen(substring)
 * postcondition: return TRUE if substring[start] ...
 *                substring[stop] is the same in forward
 *                or reverse order.  Always return TRUE if
 *                stop < start.
 */
boolean palindrome(char *substring, int stop, int start)
{

    if (stop < start)
        return TRUE;
    else
        return (substring[start] == substring[stop] &&
                palindrome(substring, stop - 1, start + 1); /* <-- replacement */
}
```

**[Turn over page, please]**

3. [**3 marks:**] Fill in the condition for the while loop (replace `/* some condition here */`), an expression involving a pointer (replace `/* some pointer expression here */`), and a loop increment (replace `/* some increment here */`), to complete the function `oddChars` below. You may **not** use subscripting, i.e. `string[k]` or similar constructs aren't allowed.

```
#include <stdio.h>
#include <string.h>

/*
 * precondition: string is an array of 1 or more
 *               characters followed by '\0'
 *
 * postcondition: the elements of string with odd
 *                index (1, 3, 5, ...) that precede
 *                '\0' are printed in sequence on
 *                the console.
 */
void oddChars(char *string)
{
    int k = 0, len = strlen(string);

    while( k + 1 < len ) { /* <-- replacement */
        printf("%c", *(string + k + 1)); /* <-- replacement */
        k += 2; /* <-- replacement */
    }
    printf("\n");
}
```

4. [**3 marks:**] What is output by the following code fragment? How would you change the code so that the output is the same, but the intent is clearer to the reader?

```
#include <stdio.h>

int main()
{
    int i = 0, j = 1;

    if (i < 1)
        if (j < i)
            printf("i < 1 and j < i\n");
    else
        printf ("i >= 1\n");

    return 0;
}
```

Output: `i >= 1`. I would indent the two lines

```
else
    printf ("i >= 1\n");
```

. . . four more spaces, so that the `else` lined up under

```
if (j < i)
```

**[End of quiz]**