CS 7B - Fall 2017 - Pointer and Reference Quiz 04 Solutions

Consider the following nearly complete program:

```
#include <iostream>
using namespace std;

char* myst01(const char* s, int num_max = 128) {
    int i = 0;
    const char* p = s;
    while (*p++) ++i;
    if (i>num_max) i = num_max;

    char* s_cpy = new char[i+1];

    char* s_cpy_ptr = s_cpy;
    int ctr = 0;
    while (*s && ctr<num_max) {
        *s_cpy_ptr++ = *s++;
        ++ctr;
    }
    *s_cpy_ptr = 0;
    return s_cpy;
}

char* myst02(const char* s, const char* x,
             int num_max_s = 128, int num_max_x = 128) {
    if (x==0 || !*x) return const_cast<char*>(s);
    if (s==0 || !*s) return 0;

    for (int i = 0; *(s+i) && i<num_max_s; ++i) {
        if (*(s+i) == *x) {
            for(int j=0; *(x+j) && j<num_max_x; ++j) {
                if (!*(s+i+j) || (i+j)==num_max_s)
                   return 0;
                if (*(s+i+j) != *(x+j)) break;
                if (!*(x+j+1) || (j+1)==num_max_x)
                    return const_cast<char*>(s+i);
            }
        }
    }
    return 0;
}

int main()
{
    // test myst01 function
    char a[] = "test";
    char* a_dpl = myst01(a,5);
    print(a_dpl);
    cout << "\n";
    char b[] = { 't','e','s','t' };
    char* b_dpl = myst01(b,4);
    print(b_dpl);
    cout << "\n";
    delete[] b_dpl;
    delete[] a_dpl;

    // test myst02 function
    char s[] = "xxxtestxxx";
    char x[] = "test";
    char* sub = myst02(s,x);
    print(sub);
}
```

1. What is happening on line 6?
   The variable `p` of type pointer to character is declared and defined as equal to the variable `s`, which is passed by value to the function `myst01()` whose scope `p` is in.

2. Describe what the `while` loop on line 7 is doing. In particular which operation happens first in the expression `p++`? Dereferencing or increment?
   The pointer `p` is first incremented and then dereferenced so that if it is pointing to the `NUL char` then the condition will be false, When the loop terminates, `i` will contain the number of characters in the c-string before the terminating `NUL`.

3. What happens on line 10? Why is 1 added to `i` here?
   The variable `s_cpy` of type pointer to `char` is created and defined as the first memory address of a block of `i+1` `char`s reserved for `s_cpy`. The "+1" is for the terminating `NUL`.

4. Rewrite the `while` loop on lines 13-17 using a `for` loop.

```
for(int ctr = 0; *s && ctr<num_max; ++ctr)
   *s_cpy_ptr++ = *s++;
```

5. Describe in detail what `myst02()` does and how it does it.
   The function `myst02()` takes two pointers to constant `char`s (C-strings) named `s` and `x`. It also takes two `int`s (by value) with names `num_max_s` and `num_max_x` that have default values of 128. If the second C-string pointer (`x`) is the `NULL` pointer, or it's pointing to the `NUL` character, then just return the first C-string, `s`, after casting it as a pointer to a `char` (not constant.)
   If the first C-string passed is the `NULL` pointer, or if it points to the `NUL` character, then just return the `NULL` pointer. Otherwise, both C-string parameters are non-empty. The `for`-loop on line 27 walks character by character throught the C-string `s` with index `i`, and if the `i`th character matches the character pointed to by `x` then we enter a sub-`for`-loop that marches character by character through the characters of the C-string `x` with the index `j`, and if the character at position `i+j` in `s` is the `NUL` character or `i+j` has reached the maximum `num_max_s` then `myst02()` returns the `NULL` pointer, otherwise, if the character at position `i+j` in `s` is *not* the same as the character in position `j` of `x`, we break out of the sub-`for`-loop and increment `i` before reentering the sub-`for`-loop. If the character at position `i+j` in `s` *is* the same as the character in position `j` then we check to see if the next character in `x` is the `NUL` character and, if so, then we've found the C-string `x` as a substring of `s` and we return the substring of `s` starting at position `i`.

6. What is the output of the `main()`? Assume the print function works as expected.
   The function `myst01()` will duplicate the C-string `a` and assign the duplicate to the C-string `a_dpl`. Similarly `b` is duplicated to `b_dpl` (though `b` is initialized in a different manner, this is otherwise exactly the same. Finally we test the function `myst02()` and see that it returns and prints "testxxx", since the substring "test" was found int "xxxtestxxx".