به نام او	**ساختمان‌های داده**	

نام و نام خانوادگی:

شماره دانشجویی:

توجه:

- **تاریخ تحویل:** ۲۴ / ۷ / ۱۴۰۰
- اگر برای جواب‌دادن به سوالی نیاز به پیش‌فرضی دارید، فرض خود را صریحا نوشته و با توجه به فرض خود به ارائه جواب بپردازید.
- به هیچ وجه تمرینی را از دیگران کپی نکنید. درصورت مشاهده تقلب و کپی در تمرینات، نمره هردو طرف **صفر** درنظر گرفته می‌شود.

دانشکده
مهندسی کامپیوتر

مدرس:
دکتر حسین رحمانی

	تمرین سری اول	**نمره**

۱	show complexity of this program (array is sorted).
	(explain your answer)
	
	int BinarySearch(int array[], int n, int value){
	int left = 1, right = n, middle;
	while(right >= left) {
	middle = (left + right)/2;
	if (array[middle] == value)
	return middle;
	if (array[middle] > value)
	right = middle - 1;
	else
	left = middle + 1;
	}
	return -1;
	}
		۲۵

۲	calculate T(n) runtime of these recursion functions with master method (explain your answer) a) $T(n) = 3T\left(\frac{n}{2}\right) + n$ b) $T(n) = T\left(\frac{n}{2}\right) + 2^n$ c) $T(n) = 3T\left(\frac{n}{2}\right) + n^2$ d) $T(n) = 8T\left(\frac{n}{2}\right) + n^3$	۲۰

١٥	Apply insertion sort to the following arrays and write all steps: a) [12,34,2, -5,13, -6,4,3] b) [8,3,23, -4,16,22] c) [9, -8,4,56,23,11,6,42]	٣
١٠	Suppose we have an array, [1, 3, 6, 7, 10], and a new element to add, x = 4. If we use the insertion sort to add x to our array, how many comparisons will it take and what will be the final index of x in the array?	٤

٣٠	Consider the program below and then answer the questions posed:
	
	a) Implement the insertFirst method that inserts a new node with the given data key at the front of the linked list. Assume the list is not empty.
	b) Implement the delete method that deletes a node with the given data key from the related LinkList object, and returns the Node containing the key. Assume the list is not empty. If the key does not exist in the list, null should be returned and no action should be taken.
	c) Predict the printed results of the main method in LinkListApp
	
	class Node {
	public int iData; // data item (key)
	public Node next; // next node in list
	
	public Node(int id)
	{ // constructor
	iData = id;
	}
	}
	class LinkList
	{
	private Node first; // reference to first node on list
	
	public LinkList()
	{ // constructor
	first = null;
	}
		٥

```
  public Node find(int key)
 { // find the node with a given key
     Node current = first; // start at the first node
     while (current != null && current.iData != key)
       current = current.next; // go to next node
     return current;
   }

   public void displayList()
{ // display the list
     for (Node current = first; current != null; current = current.next)
       System.out.println(current.iData); // print data
   }

   public void insertFirst(int key)
 { // insert a node at the front of the list
     // See question (a)...
   }

   public Node delete(int key)
{ // delete the node with a given key
     // See question (b)...
   }

}
class LinkListApp
{
   public static void main(String[] args)
{
     LinkList theList = new LinkList(); // create a list
     theList.insertFirst(22);
     theList.insertFirst(44);
     theList.insertFirst(66);
     Node d = theList.delete(44);
     d = theList.delete(88);
     theList.displayList(); // display list
   } // end main()
}
```

موفق باشید.