

Problem 3 (25 points): C Strings

The C function below is intended to find the first copy of a string *s* in a second string *t*. If the first string (not including the terminal NUL) appears within the second string, the function returns a pointer to the start of the copy. If the first string does not appear within the second string, the function returns `NULL`.

Unfortunately, Prof. Lumetta left a few bits out.

```
char* substring (char* s, char* t)
{
    char* check;
    char* cmp;
    char* scan;

    for (check = s; _____ ; check++) { // blank #1
        for (cmp = check, scan = t;
             _____ && '\0' != *scan; // blank #2
             cmp++, scan++) {
            _____ ; // blank #3
        }
        if ('\0' == *scan) {
            return _____ ; // blank #4
        }
    }
    return _____ ; // blank #5
}
```

Circle EXACTLY ONE ANSWER to indicate what should appear in each blank in the code above.

(5 points): blank #1

- A) `'\0' != check++` B) `'\0' != *check` C) `t != *check` D) `0 != check` E) `t > check`

(5 points): blank #2

- A) `'\0' != *cmp` B) `s != scan` C) `*scan == *cmp` D) `scan == cmp` E) `1`

(5 points): blank #3

- A) nothing B) `cmp = check` C) `scan--` D) `*scan = '\0'` E) `scan = check`

(5 points): blank #4

- A) `check` B) `t` C) `scan` D) `cmp` E) `NULL`

(5 points): blank #5

- A) `1` B) `scan` C) `cmp` D) `NULL` E) `check`

Problem 4 (25 points): Arrays

The function below is supposed to return the value of the largest element in an array of integers. The function takes three parameters: `list` (the array of integers), `length` (the length of the array), and `index` (a parameter written by `find_max` to indicate the position of the maximum element in the array). If multiple elements in the array have the maximum value, the function should return the index of the LAST such element. For example, if `list = [10, 89, 8, 43]`, `find_max` returns the value 89 and modifies the value pointed to by `index` to the value 1. You may assume that `length` is positive.

Unfortunately, Prof. Lumetta has again left blanks in the code.

```
int find_max ( _____ , int length, int* index) // blank #1
{
    int p;
    int m;
    int i;

    p = 0;
    m = list[p];
    for (i = _____ ; length > i; i++) {           // blank #2

        if ( _____ ) {                           // blank #3
            m = list[i];
            p = i;
        }

        _____ ;                                   // blank #4
    }
    _____ ;                                       // blank #5
}
```

Circle EXACTLY ONE ANSWER to indicate what should appear in each blank in the code above.

(5 points): blank #1

- A) `int list[]` B) `list[length]` C) `int list` D) `array list` E) `int** list`

(5 points): blank #2

- A) `m` B) `length` C) `p - 1` D) `1` E) `42`

(5 points): blank #3

- A) `m == p` B) `m <= list[i]` C) `m < list[p]` D) `m > list[i]` E) `m < list[i]`

(5 points): blank #4

- A) `*index = p` B) `m = list[i]` C) `p = *p` D) `*p = index` E) `*index = m`

(5 points): blank #5

- A) `m = length` B) `empty` C) `return p` D) `m = list[p]` E) `return m`