

## סיכום ג'אוה סמסטר א'

### פירוק מספר חיובי לספרותיו וחישוב שארית חלוקה

תרגיל חישוב עודף:

2. חנניה מהמכולת מעוניין באלגוריתם אשר יעזור לו להחזיר עודף במטבעות בצורה היעילה ביותר, כלומר במספר המטבעות הקטן ביותר.
- בהנחה כי חנניה יכול להחזיר עודף אך ורק במטבעות של 1 ש, של 5 ש ושל 10 ש, כתבו אלגוריתם המקבל כקלט את הסכום שחנניה צריך להחזיר כעודף (מספר שלם), ומציג כפלט:
  - א. את מספר המטבעות שיחזיר חנניה מכל סוג.
  - ב. את מספר המטבעות הכולל שיחזיר חנניה.למשל עבור הקלט 18 יתקבל הפלט:
  - א. 1 ש: 3, 5 ש: 1, 10 ש: 1
  - ב. 5 מטבעות.ישמו את האלגוריתם בשפת Java.

#### פתרון:

האלגוריתם להחזרת העודף במספר המטבעות הקטן ביותר

1. קלוט את סכום העודף לתוך change
2. חשב  $change / 10$  והשם לתוך tens
3. חשב  $change \% 10$  והשם לתוך change
4. חשב  $change / 5$  והשם לתוך fifths
5. חשב  $change \% 5$  והשם לתוך change
6. חשב  $change / 1$  והשם לתוך shekels
7. הצג כפלט את tens, fifths, shekels
8. הצג כפלט את סה"כ המטבעות  $((tens + fifths + shekels))$

#### התוכנית:

```
import java.util.Scanner;

public class targil2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);
        int change,tens,fifths,shekels;
        System.out.println("Enter change please: ");
        change = in.nextInt();
        tens = change /10;
        change = change % 10;
        fifths = change /5;
        change = change % 5;
        shekels = change /1 ;
        System.out.println("tens: "+tens + " fifths: "+fifths + " shekels: "+shekels);
        System.out.println("The sum of currency is : "+ (tens + fifths + shekels));
        in.close();
    }
}
```

## פלינדרום מספרים בעזרת רוורס:

```
//פונקציה שמקבלת מספר שלם ובודקת אם המספר הוא מספר פלינדרום/
public class PalindromNum {
    public static boolean isPalindrome(int num) {
        int reverse = 0; //הצהרה על משתנה שהופך את המספר
        int temp = num; //שמירה של המספר שהתקבל
        while(num > 0) { //לולאה ההופכת את המספר שהתקבל
            reverse = reverse * 10 + num % 10;
            num = num /10;
        }
        if(reverse == temp) {
            return true;
        }
        return false;
    }

    public static void main(String[] args) {

        System.out.println(isPalindrome(433353334));
    }
}
```

## הגרלה של מספרים (random) ובדיקה האם המספרים זוגיים:

### שאלה 7.13

כתבו קטע תוכנית אשר מקבל כקלט מספר שלם ומוטו. התוכנית תגריל ומוטו מספרים בתחום 10-50 ותבדוק כמה מספרים מבין המספרים שהוגרלו הם זוגיים.

```
import java.util.Random;
import java.util.Scanner;
public class targil7_13 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);
        Random rnd = new Random();
        int num , i,even,rndNum;
        even =0;
        System.out.println("Enter number:");
        num = in.nextInt();
        for(i= 0;i<num;i++) {
            rndNum = rnd.nextInt(40) + 10;
        }
    }
}
```

```

        if(rndNum % 2 == 0) {
            even++;
        }
    }
    System.out.println( even + " Even numbers");
}

```

## הגרלה של מספרים (random) תלת ספרתיים.

### שאלה 7.14

כתבו קטע תוכנית שיגריל 10 מספרים תלת-ספרתיים. התוכנית תחשב ותדפיס את סכום המספרים האי-זוגיים בלבד מבין המספרים שהוגרלו.

```

public class targil7_14 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Random rnd = new Random();
        int i, sumOfNotEven, rndNum;
        sumOfNotEven = 0;

        for(i= 0; i<10; i++) {
            rndNum = rnd.nextInt(899) + 100;
            if(rndNum % 2 == 1) {
                sumOfNotEven = rndNum + sumOfNotEven ;
            }
        }

        System.out.println( "The sum of NOT Even numbers are:" + sumOfNotEven);
    }

}

```

## • חישוב עצרת.

### שאלה 7.17

פתחו אלגוריתם שמקבל כקלט מספר חיובי שלם n, מחשב את n!, ומציג את הערך שחושב כפלט. ישמו את האלגוריתם בשפת Java. להזכירכם: n! היא מכפלת המספרים מ-1 עד n, כלומר:  $n! = 1 \cdot 2 \cdot \dots \cdot n$ .

