

Lab 4 Accompanying Document

Harry Haisty

September 2018

Question 1: Summary

First, I removed the constant identifier from each of the values so that they could be malleable. Then, I removed the asterisks and ampersands so that the address values would not be assigned to num1, num2, or num3.

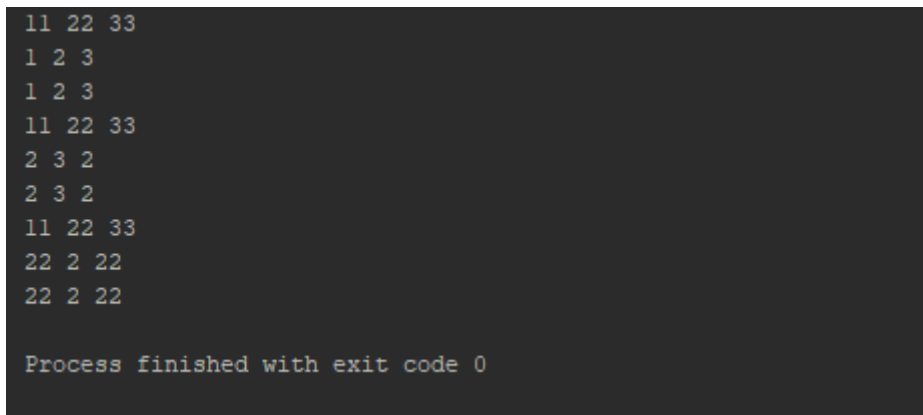
After that, the pre-existing code set p1, p2, and p3 to num1, num2, and num3, respectively. Then, I took off the asterisks on p1, p2, and p3 in the second line so that there would not be a type mismatch, which there was before.

*p2 was not a valid assignment, since the address cannot be set, so I removed the asterisk from p2.

p1, p2, and p3 were then set by the original program to each other, with p1 being set to p2, p2 being set to p3, and p3 being set to p1.

Then, num1, num2, and num3 were printed out, and their values had not changed. p1, p2, and p3 were then printed with their new assigned values from the assignment switches.

Question 1: The Output



```
11 22 33
1 2 3
1 2 3
11 22 33
2 3 2
2 3 2
11 22 33
22 2 22
22 2 22

Process finished with exit code 0
```

Question 1: The Code

```
#include <iostream>

using namespace std;

int main() {
    //remove constant from integers so they are malleable.
    //remove *'s and &'s so that value is not accessing actual location
    //initializes num1, num2, and num3 to 1, 2, and 3
    int num1 = 1, num2 = 2, num3 = 3;

    //sets p1, p2, and p3 to initialized values of num1, num2, and num3
    int p1 = num1;
    int p2 = num2;
    int p3 = num3;

    //sets num1, num2, and num3 to 11, 22, and 33 respectively
```

```

num1 = 11;
num2 = 22;
num3 = 33;

//prints out values of num1, num2, and num3 with a space between the values
cout << num1 << " " << num2 << " " << num3 << endl;

//prints out p1, p2, and p3, which are the original values of num1, num2, and num3. Twice.
cout << p1 << " " << p2 << " " << p3 << endl;
cout << p1 << " " << p2 << " " << p3 << endl;

//sets each value to the next one, so p1 and p3 end up being the same number.
p1 = p2;
p2 = p3;
p3 = p1;

//prints num1, num2, and num3, which haven't changed since the last printout.
cout << num1 << " " << num2 << " " << num3 << endl;

//prints out newly assigned values of p1, p2, and p3
cout << p1 << " " << p2 << " " << p3 << endl;
cout << p1 << " " << p2 << " " << p3 << endl;

//sets p2 to 22
p2 = 22;

//sets p1 to 22
p1 = p2;

//sets p2 to 2
p2 = p3;

//sets p3 to 22
p3 = p1;

//prints num1, num2, and num3, which haven't changed since the last printout.
cout << num1 << " " << num2 << " " << num3 << endl;

//prints out 22, 2, and 22, The newly assigned values of p1, p2, and p3
cout << p1 << " " << p2 << " " << p3 << endl;

//prints it out again
cout << p1 << " " << p2 << " " << p3 << endl;
}

```

Question 2: Summary

I implemented a recursive function that returns the value of $n +$ a function call to $n - 1$.

Question 2: The Output

```
Enter an integer number:12
12
78

Process finished with exit code 0
|
```

Question 2: The Code

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

//recursive function
int n_sum(int n) {

    //checks to see if number is equal to 0
    if (n != 0)
        return n + n_sum(n - 1);
    //if n is 0, just return 0
    else
        return n;
}

int main() {

    //declares int N
    int N;
    //prompts user to initialize int N
    cout << "Enter an integer number: ";

    //accepts user value for int N
    cin >> N;

    //prints out call to recursive function n_sum
    cout << n_sum(N) << endl;
}
```

Question 3: Summary

This recursively computes the Fibonacci sequence by taking in integer values, initialized to values 0 and 1, and then continually adding them together until they reach the threshold value of N, set by the user. Then the program continually prints out the value to the console.

Question 1: The Output

```
Enter an integer value: 5
5
11235
Process finished with exit code 0
|
```

Question 3: The Code

```
#include <iostream>

using namespace std;

int fibonacci_sequence(int n) {

    //declares and initializes num1 and num2 values to 0 and 1 so that program doesn't hang
    int num1 = 0, num2 = 1;

    //stops loop if n is smaller than threshold
    if (n < 1)
        return n;

    //loops through to user input value
    for (int i = 1; i < n; i++) {
        //prints value
        printf("%d", num2, " ");
        //sets next to sum of first two integers, per the fibonacci sequence
        int next = num1 + num2;
        //sets smaller num1 to larger num2
        num1 = num2;
        //sets num2 to larger number
        num2 = next;
    }
}

int main() {

    //declares integer N
    int N;

    //prompts user to set integer N to value
    cout << "Enter an integer value: ";
    //sets integer N to user value
    cin >> N;

    //prints out call to fibonacci sequence
    cout << fibonacci_sequence(N);
}
```

Question 4: Summary

This program was modified to explain what each of the printed values represented.

Question 1: The Output

```
The value that is stored: 5
The value in memory that count is stored: 0x61ff18
Another value in memory that count is stored: 0x61ff18
Calling the value at the location using an asterisk: 5

Process finished with exit code 0
```

Question 4: The Code

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

int main()
{
    int count = 5;
    int* pCount = &count;

    cout << "The value that is stored: " << count << endl;
    cout << "The value in memory that count is stored: " << &count << endl;
    cout << "Another value in memory that count is stored: " << pCount << endl;
    cout << "Calling the value at the location using an asterisk: " << *pCount << endl;
}
```
