CSE 122– HW 2 Rubric

Student Name: **Noah Cherry**

Grader Name: **Kyle Buchmiller**

Questions	Score	Total
Code Requirements		
1.) Fibonacci Error Term		
a. Named fib_error.c b. Writes a function to find the max n before overflow • unsigned int max_n(); • This will be about 93 on a 64 bit, much lower on a 32 bit • Should have found the equation: $n = \frac{\log(\sqrt{5}*ULONG_{MAX})}{\log(\frac{1+\sqrt{5}}{2})} - 1$ -2 forgot to subtract 1	4	6
c. Write a function to find the fibonnaci term • double fib(unsigned n) • Uses the formula given in pdf	5	5
d. Writes a function to store fibannoci values • unsigned long* fib_array(unsigned n) • Uses for loops NOT recursion • Used the formula $F_n = F_{n-1} + F_{n-2}$	6	6
e. Prints out the array of numbers one-per-line in Main (**2pts**) f. Find Relative error between array values and equation one. • If they differ, print an error message stating which term differs, how much by and the percentage error between them.	6	6
g. Capture the output for the program to a file • fib_error.out -4 no fib_error.out • 64 bit will look like the sample, 32 bit shouldn't have errors assuming your machine was 32 bit. When I changed 'lim' to 93, it printed the numbers out correctly.	4	8
h. No Memory Leaks • Test with valgrind	5	5

CSE 122– HW 2 Rubric

3.) Fibonacci Nth Term		

• Takes in a command a line argument (-n) (**2pts**)
• Checks to see if n would overflow (**5pts**)
  ○ Gives an error statement if it returns -1 and exits.
• No memory Leaks (**3pts**)
  ○ Check with valgrind.
-1 should print only the number being sought after
	9	10

4.) Code Analyzing (**9 points each**)		
I. Exact count for each line matters, T(n) can be summed differently Look at attached pdf solutions • Cost times (**5pts**) -3 c2 and c3 are off C2: n+1 C3: n • T(n) is correct (**3pts**) -1 would be correct if your costs were. • Graph is Linear (**2pts**) -2 no graph	4	10
II. Exact count for each line matters, T(n) can be summed differently Look at attached pdf solutions • Cost times (**5pts**) -4 C2: n+1 C3: n(n+1) C4: n^2 • T(n) is correct (**3pts**) -1 would be correct if your costs were • Graph is Quadratic (**2pts**) -2 no graph	3	10
III. Exact count for each line matters, T(n) can be summed differently Look at attached pdf solutions • Cost times (**5pts**) -4 C2: n+1 C3: $n(n^2+1)$ C4: $n(n^2)$ • T(n) is correct (**3pts**) -1	3	10

CSE 122– HW 2 Rubric

• Graph is Cubic (**2pts**) -2 no graph		
IV. Exact count for each line matters, T(n) can be summed differently Look at attached pdf solutions • Cost times (**5pts**) -4 C2: 1 C3: n(n+1) / 2 C4: (n-1)n / 2 • T(n) is correct (**3pts**) -3 no T(n) equation • Graph is Quadratic (**2pts**) -2 no graph	1	10
V. EXTRA CREDIT The student must get exactly the following (**2 extra pts**): ➓ c1 = 1 ➓ c2 = n+1 ➓ c3 = (1/6)(n)(2n^2-3n+7) The student must get answers similar to the following (**5 extra pts**): ➓ c4 = (1/60)(n)(n+1)(2n+1)(3n^2+3n+4) ➓ c5 = (1/60)(n)(6n^4+15n^3+20n^2+15n-56) REQUIRED • Graph is Cubic (**3pts**) -3 no graph	0	3
VI. EXTRA CREDIT The student must get exactly the following (**1 extra pt**): ➓ c1 = 1 ➓ c2 = n The student must get answers similar to the following	0	3

CSE 122– HW 2 Rubric

(**6 extra pts**): ➓ c3 = (1/6)(n)(n+1)(2n+1) ➓ c4 = (1/6)(n)($2n^2$+3n-5) ➓ c5 = (1/2)(n^2+n-2) ➓ c6 = (n(n+1))/2 REQUIRED • Graph is Cubic (**3pts**) -3 no graph		
Subjective Criteria – This week these are discounted from the total.		
• Used a Makefile. (2 points) -2 no makefile • Tarball named correctly (2 points) • Has doxygen comments (2 points) • General code quality (coding style, spacing, etc.) (2 points)	6	8
• +4 for a good foo.out. If you would have went to 100000 on foo4, it would have been +5.	4	
	60	100

CSE 122– HW 2 Rubric

Comments:

Decent work overall, but forgetting some files really hurt your grade. Be careful to remember to include everything required!