Student 1 **Last** Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 1 **First** Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 1 ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 2 **Last** Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 2 **First** Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 2 ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INCLUDE** this page**,** the next page, and your solution in your submission

* Your solution **MUST BE TYPED**
* Each written solution **MUST** begin on its own page
* Write your solution with enough detail that the average student could generate a program but at a high enough level so that people outside the class can understand what is to be done
  + Focus on what you want to do and not so much on how it should be done as this will help you come up with a more coherent description
* You **MUST** number each one of your steps starting at 1
* Your **should NOT** include any of the following in your solution
  + Mention of types or conversions between types
    - This is becauses types are a programming construct and are not related to problem solving
  + What files you need to include
    - This is because included files have to do with programming and not problem solving
  + Getting input and displaying output
    - All programs have to do this so we won’t be mentioning it
  + Mention of builtin functions
    - For example if you needed to find the square root of x you would say either take the square root of x or and not sqrt(x)
* You **should** include the following in your solution
  + Any mathematical formulas that you are not given
    - If you are given a mathematical formula in the prompt you can say “I use the formula provided in the prompt to calculate …”
  + Definitions of functions
    - This will allow you to break the problem into manageable parts and you can refer to those functions in your main solution
    - When defining a function do as we do when programming but leave out the type declarations
    - Call a function as we do when programming
* You **may** use the following in your solution
  + Mathematical operators (+, -, \*, etc) but they should be used sparingly if you choose to use them as we are more interested in what you are doing in your program than how you are accomplishing it.
  + Declare and assign values to variables to be able to easily reference them in your solution but this should probably be done sparingly as your answer should not look like code

Problem Name: **change.c**

1. For each domination from largest to smallest
   1. Number of bills of that denomination is amount owed / denomination
   2. Amount owed = remainder of the previous division (1.a)

Problem Name: **grade\_need.c**

Problem Name: Loan Cost

1. Monthly interest rate is annual\_interest\_rate / 12
2. Find the monthly payments using the given formula
   1. Making sure to use monthly interest rate
3. Total paid for the loan is number of payments \* monthly payment
4. The cost of the loan is total paid for the loan - amount borrowed

Problem Name: Complex Multiplication

1. Let the first complex number be
2. Let the first second complex number be
3. The real part of the product =
4. The imaginary part of the prouduct =