

CISC 181 sections 010-018 (Conrad) Midterm I March 15, 2004

Name _____

Section (circle one, (2 pts)): 010 013 016
 011 014 017
 012 015 018

Circle one:

Freshman Sophomore Junior Senior Other

General Instructions

- DO NOT WRITE YOUR NAME ON ANY PAGE EXCEPT THIS ONE!
- You have 50 minutes
- **Pace Yourself!!!!**

Pay attention to the point values. When there are 10 minutes left, skim through and be sure you have at least written *something* for the questions that are worth many points.

- Read *all* the directions *carefully* on each problem.
- Good luck.

1. (30 pts) Write a complete C++ program to solve the following problem, including

- an opening comment (*don't* put your name in the comment! -2 pts if you do!)
- all necessary “stuff” that goes before the main program
- a full main program complete with comments

Problem Statement: Ask the user of the program to input three non-negative numbers. Determine which of the three numbers is largest. Then, scale the numbers down by dividing all three by the largest one. Then output all three numbers.

Examples:

- For input of 3 9 10, the output should be 0.3 0.9 1
- For input of 45 180 90, the output should be 0.25 1 0.5

Your program should provide appropriate prompts to the user, and should also label the output appropriately and neatly.

There are two error conditions you should check for. If either of these error conditions occurs, print an error message, and terminate the program immediately.

- If any of the numbers input is negative.
- If the largest of the three numbers is zero. (Note that this can only occur if all three are zero.)

Extra space in case you need it

2. (10 pts) Two of your lab exercises involved taking a number apart, digit by digit, by using division and the modulus operator. This questions tests your knowledge of that concept, plus your understanding of recursion.

Write a C++ function (just the function, not a complete C++ program) with the following prototype:

```
int sumOfDigits(int x);
```

The function must use recursion (not iteration) to return the sum of the digits in a number. Negative values are treated as if they were positive. Examples:

- If you pass in 345, the function returns 12, since $3 + 4 + 5 = 12$
- If you pass in 12, the function returns 3, since $1 + 2 = 3$
- If you pass in -12, the function still returns 3 (treat negative numbers as if they were positive).
- If you pass in 7, the function returns 7.

Clues:

- If the number passed in is negative, multiply it by -1 right away.
- For any number between 0 and 9, the function returns the same number you passed in. This is your base case.
- For the other case, consider the following:
 - get the final digit using the mod operator
 - add that to the result of a recursive call on the number you get when you strip off the final digit (you can do that by dividing by 10).

Extra space in case you need it

3. (10 pts) One of your lab exercises required you to write two different versions of a function that calculates the n th Fibonacci number, one using recursion, and the other using iteration. This question is a variation on that exercise.

Write a function (only the function definition, not a complete C++ program) that will take one integer parameter called x and return an integer result.

The function should be called `isFibonacci`. It should:

- return 1 if the number x is a Fibonacci number
- return 0 if x is *not* a Fibonacci number.

For example, your function should return 1 if you pass in any of the following values: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, etc.

Your function should return 0, if you pass in any negative number, or any non Fibonacci number such as 4, 6, 7, 9, 10, 11, 12, 14, etc.

If it is helpful, you may assume that there is already a function available for your use with the prototype:

```
int fibonacci(int n);
```

that returns the n th Fibonacci number in the sequence. For example:

```
fibonacci(0) returns 0
fibonacci(1) returns 1
fibonacci(2) returns 1
fibonacci(3) returns 2
fibonacci(4) returns 3
fibonacci(5) returns 5
fibonacci(6) returns 8
etc...
```

You can assume that this implementation of Fibonacci is efficient and correct, so if it makes things easier, you may use it in your answer (i.e. call it from inside your `isFibonacci` function.)

Extra space in case you need it

13. Number conversions:

(a) (3 pts) Convert 68 from decimal to binary

(b) (3 pts) Convert 1F7C from hexadecimal to binary

(c) (3 pts) Convert 3E from hexadecimal to decimal

14. (1 pts) Write an example of a preprocessor directive

15. Describe the functions of each of the following parts of a computer.

(a) (2 pts) CPU

(b) (2 pts) RAM

16. When calling a function, a one of the things that gets pushed on the stack as part of the call frame is the current value of the program counter (PC).

(a) (2 pts) What is the program counter? (describe briefly)

(b) (2 pts) Why it necessary to push its value on the stack (as part of the call frame) when calling a function?

17. Suppose I start with an empty stack, and then do the following sequence of operations:

- push 19
- push 26
- push 13
- push -4
- pop
- *pop*
- push 11
- push 16
- pop
- push 12

(a) (2 pts) What value will be returned as a result of the *second* pop operation (the one in *italics*?)

(b) (3 pts) Draw what the stack will look like at the end of the sequence of operations?


```
// helloThere.cpp

#include <iostream>

using std::endl;
using std::cout;

int helloThere(int x, int y)
{
    int a = 5;

    cout << "Hello There!" << endl;

    return (y + a * x);
}

int main(void)
{
    int a = 2;
    int b = 3;

    int c = -4;

    c = a + 2 * b;

    cout << "c=" << c << endl;

    b = helloThere(a, c);

    cout << "a=" << a;
    cout << " b=" << b << endl;
    cout << " c=" << c << endl;

    return 0;
}
```

Total Points: 100