- 1. אתחל את factorial ל 1
- 2. אתחל את n ל 1
- 3. קלוט מספר לתוך n
- 4. בצע n-1 פעמים
- 5. הכפל את factorial ב i
- 6. הצג את factorial

```

import java.util.Scanner;
public class targil_7_17 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);
        int factorial, n, i;
        factorial = 1;
        System.out.println("Enter number:");
        n = in.nextInt();

        for(i = 1; i <= n; i++) {

```

```

        factorial = factorial * i;
    }
    System.out.println(factorial);
}

```

## החלפת אות באות שאחריה (char) ושרשור לתוך String

פתחו אלגוריתם אשר הקלט שלו הוא סדרת תווים המהווה מילה באנגלית, ומסתיימת בזקיף '\*'. הפלט שלו הוא מילת הקלט והיא מוצפנת באופן הבא: כל אות במילה מוחלפת באות העוקבת לה "בצורה מעגלית" ב-א"ב האנגלי (כלומר, כל אות מלבד האות Z מוחלפת באות העוקבת לה, והאות Z מוחלפת באות A). למשל עבור הקלט ZEBRA\* הפלט הוא AFCSB. ישמו את האלגוריתם בשפת Java.

1. קלוט אות לתוך letter
2. כל עוד letter שונה מ \* בצע
  - 2.1 אם letter שווה ל – Z
    - 2.1.1 השם את A לתוך letter
    - 2.1.2 הצג את letter
  - 2.2 אחרת השם letter ועוד 1 ב letter
    - 2.2.1 הצג את letter
  - 2.3 קלוט אות לתוך letter

```

public static void main(String[] args) {
    // TODO Auto-generated method stub
    Scanner in = new Scanner(System.in);
    char letter;
    System.out.println("Enter a letter: ");
    letter = in.next().charAt(0);
    String str = "";
    while(letter != '*' )
    {
        if(letter == 'Z')
        {
            letter = 'A';
            str+=letter;
        }
        else
        {
            letter = (char) (letter + 1) ;
            str+=letter;
        }

        letter = in.next().charAt(0);
    }
    System.out.println(str);
}

```

## תוכנית המדמה הטלת קוביות עד שתוצאת ההטלה היא "שש-בש" ומציגה את מספר ההטלות שהתבצעו

טורניר השש-בש מתחיל, אך איבדתם את הקוביות! פתחו אלגוריתם אשר מדמה הטלת שתי קוביות ומציג את תוצאות ההטלה. האלגוריתם מדמה את ההטלות עד שתוצאת ההטלה היא "שש-בש", כלומר צמד המספרים 5 ו-6 או 6 ו-5. לסיום האלגוריתם מציג כפלט את מספר ההטלות שהתבצעו עד שהתקבלה התוצאה שש-בש. ישמו את האלגוריתם בשפת Java.

### פתרון:

1. אתחל את counter ב 0
1. הגרל מספר בין 1 ל 6 והשם ב dice1
2. הגרל מספר בין 1 ל 6 והשם ב dice2
3. כל עוד dice1 לא שווה ל 6 ו dice2 לא שווה ל 5 או dice1 לא שווה ל 5 ו dice2 לא שווה ל 6
- 3.1 הגדל ב 1 את counter
- 3.2 הגרל מספר בין 1 ל 6 והשם ב dice1
- 3.3 הגרל מספר בין 1 ל 6 והשם ב dice2
4. הצג את counter

```
import java.util.Random;

public class targil_7_32 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Random rnd = new Random();
        int dice1,dice2,counter;
        counter = 0;
        dice1 = rnd.nextInt(6) + 1;
        dice2 = rnd.nextInt(6) + 1;
        while((dice1!=5 && dice2 !=6) || (dice1!=6 && dice2 !=5))
        {
            counter++;
            dice1 = rnd.nextInt(6) + 1;
            dice2 = rnd.nextInt(6) + 1;
        }
        System.out.println("The dice was thrown " + counter + " times");
    }

}
```

## תוכנית להצגת חזקה ללא math.pow

פתחו אלגוריתם אשר הקלט שלו הוא מספר חיובי שלם, והפלט שלו הוא החזקה הקטנה ביותר של 2 אשר גדולה מנתון הקלט. למשל: עבור הקלט 7 הפלט הדרוש הוא 8 (כי  $2^3=8$ ), ועבור הקלט 8 הפלט הדרוש הוא 16 (כי  $2^4=16$ ). ישמו את האלגוריתם בשפת Java. במהלך הפיתוח הקפידו על ניסוח תת-משימות לביצוע-חוזר ועל ניסוח תנאי לביצוע-חוזר.

אתחל את sum ב-1

1. קלוט מספר שלם לתוך num

2. כל עוד sum קטן או שווה ל num בצע

2.1 הכפל את num פי 2

3. הצג את sum

```
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    int num, sum;
    sum = 1;
    System.out.println("Enter number:");
    num = in.nextInt();
    while (sum <= num)
    {
        sum = sum * 2;
    }
    System.out.println(sum);
}
```

## תוכנית סף ההכפלה

בצלחת פֶּטְרִי החיידקים מכפילים את עצמם פי 5 בכל דקה עד אשר מספרם עובר סף מסוים הנקרא "סף ההכפלה".

