

פתרון לשאלון 97104 – אלגוריתמיקה ותכנות – מועד א' קיץ 24

שאלה 1 (12 נקודות)

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int twoDigitNumCounter= 0 ;
        int sumOfeven = 0;

        int num2;
        for (int i=0;i<520;i++) {

            if (num1 >=10 && num1 <=99)
                twoDigitNumCounter++;
            if (num2 >=10 && num2 <=99)
                twoDigitNumCounter++;
            if (num1%2==0)
                sumOfeven+=num1;
            if (num2%2==0)
                sumOfeven+=num2;
        }
        System.out.println("sum of even numbers: " + sumOfeven);
        System.out.println("number of two digit numbers:" + twoDigitNumCounter);
    }
}
```

שאלה 2 (12 נקודות)

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String str;
        int sameStartEndCounter = 0;
        str = in.nextLine();
        int len = str.length();
        while (len%2==0) {
            if (str.contains("Z")) // str.indexOf('Z')!=-1
                System.out.println("string contains Z");

            if (str.charAt(0)==str.charAt(len-1))
                sameStartEndCounter++;
            str = in.nextLine();
            len = str.length();
        }
        System.out.println("Number of strings with same start and end: "+ sameStartEndCounter);
    }
}
```

```
import java.util.Random;

public class Main {
    public static void main(String[] args) {
        Random rand = new Random();
        int[] arr = new int[100];
        for (int i=0;i<100;i++)
            arr[i] = 0;
        for (int i=0;i<40;i++)
            arr[10 + rand.nextInt(90)]++;
        int max = arr[10];
        int maxIndex = 0;
        for (int i=10;i<100;i++) {
            if (arr[i] > max) {
                max = arr[i];
                maxIndex = i;
            }
            if (arr[i]== 0)
                System.out.println(i+ " not genetated");
        }
        System.out.println(maxIndex + " Generated maximum times");
    }
}
```

שאלה 4 (12 נקודות)

```
public static boolean isTripple(int[] arr) {
    int threeSum = 0;
    for (int i=0; i<arr.length; i++) {
        threeSum+=arr[i++];
        threeSum+=arr[i++];
        threeSum+=arr[i];
        if (threeSum != 0)
            return false;
    }
    return true;
}
```

```
public static boolean isTripple(int[] arr) {
    int threeSum = 0;
    for (int i=0; i<arr.length; i+=3) {
        threeSum+=arr[i];
        threeSum+=arr[i+1];
        threeSum+=arr[i+2];
        if (threeSum != 0)
            return false;
    }
    return true;
}
```

שאלה 5 (12 נקודות)

סעיף א (5 נקודות)

```
public static boolean isPerfectK(int[] arr, int k) {

    if (arr.length<k)
        return false;
    int sumFive = 0;
    for (int i=0; i<k; i++)
        sumFive+=arr[i];
    return ((sumFive %k )==0);
}
```

סעיף ב (5 נקודות)

```
public static boolean isSuperPerfect(int[] arr) {
    boolean flag = true;
    for (int i=1; i<arr.length; i++)
        if (flag)
            flag = isPerfectK(arr, i);
    return flag;
}
```

```
public static boolean isSuperPerfect(int[] arr)
{
    for (int i=1; i<arr.length; i++)
        if (! isPerfectK(arr, i)) return false;
    return true;
}
```

סעיף ג (2 נקודות)

סיבוכיות סעיף א : $O(n)$ כאשר n -מספר האיברים במערך , אפשר גם $o(k)$
סיבוכיות סעיף ב : $O(n^2)$ כאשר n מספר האיברים במערך.

שאלה 6 (12 נקודות)

סעיף א (4 נקודות)

1. דוגמה למספר num שיתן 3 בהרצה של one(num,7)

7707

2. הפעולה סופרת ומחזירה כמה פעמים מופיעה הספרה dig במספר num.

סעיף ב (4 נקודות)

1.

```
int[] arr = {24, 34783, 1245, 68, 468, 9445};
two(arr,4)
i=0 arr[i] = 24 one(24,4) , res=1
i=1 arr[i] = 34783 , one(34783,4) res=2
i=2 arr[i] = 1245 , one(1245,4) res=3
i=3 arr[i] = 68 , one(68,4) res=3
i=4 arr[i] = 468, one(468,4) res=4
i=5 arr[i] = 9445 , one(9445,4) res=6
res=6
```

2. two מחזירה את כמות החזרות של הספרה dig בכלל איברי המערך arr.

