

**שאלה 1 (14 נקודות)**

```
public static boolean arrayGT(int[] x, int[] y) {
    boolean answer;
```

if (x.length==0)
answer=false;
else {
answer = true;
int max = x[0];
for (int i=1; i<x.length; i=i+1)
if (x[i]>max)
max=x[i];
for (int j=0; j<y.length && answer; j=j+1)
if (max <= y[j])
answer = false;
}

```
return answer;
```

```
}
```

לשימוש הבוחנים						
7	6	5	4	3	2	1
		סופי				

**שאלה 2 (הקף בעיגול מצד שמאל תשובה אחת לכל היותר עבור כל סעיף) (11 נקודות)**

- א. `arr != null`  לא נשמרת  נשמרת
- ב. `n >= 0`  לא נשמרת  נשמרת
- ג. `i < n`  לא נשמרת  נשמרת
- ד. `i < n - 1`  לא נשמרת  נשמרת
- ה. `i < n - 2`  לא נשמרת  נשמרת
- ו. ערכי מערך `arr` בתחום `[0, i+1]` ממוינים בסדר יורד.  לא נשמרת  נשמרת
- ז. ערכי מערך `arr` בתחום `[0, i]` ממוינים בסדר יורד.  לא נשמרת  נשמרת
- ח. ערכי מערך `arr` בתחום `[0, i-1]` ממוינים בסדר יורד.  לא נשמרת  נשמרת
- ט. כל ערכי מערך `arr` בתחום `[i+1, n-1]` קטנים או שווים לערכי המערך בתחום `[0, i]`.  לא נשמרת  נשמרת
- י. כל ערכי מערך `arr` בתחום `[i, n-1]` קטנים או שווים לערכי המערך בתחום `[0, i-1]`.  לא נשמרת  נשמרת

**שאלה 3 (20 נקודות)**

```
public static boolean rec(String s1, String s2) {
    boolean answer;
```

<code>if (s1.length() == 0)</code>
<code>    answer = true;</code>
<code>else if (s2.length() == 0)</code>
<code>    answer = false;</code>
<code>else if (s1.charAt(0) == s2.charAt(0))</code>
<code>    answer = rec(s1.substring(1), s2.substring(1));</code>
<code>else</code>
<code>    answer = rec(s1, s2.substring(1));</code>

```
return answer;
```

```
}
```

## שאלה 4 (13 נקודות)

```
public class DerivativeThresholdFilter implements Filter {  
    private Variable var;  
    private ThresholdFilter filter;  
  
    public DerivativeThresholdFilter(Variable var,  
    ThresholdFilter filter) {  
        this.var = var;  
        this.filter = filter;  
    }  
  
    public boolean accept(Object obj) {  
        boolean ans = false;  
  
        if (obj instanceof Expression) {  
            ans =  
filter.accept(((Expression) obj).derivative(var));  
        }  
  
        return ans;  
    }  
}
```

**שאלה 5 (12 נקודות)**

```
public Set union(Set other) {
```

Set result = new MySet();
Iterator thisIterator = iterator();
while (thisIterator.hasNext())
result.add(thisIterator.next());
Iterator otherIterator = other.iterator();
while (otherIterator.hasNext())
result.add(otherIterator.next());
return result;

```
}
```

**שאלה 6**

א. (10 נקודות)

```
public void removeOddItems() {
```

Link current=this;
while (current!=null && current.next!=null) {
current.next=current.next.next;
current=current.next;
}

```
}
```

ב. (10 נקודות)

```
public boolean isSimpleCircular() {
```

```
    boolean ans = false;
```

```
    Link curr = this.next;
```

```
    while (curr!=null & !ans) {
```

```
        if (curr.next == this)
```

```
            ans = true;
```

```
            curr = curr.next;
```

```
    }
```

```
    return ans;
```

```
}
```

שאלה 7 (10 נקודות)

```
public static int findMissingNum(int n) {
    Scanner sc = new Scanner(System.in);
    int answer;
    int num;
```

```
    answer = n*(n+1)/2;
```

```
    for(int i=1; i<n; i=i+1) {
        num = sc.nextInt();
```

```
        answer = answer - num;
```

```
    }
```

```
    return answer;
```

```
}
```