Consider the following nearly complete program:

```cpp
#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?

2. Describe what the `while` loop on line 7 is doing. In particular which operation happens first in the expression `p++`? Dereferencing or increment?

3. What happens on line 9? Why is 1 added to `i` here?

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

5. Describe in detail what `myst02()` does and how it does it.

6. What is the output of the `main()`? Assume the print function works as expected.