סעיף ד (4 נקודות)

1. דוגמה למערך שthree(arr) יחזיר 6

```
Arr = {6,6,6,6,4,44,66};
```

2. three תחזיר את הספרה שמופיע הכי הרבה באיברי המערך arr (סכום המופעים של הספרה בכל האיברים של המערך)

שאלה 7 (12 נקודות)

סעיף א (4 נקודות)

"123a3A"

תחזיר 4

סעיף ב (3 נקודות)

דוגמה למחרוזת שתדפיס %%

"123456"

סעיף ג (3 נקודות)

דוגמה למחרוזת שתחזיר 2 :

"Aa12Aa"

עבור מחרוזת זו הפעולה תדפיס :

%0#1?4#5?%

סעיף ד (2 נקודות)

הפעולה תחזיר את כמות התווים במחרוזת str שהם לא אותיות באנגלית, היא תדפיס תמיד % בהתחלה ואכ בכל מקום שיש אות קטנה תדפיס את האינדקס שלה ואכ '?' ובכל מקום שיש אות גדולה תדפיס את האינדקס שלה ואכ '#', ובסוף הריצה תדפיס %

שאלה 8 (15 נקודות)

סעיף א (6 נקודות)

```
public static boolean isPalindrom(int[] arr) {
    int len = arr.length;
    for (int i=0;i<len/2;i++)
        if (arr[i]!=arr[len-1-i])
            return false;
    return true;
}
```

סעיף ב (6 נקודות)

```
public static boolean isEvenPalindrom(int[] arr) {
    int len = arr.length;
    int[] evenArr = new int[len];
    int evenLength = 0;
    for (int i=0;i<len;i++)
        if (arr[i]%2==0)
            evenArr[evenLength++] = arr[i];
    int[] finalEvenArr = new int[evenLength];
    for (int i=0;i<evenLength;i++)
        finalEvenArr[i]=evenArr[i];

    return isPalindrom(finalEvenArr);
}
```

סעיף ג (3 נקודות)
שניהם $O(n)$

פתרון נוסף:

```
public static boolean isPalindrom(int[] arr) {
    int i = 0, j = arr.length-1;
    while(i < j) {
        if (arr[i] != arr[j])
            return false;
        i++;
        j--;
    }
    return true;
}
```

סעיף ד (6 נקודות)

```
public static boolean isEvenPalindrom(int[] arr) {
    int i = 0, j = arr.length-1;
    while(i < j) {
        if(a[i]%2!=0) i++;
        else
            if(a[j]%2!=0) j--;
        else
            {
                if (arr[i]!=arr[j]) return false;
                i++;
                j--;
            }
    }
    return true;
}
```

שאלה 9 (15 נקודות)

סעיף א (5 נקודות)

```
public static int index(int[] arr, int pos, int num) {
    for (int i = pos+1; i<arr.length;i++)
        if (arr[i]==num)
            return i;
    return -1;
}
```

סעיף ב (5 נקודות)

```
public static boolean exist(int[] arr1, int[] arr2) {
    int prevIndex = -1;
    int currIndex = 0;
    for (int i=0;i<arr2.length;i++) {
        currIndex = index(arr1,prevIndex,arr2[i]);
        if (currIndex == -1)
            return false;
        prevIndex = currIndex;
    }
    return true;
}
```

סעיף ג (5 נקודות)

```
public static boolean areEqual(int[] arr1,int index,int[]arr2) {
    int len1=arr1.length;
    int len2=arr2.length;
    if (len1<(len2+index))
        return false;
    for (int i=0;i<len2;i++)
        if (arr1[index+i]!=arr2[i])
            return false;
    return true;
}
public static boolean isSubArray(int[] arr1, int[]
arr2) {
    int prevIndex=0;
    int currIndex=0;
    for (int i=0;i<arr1.length-arr2.length;i++) {
        currIndex = index(arr1,prevIndex,arr2[0]);
        if (currIndex== -1)
            return false;
        if (areEqual(arr1,currIndex,arr2))
            return true;
        prevIndex = currIndex+1;
    }
    return false;
}
```

פתרון אחר

```
public static boolean isSubArray(int[] arr1, int[] arr2) {
    for (int i=0;i<arr1.length-arr2.length;i++) {
        boolean flag = true;
        for( int j = 0; j<arr2.length; j++)
            if(arr1[i+j] != arr2[j])
                flag = false;
        if(flag) return true;
    }
    return false;
}
```

שאלה 10 (15 נקודות)

סעיף א (5 נקודות)

```
int[] arr = {5,7,12,15,21,26,40,51,71,84};  
what(arr, 2, 6)  
temp = arr[2]=12, arr[2]=arr[6]=40  
arr[2]=temp=12  
start=3,end=5  
arr[3] = 26, a[5]=15  
start=4,end=4  
end.  
arr = {5,7,40,26,21,15,12,51,71,84};
```

סעיף ב (2 נקודות)

הפעולה מעבירה הופכת את סדר האברים ב arr החל ממיקום start, ועד מיקום end.

סעיף ג (4 נקודות)

```
why(arr, 4)  
int[] arr = {5,7,12,15,21,26,40,51,71,84}  
s=4%10=4  
what(arr,0,9) => arr = {84,71,51,40,26,21,15,12,7,5}  
what(arr,0,3)-=>arr = {40,51,71,84,26,21,15,12,7,5}  
what(arr,4,9) => arr = {40,51,71,84,5,7,12,15,21,26}
```

סעיף ד (2 נקודות)

S=0/10/20/30

כל כפולה של 10

כי אז s בשורה הראשונה יהיה 0. ואז בעצם יש הפעלה :

what(arr,0,9) – היפוך מיקום אברי המערך –
what(arr,0,-1) – לא ישנה כלום –
what(arr,0,9) – היפוך מיקום אברי המערך –

כלומר המערך יתהפך ויוחזר אב למצבו המקורי.

