Honors Computer Science I (COP 3502) Exam #1 Solutions
Date: 2/5/2026

1) (8 pts) Write a function that takes in integer array, `numbers`, and the length of the array, `n`, and dynamically allocates a new integer array of the same size, storing the values from numbers in this new array in reverse order, returning a pointer to this newly created array. Fill in the code for the function below:

```
int* reverseCopy(int* numbers, int n) {

    int* res = calloc(n, sizeof(int));   // 3 pts
    for (int i=0; i<n; i++)              // 1 pt
        res[i] = numbers[n-1-i];         // 3 pts
    return res;                          // 1 pt
}
```

2) (8 pts) A program that solves a very difficult problem on an input of size n runs in $O(n^n)$ time. For n = 4, the program takes 2^{-15} seconds to complete. How long, **in seconds,** would we expect the algorithm to take when run on an input with size n = 8?

Let $T(n) = cn^n$ be the runtime of the program on an input of size n, where c is a constant.

$T(4) = c(4^4) = 2^{-15}$ sec, thus, $c = \frac{2^{-15}}{4^4} sec = \frac{2^{-15}}{(2^2)^4} sec = \frac{2^{-15}}{2^8} sec = 2^{-23} sec$

We must solve for T(8). $T(8) = c(8^8)$ sec = $(2^{-23}$ sec) x $(2^3)^8 = 2^{-23}$ x 2^{24} sec = **2 seconds**

Grading: 1 pt for setting up T(n), 3 pts for solving for c,
1 pt plug in n = 8, 1 pt to get to 2 seconds

3) (15 pts) Consider the following struct:

```
#define MAXLEN 50

typedef struct UCFStudent {
     char* first;
     char* last;
     int UCFID;
} UCFStudent;
```

Write a function that takes in an integer, n, representing the number of students to read data for, and then reads, from standard input, n students' worth of data (first name, followed by a space, followed by the last name, followed by a space, followed by the UCFID of the student, one student per line), creates an **array of pointers to UCFStudent**, fills in each of the n structs with the appropriate information, and returns a pointer to the array that was just created. You may assume that no student's name is more than 50 characters long, but when you allocate space to store their first and last name in a UCFStudent struct, make sure to allocate ONLY the space you need. Fill in the function below. Note: n has already been read in, your code should have n scanf statements, each of which read in three things. Declare any local variables you see fit.

```
UCFStudent** readStudentData(int n) {

    UCFStudent** res = calloc(n, sizeof(UCFStudent*));   // 2 pts
    for (int i=0; i<n; i++) {                            // 1 pt

        res[i] = malloc(sizeof(UCFStudent));             // 2 pts

        char readF[MAXLEN+1], readL[MAXLEN+1];           // 1 pt

        // 2 pts
        scanf("%s%s%d", readF, readL, &( res[i]->UCFID));

        // 2 pts
        res[i]->first = calloc(strlen(readF)+1, sizeof(char));

        // 1 pt
        strcpy(res[i]->first, readF);

         // 2 pts
        res[i]->last = calloc(strlen(readL)+1, sizeof(char));

        // 1 pt
        strcpy(res[i]->last, readL);
    }

    return res;      // 1 pt
}
```

4) (8 pts) With proof, answer the following questions about the function below:

```
int f(int n) {

    if (n<2) return 0;

    for (int i=2; i*i<=n; i++)
        if (n%i == 0)
            return 0;

     return 1;
}
```

(a) (2 pts) What is the best case run time (Big-Omega), in terms of n, of this function?

Ω(1) – this is because the first loop iteration could trigger the if and return 0 with constant amount of work. **(Grading: 1 pt answer 1 pt explanation)**

(b) (4 pts) What is the worst case run time (Big-Oh), in terms of n, of this function?

$O(\sqrt{n})$, This is because the worst case is when the if never triggers and $i^2 > n$. Solving this equation gives us $i > \sqrt{n}$, this is roughly equal to the most number of times the loop could run. The work in the loop is constant time. **(Grading: 2 pts, 2 pts explanation)**

(c) (2 pts) What type of inputs cause the worst case run-time for this function?

