

שאלה 1, סעיף א (8 נק')

`public static boolean lexLT(String s1, String s2) {`

<code>for (int i=0; i&lt;s1.length() &amp;&amp; i&lt;s2.length(); i=i+1) {</code>
<code>    if (s1.charAt(i) &lt; s2.charAt(i))</code>
<code>        return true;</code>
<code>    if (s1.charAt(i) &gt; s2.charAt(i))</code>
<code>        return false;</code>
<code>}</code>
<code>return s1.length() &lt; s2.length();</code>

`}`

שאלה 1, סעיף ב (12 נק')

`public static int fibIter(int n) {`

<code>int a=0, b=1, tmp;</code>
<code>for(int i=0; i&lt;n; i=i+1) {</code>
<code>    tmp=b;</code>
<code>    b=a+b;</code>
<code>    a=tmp;</code>
<code>}</code>
<code>return a;</code>

`}`

5	4	3	2	1
: יו"פ 1				

שאלה 2 (20 נק') (מומלץ להוסיף פונקצית עזר – אך אם אינכם רוצים אז רשמו את תשובתכם בשורות הנתונות)

```
public static int change(int sum, int[] coins){
```

one solution:	a better solution:
return change(sum, coins, 0);	int[][] memo = new int[sum+1][coins.length];
	return change(sum, coins, 0, memo);

```
}
```

**one solution:**

הוסף פרמטרים במידת הצורך כאן:

```
public static int change(int sum, int[] coins, int first) {
```

if (sum == 0) return 1;
else if (sum>0 && first < coins.length)
return
change(sum-coins[first], coins, first) + change(sum, coins, first+1);
else return 0;

```
}
```

**a better solution:**

הוסף פרמטרים במידת הצורך כאן:

```
public static int change(int sum, int[] coins, int first, int[][] memo) {
```

if (sum == 0) return 1;
else if (sum>0 && first < coins.length){
if (memo[sum][first]==0)
memo[sum][first] = change(sum-coins[first], coins, first, memo) +
change(sum, coins, first+1, memo);
return memo[sum][first];
}
else return 0;

שאלה 3, סעיף א (8 נק') יש להשלים (במידת הצורך) את שורת ההכרזה של הרכיב

`public abstract class Shape {`

`private Color color;`

`public Shape(Color color) {`

`this.color = color;`

`}`

`public abstract double getPerimeter();`

`public abstract double getArea();`

`public Color getColor() {`

`return color;`

`}`

`public abstract void move(Point p);`

`public abstract boolean contains(Point p);`

`}`

שאלה 3, סעיף ב (5 נק') יש להשלים (במידת הצורך) את שורת ההכרזה של הרכיב

**public class Circle extends Shape {**

<code>private Point center;</code>
<code>private double radius;</code>
<code>public Circle(Point center, double radius, Color color) {</code>
<code>    super(color);</code>
<code>    this.center = new Point(center);</code>
<code>    this.radius = radius;</code>
<code>}</code>
<code>public Point getCenter() {</code>
<code>    return new Point(center);</code>
<code>}</code>
<code>public double getRadius() {</code>
<code>    return radius;</code>
<code>}</code>
<code>public double getPerimeter() {</code>
<code>    return 2 * Math.PI * radius;</code>
<code>}</code>
<code>public double getArea() {</code>
<code>    return Math.PI * radius * radius;</code>
<code>}</code>
<code>public void move(Point p) {</code>
<code>    center.move(p);</code>
<code>}</code>

המשך שאלה 3, סעיף ב (5 נק')

```
public boolean contains(Point p) {  
    return (center.distance(p) < radius);  
}
```

