

## המשך תרגיל כיתה 1 – רקורסיה

### שאלה 1

כתבו פונקציה רקורסיבית לחישוב אצרת:

`int fact (n)`

### שאלה 2

כתוב פונקציה רקורסיבית המחשבת את איבר  $n$  של סדרה פיבונצ'י:

$(a_1 = 1, a_2 = 1, a_{n+2} = a_{n+1} + a_n)$  1, 1, 2, 3, 5, 8, 13, 21, 34, ...

`int fibonacci(int num)`

### שאלה 3

כתוב פונקציה רקורסיבית המקבלת מחרוזת והמדפיסה אותה בסדר הפוך:

`public static void printBackward(String s)`

### שאלה 4

נתונה פונקציה רקורסיבית:

```
public static String mystery( String s){
    int len = s.length();
    if (len<=1) return s;
    String a = s.substring(0,len/2);
    String b = s.substring(len/2, len);
    return mystery(b)+mystery(a);
}
```

מה תהיה התוצאה של הרצת הפונקציה:

```
System.out.println("mystery: "+mystery("אבגדהוזחט"));
```

ענה בלי הפלת הפונקציה.

## פתרון

```
public static int factor(int n){
    if (n > 1){
        int naz = factor(n-1);
        return n*naz;
    }
    else{
        return 1;
    }
}

// fibonacci numbers: 1 1 2 3 5 8 13 21 34 55 . . .
public static int fibonacci(int num)
{
    if (num == 0 || num == 1)
    {
        return 1;
    }
    else
    {
        return (fibonacci(num - 1) + fibonacci(num - 2));
    }
}
```

```

// print reverse string
public static void printBackward(String s){
    int size = s.length();
    if (size>0){
        System.out.print(s.substring(size-1, size));
        printBackward(s.substring(0, size-1));
    }
}

public static int reverseNumber(int number){
    int result = 0;;
    while (number != 0){
        result = result*10 + number%10;
        number = number/10;
    }
    return result;
}

public static int reverseNumberRecurs(int number, int result){
    if(number == 0){
        return result; //base (exit condition)
    }
    result = result*10 + number%10;
    return reverseNumberRecurs(number/10,result); //recursive call
}

public static int summ(int n){ // calculate 1+2+...+n
    if (n > 1){
        return n+summ(n-1);
    }
    else{
        return 1;
    }
}

public static void triangle(int n){ // print triangle
    char x = '*';
    if (n > 1){
        line(n);
        System.out.println();
        triangle(n-1);
    }
    else{
        System.out.println(x);
    }
}

public static void line(int n){ // print line
    char x = '*';
    if (n>1){
        line(n-1);
    }
    System.out.print(x);
}

public static int power(int n, int m){ // power: n^m
    if (m==0){
        return 1;
    }
    return n*power(n,m-1);
}

```

```
}  
public static int maxValue(int[] arr, int beg){  
    if (beg == 0){  
        return arr[0];  
    }  
    int x = maxValue(arr, beg-1);  
    return x>arr[beg] ? x : arr[beg];  
}
```