Name: **Exam #1** ________ **/ 100 pts.**

Instructions: This exam is to be taken in silence, without notes, books, or electronic devices (including "smart" watches or earbuds). The time limit to complete it is the duration of the class period (1 hour and 15 minutes). Answer the following questions and problems, trying to be as clear and as accurate as possible. Take your time to read the statements carefully before trying to answer them. If you need more space, write on the back of your test page and indicate it clearly. When writing code, make sure your special punctuation characters are legible, and your lowercase and uppercase letters are easy to distinguish. As usual, every statement or series of statement is assumed to be in a valid class and method, and you can use the `C.W()` and `C.WL()` abbreviations.

____ **/ 20 pts.** **Problem 1** This problem is about binary search. Assume `arrayEx` is an array of `int` containing the following values:

```
-----------------------------------------------------------------------
| 15 | 25 | 30 | 45 | 60 | 75 | 90 | 105 | 115 | 165 | 180 | 190 | 210 |
-----------------------------------------------------------------------
```

and consider the following algorithm:

```
bool fsf = false;
int tar = 105;
int sta = 0;
int end = arrayEx.Length - 1;
int mid, cur;
while (sta <= end && !fsf)
{
    mid = (sta + end) / 2;
    cur = arrayEx[mid];
    if (tar == cur){
        fsf = true;
    }
    else if (tar > cur){
        sta = mid + 1;
    }
    else {
        end = mid - 1;
    }
}
```

Complete the following table, giving the value of `sta`, `end`, `mid`, `fsf` and `cur` before the **`while`** loop is executed, after it has executed one time, two times, etc. If the value is not set at this point, write "undef", and if the loop stops before the Xth iteration, simply cross out the Xth iteration. Report the value even if it did not change.

	Before loop	After 1 iteration	After 2 iterations	After 3 iterations	After 4 iterations
`sta`					
`end`					
`mid`					
`fsf`					
`cur`					

____ / 30 pts. **Problem 2** The goal of this problem is to help you implement and use a simple BankAccount class with methods and properties capable of throwing exceptions.

1. We start by implementing our class. Answer the following, assuming we are inside a **public class** BankAccount{…} class definition:

   (a) Write an automatic property for an Owner string.

   (b) Write a property for a decimal Balance such that an ArgumentException exception will be thrown if an argument less than 0 is used to set its value.

   (c) Write a Deposit method that takes a decimal as input and increases Balance by this amount. An ArgumentException exception should be thrown if the argument is not strictly positive.

   (d) Write a Withdraw method that takes a decimal as input and decreases Balance by this amount. An ArgumentException exception should be thrown if the argument is not strictly positive, or if it is greater than Balance.

2. We now use our class. Write statements (to be inserted in a `Main` method) that
   (a) Create a `BankAccount` object called `account`, using the default constructor.

   (b) Ask the user to enter their name and sets `account`'s `Owner` using their answer.

   (c) Ask the user to enter their initial deposit, and sets `account`'s `balance` using the answer. The exception thrown if the user enters a `string` that is not a number, and the exception thrown by `Balance` if the number entered is a negative value should be caught[1]. Your code should then display "Thanks for banking with us!" regardless of whether an exception was thrown.

[1] To earn bonus points, catch both exceptions separately, that is, display a different message.

____ / 25 pts. **Problem 3** *To receive credit for this problem, answer the following questions using a 2-dimensional jagged array* ***or*** *a 2-dimensional rectangular array (your choice,* ***but be consistent****).*

1. Write statements that declare a 2-dimensional array with at least 2 rows containing values 1 through 6.

2. Write statements that display the values stored in a 2-dimensional array called `arTest`.

3. Write statements displaying "Ranked" if the sum of the values stored in each row is greater than the sum of the values in the preceding row in the `arTest` array. For example, the following rectangular and jagged arrays are both ranked, as 4 < 5 < 6.

```
2 1 1                                4 0
3 1 1                                3 1 1
5 1 0                                2 2 1 1
```

___ / 15 pts. **Problem 4** This problem is about references and generic data type (for the last one).

1. Consider the following method:

   ```
   static void NameChange(ref string nameP, string newnameP, out string
    ↪  oldnameP){
     oldnameP = nameP;
     nameP = newnameP;
   }
   ```

   Assume given a `string` `name` variable containing a value. Write a short program (possibly declaring additional variables) that

   (a) Asks the user their new name,
   (b) Calls the `NameChange` method with appropriate arguments,
   (c) Displays the new name and the old name.

2. The following method would *not* compile. Explain why.

   ```
   static void Test(int a, out int b){
       if (b < 0) { b = a; }
       else { b++;}
   }
   ```

3. Write a `SafeLength` method that takes an input an array of *any type* and returns its length if the argument is not **`null`**, -1 otherwise.

____ / 20 pts. **Problem 5** This problem is about arrays.

1. Given an `int` `arrayP` array and an `int` `target` variable, write the code that performs a linear search: it should display "Found it!" if `target` is in `arrayP`.

2. Given an `int` `matrixP` *2-dimensional rectangular* array and an `int` `target` variable, write the code that performs a linear search: it should display "Found it!" if `target` is in `arrayP`.