} // Circle

שאלה 3, סעיף ג (12 נק') יש להשלים (במידת הצורך) את שורת ההכרזה של הרכיב

**public abstract class Polygon extends Shape {**

<code>protected Point[] points;</code>
<code>public Polygon(Point[] points, Color color) {</code>
<code>    super(color);</code>
<code>    this.points = new Point[points.length];</code>
<code>    for (int i = 0; i &lt; points.length; i = i+1)</code>
<code>        this.points[i] = new Point(points[i]);</code>
<code>    }</code>
<code>public double getPerimeter() {</code>
<code>    double res = 0;</code>
<code>    for (int i = 0; i &lt; points.length; i = i+1)</code>
<code>        res += points[i].distance(points[(i+1)%points.length]);</code>
<code>    return res;</code>
<code>    }</code>
<code>public void move(Point p) {</code>
<code>    for (int i = 0; i &lt; points.length; i=i+1) {</code>
<code>        points[i].move(p);</code>
<code>    }</code>
<code>    }</code>

`}//Polygon`

שאלה 4 (15 נק')

סעיף א. (7 נקודות)

השלימו

```
public class QueueAsCircularList implements Queue {
    private CircularLinkedList cl;
    public QueueAsCircularList () {
        cl = new CircularLinkedList ();
    }
    public boolean isEmpty() { //אין צורך לממש }

public void enqueue(Object o) {
    cl.addLast(o);
}

public Object dequeue(){
    return cl.removeFirst();
}
} // QueueAsCircularList
```

שאלה 4, סעיף ב. (8 נקודות) השלימו

```

public class CircularLinkedList {
    private Link pointer; // מצביע אחד לשימוש כרצונכם

    public CircularLinkedList() {pointer = null; }

    השלימו ערך מוחזר, שם ופרמטר (אם צריך)
    public void addLast (Object data) {
        if (size() < 2) { אין צורך לממש }
        else {


|                                                         |
|---------------------------------------------------------|
| Link newLink = new Link(data);                          |
| newLink.next=pointer.next;                              |
| pointer.next = newLink;                                 |
| pointer=pointer.next; // pointer points to last element |
|                                                         |
|                                                         |
|                                                         |
|                                                         |


        }
    } //else
}

    השלימו ערך מוחזר, שם ופרמטר (אם צריך)
    public Object removeFirst (_____) {
        if (size() < 2) { אין צורך לממש }
        else {


|                                 |
|---------------------------------|
| Object ret = pointer.next.data; |
| pointer.next=pointer.next.next; |
| return ret;                     |
|                                 |
|                                 |
|                                 |
|                                 |


        }
    } //else
}
} // CircularLinkedList

```



שאלה 5, סעיף א (5 נק')

במחלקה IntBinaryTree :

```
public int sum() {
```

<code>int ans = 0;</code>
<code>if (root != null)</code>
<code>    ans = root.sum();</code>
<code>return ans;</code>

```
}
```

במחלקה IntBinaryNode :

```
public int sum() {
```

<code>int ans = data;</code>
<code>if (left != null)</code>
<code>    ans = ans + left.sum();</code>
<code>if (right != null)</code>
<code>    ans = ans + right.sum();</code>
<code>return ans;</code>

```
}
```

שאלה 5, סעיף ב (7 נק')

במחלקה IntBinaryTree :

```
public boolean isHeap() {
```

```
    boolean ans = true;
    if (root != null)
        ans = root.isHeap();
    return ans;
```

```
}
```

במחלקה IntBinaryNode :

```
public boolean isHeap() {
```

```
    boolean ans = true;
    if (left != null)
        ans = (data > left.data);
    if ((ans) && (right != null))
        ans = (data > right.data) && right.isHeap();
    if ((ans) && (left != null))
        ans = left.isHeap();
    return ans;
```

```
}
```

שאלה 5, סעיף ג (8 נק')

במחלקה IntBinaryTree :

```
public int maxHeapSum() {
```

```
int ans = 0;
if (root != null)
    ans = root.maxHeapSum();
return ans;
```

```
}
```

במחלקה IntBinaryNode :

```
public int maxHeapSum() {
```

```
if (isHeap())
    return sum();

int leftMaxHeapSum = 0;
int rightMaxHeapSum = 0;

if (left != null)
    leftMaxHeapSum = left.maxHeapSum();
if (right != null)
    rightMaxHeapSum = right.maxHeapSum();
return Math.max(leftMaxHeapSum, rightMaxHeapSum);
```

```
}
```

בהצלחה !