COP 3502 (Computer Science I) Test #1
Date: 9/13/2013

First Name: _____ , Last Name: _____

Questions 1 - 3 concern the following set of #defines and struct definitions:

```
#define NAMESIZE 20
#define TEAMSIZE 100

typedef struct {
    char first[SIZE];
    char last[SIZE];
    int PID;
    double gpa;
} studentT;

typedef struct {
    studentT captain;
    studentT members[TEAMSIZE];
} teamT;
```

1) (8 pts) With justification, determine the number of bytes allocated in the malloc statement shown below. Assume that a char is stored in 1 byte an int is stored in 4 bytes and a double is stored in 8 bytes.

```
teamT* IMTeams = malloc(sizeof(teamT)*50);
```

_____

2) (8 pts) Draw a detailed picture of memory right after the malloc statement in question 1 has executed. (Note: Since it's unreasonable to draw arrays of large sizes, to indicate an array, draw dividing lines for the first two elements, followed by ... and a dividing line for the last box and label the first and last indexes.)

3) (25 pts) Write a studentcmp function that takes in pointers to two studentT's and compares them using the following criteria:

a) If the first student (pointed to by ptrA) has a lower GPA than the second, a negative integer is returned. If the first student has a higher GPA than the second, a positive integer is returned.

b) If the two GPAs are equal, then ties are broken by strcmp in the string library applied to the last names. (In these cases, if the last names are different, just return what strcmp does.)

c) If the two last names are the same, then those ties are broken by strcmp applied to the first names. (In these cases, if the first names are different, just return what strcmp does.)

d) Break the tie in this case by returning the difference in PIDs.

```
#include <string.h>

int studentcmp(const studentT* ptrA, const studentT* ptrB) {









}
```

4) (8 pts) What is the return value of the function call f(3, 5), where f is the function shown below?

```
int f(int a, int b) {
    if (a == 0) return 1;
    return b*f(a-1,b-1)/a;
}
```

_____

5) (9 pts) Recalling that n! (read n factorial) is simply the product of the first n positive integers, express f(a,b), where f is the function defined in question 4 in terms of the factorial function only.

_____

6) (8 pts) In class you were shown code to execute fast modular exponentiation. In this code, the type "long long" was used, which stores a 64 bit signed integer. Why was it necessary to use this type even though the actual parameters to the fast modular exponentiation function could all be stored in the regular int type?

7) (10 pts) What does the recursive function below compute? (In order to receive full credit, your answer has to be very specific and accurate.)

```
int f(int* array, int low, int high) {
    if (low == high) return array[low];
    int mid = (low+high)/2;
    int left = f(array, low, mid);
    int right = f(array, mid+1, high);
    if (left > right) return left;
    return right;
}
```

8) (15 pts) The Catalan numbers, C_k, can be defined as follows: $C_0 = 1$ and $C_n = \frac{2(2n+1)}{n+2} C_{n-1}$, for all integers n > 0. (Note: The definition above produces only integers for Catalan numbers.) Write a recursive function that computes the n^{th} Catalan number. Do not worry about overflow issues for the purposes of this question. You will get full credit if you use recursion and if your code is "theoretically" correct. **(But do worry about integer division and its effects.)**

```
int catalan(int n) {









}
```

9) (6 pts) The function below is supposed to execute a floodfill on an integer board with dimensions SIZE x SIZE, starting at the array indexes x and y. (Assume all supporting code exists and does what we might expect it to.) used[i][j] stores 1 if location (i,j) on the board has been previously filled and 0 otherwise. What is the maximum number of recursive calls that might be called directly by one call of the fill function?

```
void fill(int board[][SIZE], int used[][SIZE], int x, int y) {

    used[x][y] = 1;
    // Some code here.

    for (dx=-1; dx<=1; dx++)
        for (dy=-1; dy<=1; dy++)
            if (inbounds(x+dx, y+dy) && !used[x+dx][y+dy])
                fill(board, used, x+dx, y+dy);
}
```

_____

10) (3 pts) Tomorrow, UCF's football team faces the Penn State Nittany Lions in an away game. In which state will the game take place?

_____

Scratch Page - Please clearly mark any work on this page you would like graded.