Prime numbers, since they aren't divisible by any integers in between 2 and their square root. Squares of primes also cause this worst case behavior.
(Grading: 2 pts all or nothing no explanation needed)

5) (8 pts) An arithmetic sequence is a sequence of terms such that the difference between each pair of successive terms is the same. For example, 13, 17, 21, 25, 29, 33 is an arithmetic sequence starting with 13 (first term), with a common difference of 4, and 6 terms. Write a **recursive function** that takes in the first term of an arithmetic sequence, its common difference, and the number of terms in the sequence and returns the sum of the sequence. To earn full credit, your function must NOT use any knowledge of the arithmetic sequence sum formula. Fill in the function below (the formal parameters should be self-explanatory!):

```
// Pre-condition: nTerms >= 0.
int sumArithSequence(int firstTerm, int diff, int nTerms) {

    if(nTerms == 0) return 0;
    return firstTerm + sumArithSequence(firstTerm+diff, diff, nTerms-1);

}
```

Grading: 2 pts BC, 1 pt return, 1 pt firstTerm+, 1 pt rec call, 1 pt for each parameter.

6) (12 pts) One way to derive the formula for $\sum_{i=1}^{n} i$, is to use the following statement

$$\sum_{i=1}^{n}(2i+1) = \sum_{i=1}^{n}[(i+1)^2 - i^2]$$

and note that

$$\sum_{i=1}^{n}(i+1)^2 = \sum_{i=2}^{n+1} i^2$$

Use these two facts to prove a closed-form formula for $\sum_{i=1}^{n} i$. In your proof you may assume that you know the sum of a constant.

$$\sum_{i=1}^{n}(2i+1) = \sum_{i=1}^{n}[(i+1)^2 - i^2]$$
$$= \left(\sum_{i=1}^{n}(i+1)^2\right) - \left(\sum_{i=1}^{n} i^2\right)$$
$$= \left(\sum_{i=2}^{n+1} i^2\right) - \left(1 + \sum_{i=2}^{n} i^2\right)$$
$$= (n+1)^2 + \left(\sum_{i=2}^{n} i^2\right) - 1 - \left(\sum_{i=2}^{n} i^2\right)$$
$$= (n+1)^2 - 1 = n^2 + 2n + 1 - 1 = n^2 + 2n$$

Now, use this fact to solve for $\sum_{i=1}^{n} i$:

$$\sum_{i=1}^{n}(2i+1) = n^2 + 2n$$
$$\left(\sum_{i=1}^{n}(2i)\right) + \left(\sum_{i=1}^{n} 1\right) = n^2 + 2n$$
$$\left(2\sum_{i=1}^{n} i\right) + n = n^2 + 2n$$
$$\left(2\sum_{i=1}^{n} i\right) = n^2 + n$$

$$\sum_{i=1}^{n} i = \frac{n^2+n}{2} = \frac{n(n+1)}{2}$$

Grading: 6 pts to show sum of (2i+1), 6 pts to use that to get formula for sum of i, partial up to grader.

7) (15 pts) Complete the function below so that it takes in a grid of characters(`grid`), a 0-based row index in the grid(`myr`), a 0-based column index in the grid(`myc`), the number of rows in the grid(`r`), and the number of columns in the grid(`c`), and completes a floodfill from row `myr`, column `myc`, on the connected region from that location containing the '-' character. Your floodfill should change each of the '-' characters in this connected component to '*' **and return the total number of characters changed**. A region of '-' characters is in the same connected component if all pairs of characters can be reached from each other by going up, down, left or right to other '-' characters only. Please use the DX, DY arrays provided. If the character at row myr and column myc isn't '-', or if it's out of bounds, no action should be taken except returning 0. Assume no inbounds function exists, so write the corresponding code within the fill function.

```
const int DX[] = {-1,0,0,1};
const int DY[] = {0,-1,1,0};

int fill(char** grid, int myr, int myc, int r, int c) {

    if(myr >= r || myr < 0 || myc >= c || myc < 0) // 3 pts
        return 0;

    if (grid[myr][myc] != '-') return 0;  // 3 pts

    grid[myr][myc] = '*';                 // 1 pt
    int res = 1;                          // 1 pt

    for (int i=0; i<4; i++)               // 1 pt
        res += fill(grid, myr+DX[i], myc+DY[i], r, c); // 5 pts

    return res;                           // 1 pt
}
```

8) (1 pt) The upcoming Olympic games are titled the "Milano Cortina 2026" Olympics. With what Pepperidge Farm cookie does the city Milano share a name?

Milano (Give to All)