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.

____ / **35 pts.** **Problem 1** Consider the implementation of "simple" dictionary `SDictionary` shared on page 6. Remember that, for example, `"Bob"[0]` is `'B'` and use the correspondence below between characters and their integer representation to help you (i.e., `(int)'B'` is 66):

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90

1. Fill the `table` array below after the following have been performed:

```
SDictionary friends = new SDictionary(11);
friends.Add("Bob", null);
friends.Add("Pete", null);
friends.Add("Mary", null);
friends.Add("Lora", null);
```

   Indicate only the keys' first letter, don't copy the values (set to **`null`** for simplicity).

0	1	2	3	4	5	6	7	8	9	10

2. What would happen if `friends.Add("Lora", null);` was executed again? Is it what is expected from a dictionary?

3. Write a `ToString` method for the `SDictionary` class, that returns a `string` containing all the keys and values stored in the dictionary.

4. What would happen if we were to try to insert 12 elements in our `friends` object?

5. Consider the following `Delete` method:

```
public bool Delete(string keyP)
{
    int count = 0;
    int index = GetIndex(keyP, count);
    bool found = false;
    while (table[index] != null && !found)
    {
        if (table[index].Key.Equals(keyP))
        {
            found = true;
            table[index] = null;
        }
        count++;
        index = GetIndex(keyP, count);
    }
    return found;
}
```

Complete the series of instructions below such that `demo.Delete(error)` would return **`false`** even though the string `error` *is* the key of a value present in the `demo` dictionary object.

```
class Program{
    static void Main(){
    SDictionary demo = new SDictionary(    ); // Complete it.

    string error =                             // Fill me









    Console.WriteLine($"{error} was in demo:
↪ {demo.Delete(error)}.");
    }
}
```

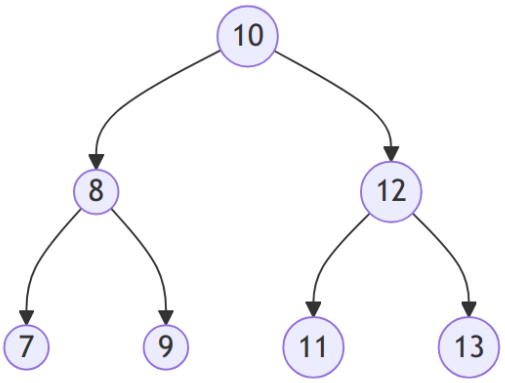
____ / 30 pts. **Problem 2** This problem is about binary search trees.

1. Starting with a single, empty binary search tree, draw the tree after 12, 6, 8, and 9 have been inserted.

2. Starting with a single, empty binary search tree, draw the tree after 8, 12, 6 and 9 have been inserted.

3. Consider the following method and tree, where `root` is the node carrying the value 10, and suppose that we are executing `root = RotaterightChild(root);`:

```
private Node RotaterightChild(Node nTop)
{
    Node nRight = nTop.right;
    nTop.right = nRight.left;
    nRight.left = nTop;
    return nRight;
}
```



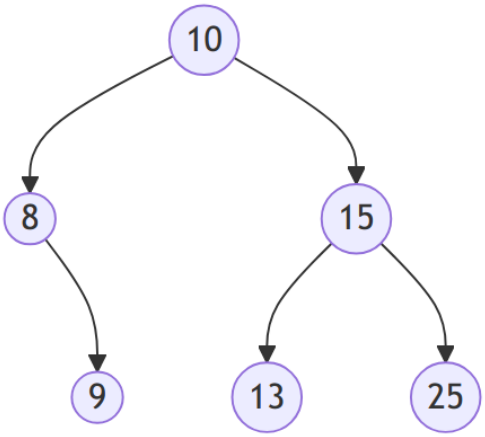
(a) Indicate on the drawing above which nodes are `root`, `nTop`, `nRight` and `nRight.left` *before* the `RotaterightChild` method is done executing.
(b) Draw the tree after `RotaterightChild` is finished.

____ / 15 pts. **Problem 3** Write the `DoubleleftChild` method that transform the tree on the left-hand side below into the tree on the right-hand side below.


```
private Node DoubleleftChild(Node nodeP)
{
```

```
 return
}
```

____ / 15 pts. **Problem 4** Consider the following AVL tree:



1. Give its *inorder* traversal.

2. Give an order in which the values could have been inserted (for example, even if this is incorrect, "9, 13, 25, …") to obtain this tree.

3. Draw next to the drawing the tree obtained after 10 was removed.

____ / 15 pts. **Problem 5** This problem is about min-priority queue, i.e., queues where the most important element is the one with the lowest priority. Draw the priority queue after

1. Inserting values with priority 10, 2, 5, 7,
2. Removing the most important element,
3. Inserting values with priority 3, 12,

(you can either draw the queue after each step, or only at the very end)

1. If the queue is implemented using an array.

2. If the queue is implemented using a binary heap.

```
public class SDictionary
{

  private class Cell
  {
    public string Value { get; set; }
    public string Key { get; set; }

    public Cell(string keyP, string valueP)
    {
      Key = keyP;
      Value = valueP;
    }

    public override string ToString( )
    {
      return Key + ":" + Value;
    }
  }

  private Cell[] table;

  public SDictionary(int size = 31)
  {
    table = new Cell[size];
  }

  public int GetIndex(string keyP, int countP)
  {
      return ((int)(keyP[0]) + countP) % table.Length;
  }

  public void Add(string keyP, string valueP)
  {
      int count = 0;
      int index = GetIndex(keyP, count);
      while (table[index] != null)
      {
          count++;
          index = GetIndex(keyP, count);
      }
      table[index] = new Cell(keyP, valueP);
  }
}
```