

Quiz 4 CMPSC 16, PROBLEM SOLVING WITH COMPUTERS, SUMMER 2016

NAME:

DATE:

Problem 1.(10 points.) What property of a function makes it recursive (i.e., what do recursive functions do that make them different from normal functions)?

They call themselves.

Problem 2.(10 points.) What is the role of the stack when a recursive function is executing?

The stack is a last-in-first-out (LIFO) data structure that stores the temporary variables for each instantiation of the function call (i.e., it stores all of the variables etc.

Problem 3.(10 points.) What address will `ptr[2]` reference if `char *ptr = 0x1000`?

It will point to the memory location 0x1002 since `ptr[i]` is just shorthand for `*(ptr+i)` and each char is 1 byte. (Note that if `ptr` was an `int`, this would be 0x1008)

Problem 4.(10 points.) What is special about the `'\0'` character when dealing with character arrays (i.e., `char []`)?

It denotes the end of a string.

Problem 5.(10 points.) What will this program output?

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

int main() {

    string start = "cs_sixteen";
    vector<char> str_vector;

    int i = 0;
    for (char c: start) {
        if (i) {
            str_vector.push_back(tolower(c));
        } else {
            str_vector.push_back(toupper(c));
        }
        i = !i;
    }

    for (char c: str_vector) {
        cout << c;
    }
    return 0;
}
```

Cs sIxTeEn

Problem 6.(10 points.) What will this program output?

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

int main() {

    int numbers[] = {1,2,3,4,5};
    int *ptr = numbers;

    for (int i = 0; i < 5; i++) {
        *(ptr+i) = *(ptr+i) + 5;
    }
    for (int i = 0; i < 5; i++) {
        cout << numbers[i] << " ";
    }

    return 0;
}
```

6 7 8 9 10

Problem 7 [Recursion].(10 points.) What should this program do? Where is the bug, and what effect will it have? Fix it.

```
#include <iostream>

using namespace std;

long fact(long i) {
    return i * fact(i-1);
}

int main(int argc, const char * argv[]) {

    int number = 0;
    long answer = 0;

    cout << "Give me a number:" << endl;
    cin >> number;

    answer = fact(number);

    cout << "Answer:_" << answer << endl;

    return 0;
}
```

This program will compute the factorial of a number (e.g., number!). The bug is that there is no base case for the recursive function, so the program will recurse infinitely, resulting in a stack overflow!

Problem 8 [Pointer Confusion].(10 points.) What should this program do? Where is the bug, and what effect will it have? Fix it.

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

int main() {

    const int nums = 5;
    long long big_numbers[nums] = {0};

    for (int i = 0; i < nums; i++) {
        cout << "Give_me_a_number:" << endl;
        cin >> big_numbers[i];
    }

    // Pointers to the first and last element
    long long *first = big_numbers;
    long long *last = big_numbers+nums-1;

    long long tmp = big_numbers[0];
    *first = *last;
    *last = *(long long *)tmp;

    for (long long ll: big_numbers) {
        cout << ll << "_";
    }

    return 0;
}
```

This program will read in 5 numbers and swap the first in the last. However, it treats the first number input by the user as a pointer, so it will copy the contents of that address (instead of the value) into the array, and print it. (This is known as a memory disclosure bug)

Problem 9.(10 points.) What will this program do (brief description)?

```
#include <iostream>
#include <vector>

using namespace std;

bool check(char *ptr1, char *ptr2) {
    while (ptr2 > ptr1) {
        if (*ptr1 != *ptr2) {
            return false;
        }
        ptr1++;
        ptr2--;
    }
    return true;
}

int main(int argc, const char * argv[]) {

    int len = 0;

    cout << "How long:" << endl;
    cin >> len;

    char *input = new char(len);
    char *ptr = input+len-1;

    cout << "Type your string:" << endl;
    cin >> input;

    if (check(input, ptr)) {
        cout << "Yes" << endl;
    } else {
        cout << "No" << endl;
    }

    return 0;
}
```

It will read a string from the user of a given length, and output “Yes” if its a palindrome or “No” otherwise.

Problem 10.(10 points.) What will this program do (brief description)?

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

// Note: substr(x,y) returns a substring starting at x and ending at y
//       Ex: string("abcdef").substr(1,2) returns "bc"

string do_something(const string& str)
{
    if (str.length() <= 1) {
        return str;
    }
    else if (str[0] == 'x') {
        return do_something(str.substr(1, (str.length() - 1))) + str[0];
    }
    else {
        return str[0] + do_something(str.substr(1, str.length()));
    }
}

int main() {
    string str;

    cout << "Give_me_some_chars:" << endl;
    cin >> str;

    string new_str = do_something(str);
    cout << "Output:_" << new_str << endl;

    return 0;
}
```

It will take in a string and return that string with all x's moved to the end (e.g., "xyxz" becomes "yzxx").