סעיף ה (2 נקודות)

הפעולה עושה סיבוב מעגלי ימינה s פעמים!

שאלה 11 (15 נקודות)

סעיף א (7 נקודות)

1. mystery(24,15) - כל זוג עם כמות זהה של ספרות וסכום ספרות זהה יחזיר 2 -
2. mystery(600,22) - ממוצע ספרות זהה אבל כמות ספרות שונה יחזיר 3 -
3. mystery(12,234) - כל זוג עם סכום שונה וכמות ספרות שונה יחזיר 4 -
4. לא קיים מספר תלת ספרתי שעבורו לא נמצא זוג שיחזיר 2, כי תמיד המספר השלילי למספר שבחרנו ייקיים
5. עבור 100 לא קיים זוג שיחזיר 3, אין מספר עם מספר ספרות שונה וממוצע זהה

סעיף ב (6 נקודות)

```
int[] arr = {3458,-45,7681,-875,6,13571,43};  
  
3458 - num2=20, num1=4  
  
-45 - num2=9,num1=2 => pos1=0,pos2=0  
  
7681 - num2 =22,num1=4 => pos1 = 0, pos2 = 2  
  
....  
  
13571 - num2 = 17,num2 =5 => pos1=5,pos2 =2  
  
false
```

סעיף ג (2 נקודות)

מחזירה true אם יש במערך איבר בעל מספר הספרות הגבוה ביותר וגם סכום הספרות שלו גבוה ביותר. אחרת
יחזיר false

שאלה 12 (11 נקודות)

סעיף א (8 נקודות)

```
public static boolean superEven(int num) {
    int cnt=0;
    while (num>0) {
        if (num%2!=0)
            return false;
        cnt++;
        num/=10;
    }
    return (cnt%2==0);
}

public static boolean superEven(int[] arr) {
    int len = arr.length;
    if (len%2!=0)
        return false;
    int cnt=0;
    for (int i=0;i<len;i++)
        if (superEven(arr[i]))
            cnt++;
    return (cnt>len/2);
}

public static boolean superEven(int[][] mat) {
    if (mat.length%2!=0)
        return false;
    for (int i=0;i<mat.length;i++)
        if (!superEven(mat[i]))
            return false;
    return true;
}
```

סעיף ב (3 נקודות)

$O(n*m)$

$n*m$ גודל המערך הדו מימדי

d - מספר הספרות נחשיב כקבוע למשל מקסימום 7 לכן לא נמצא בסיבוכיות או $O(\log_{10} N)$

שאלה 13 (11 נקודות)

סעיף א (3 נקודות)

```
public class Magor {  
  
    private String majName;  
    private int code;  
}
```

```
public class College {  
  
    private String name;  
    private String city;  
    private Magor[] magorArr;  
}
```

```
public class Network {  
    private College[] colleges;  
}
```

סעיף ב (4 נקודות)

```
public void printCities(String city) {  
    int cnt=0;  
    for (int i=0;i<this.colleges.length;i++)  
        if (this.colleges[i].getCity().equals(city)) {  
            cnt++;  
            System.out.println(colleges[i].getName());  
        }  
    if (cnt==0)  
        System.out.println("No collages!!");  
}
```

סעיף ג (4 נקודות)

```
public String[] canOpenNewMagor(int code) {  
    int cnt=0;  
    String res="";  
    boolean found;  
    for (int i=0;i<this.colleges.length;i++) {  
        found=false;  
        if (this.colleges[i].getMagorArr().length<10)  
            for (int j =0;j<this.colleges[i].getMagorArr().length;j++)  
                if (this.colleges[i].getMagorArr()[j].equals(code))  
                    found=true;  
        res = res + " " + this.colleges[i].getName();  
    }  
    return res.split("\\s+");  
}
```

שאלה 14 (11 נקודות)

סעיף א (5 נקודות)

```
int[] brr = {15,5,3,16,10,8};
what(brr,0,brr.length-1)
begin=0,end=5
what(brr,1,5)
what(brr,2,5)
what(brr,3,5)
what(brr,4,5)
what(brr,5,5)
brr[4]=brr[4]-brr[5] = 10-8=2
brr[3]=brr[3]-brr[4] = 16-2=14
brr[2] = brr[2]-brr[3]= 3-14=-11
brr[1] = brr[1]-brr[2] = 5-(-11)=16
brr[0]=brr[0]-brr[1] = 15-16=-1
```

```
brr = {-1,16,-11,14,2,8}
```

2. מטרת הפעולה what בכל איבר של arr החל מהאיבר begin – יהיה חיסור ואכ סיכום לסירוגין של האיברים עד end

סעיף ב (6 נקודות)

```
int[] crr = {6,5,7,8,10,3};
```

הפעולה משנה את איברי המערך למערך הפרשים בין איברים סמוכים. החל מ מקום begin, ועד end, האיבר האחרון end יהיה ללא שינוי