פתחו אלגוריתם אשר הקלט שלו הוא שני נתונים: מספר חיידקים התחלתי בצלחת וסף ההכפלה. הפלט שלו הוא מספר החיידקים שיהיו בצלחת בדקה שבה יעבור מספרם את סף ההכפלה. ישמו את האלגוריתם בשפת Java.

אתחול petri ב-0

1. קלוט מספר לתוך bacteria

2. קלוט מספר לתוך limit

3. כל עוד limit גדול מ bacteria בצע

3.1 הכפל bacteria פי 5

3.2 bacteria קטן limit

3.2.1 השם את bacteria ב petri

4. הצג את petri

```
petri = 0;
System.out.println("Enter numbers of bacteria");
```

```

bacteria = in.nextInt();
System.out.println("Enter the limits: ");
limit = in.nextInt();
while(limit > bacteria)
{
    bacteria = bacteria * 5;
    if(bacteria < limit) {
        petri= bacteria;
    }
}

System.out.println(petri);
}

```

## משתנים מטיפוס בוליאני

מטרת הבעיה הבאה: הצגת משתנה מטיפוס בוליאני

פתחו אלגוריתם שהקלט שלו הוא מספר שלם גדול מ-1 ולאחריו רשימה של מספרים שלמים נוספים הגדולים מ-1. סדרת המספרים מסתיימת במספר 0. אם המספר הראשון שנקרא הוא זוגי, אז הפלט הוא כל המספרים מרשימת המספרים וכל מספר מוצג פעמיים. אם המספר הראשון שנקרא הוא אי-זוגי אז הפלט הוא כל המספרים מרשימת המספרים וכל מספר מוצג פעם אחת בלבד.

למשל עבור הקלט: 8 2 4 7 0 הפלט המתאים הוא 2 2 4 4 7 7

ועבור הקלט: 9 2 4 7 0 הפלט המתאים הוא 2 4 7

## האלגוריתם

מאחר שהרשימה מסתיימת בזקיף, נשתמש בהוראה לביצוע-חוזר-בתנאי התלוי בזקיף.

1. קאוט מספר ב- num
2. בדוק אם num זוגי ואם לא אז צא את הבדיקה ב- isEven
3. קאוט את המספר הראשון ברשימה ב- numInList
4. כל עוד  $numList \neq 0$  בצע:
  - 4.1. אם ערכו של isEven הוא true
    - 4.1.1. הצג פעמיים את numInList
    - 4.2. אחר
    - 4.2.1. הצג פעם אחת את numInList
    - 4.3. קאוט את המספר הבא ברשימה ב- numInList



## התוכנית המלאה

```
/*
קלט: מספר שלם ולאחריו רשימת מספרים שלמים גדולים מ-1 שמסתיימת ב-0
פלט: אם המספר הראשון זוגי יוצגו המספרים ברשימה פעמיים, אחרת יוצגו
המספרים ברשימה פעם אחת בלבד
*/
import java.util.Scanner;
public class IfEven
{
    public static void main (String [] args)
    {
        int num;                //המספר הראשון בקלט
        int numInList;           //מספר תורן ברשימת הקלט
        boolean isEven;          //האם המספר הראשון זוגי
        Scanner in = new Scanner(System.in);
        // קליטת המספר הראשון ובדיקה אם הוא זוגי
1.    System.out.print("Enter the first number: ");
2.    num = in.nextInt();
3.    isEven = (num % 2 == 0);
4.    System.out.print("Enter a number. Enter 0 to end the list:");
5.    numInList = in.nextInt();
6.    while (numInList != 0)
        {
6.1.        if (isEven)
6.1.1.            System.out.print(numInList + " " + numInList + " ");
6.2.        else
6.2.1.            System.out.print(numInList + " ");
6.3.            System.out.print("Enter the next number. " +
                                "Enter 0 to end the list");
6.4.            numInList = in.nextInt();
        } // while
    } // main
} // IfEven
```

## הקשר הלוגי לא (NOT)

### משחק ניחושים

במשחק הניחושים מגריל המחשב מספר בתחום 1-100, והשחקן צריך לנחש אותו. בתום המשחק יודיע המחשב לשחקן כמה ניסיונות ניסה עד שהצליח לנחש. לפניכם קטע תוכנית ב-Java. השלימו את הקטע כך שיבצע את הנדרש:

```
Random rnd = new Random();
int num = rnd.nextInt(100) + 1 ; // המספר שהמחשב יגריל
int numOfGuesses = 0 ; // מונה לספירת מספר ניחושים
boolean found = true;
while (!found)
{
    numOfGuesses++;
    System.out.print("Enter your guess: ");
    guess = in.nextInt();
    if (guess == num)
    {
        System.out.println("Correct!! ");
        found = true;
    }
    else if (guess > num )
```



```
System.out.println("Your guess is too big");
else
System.out.println("Your guess is too small");
} // while
System.out.println("It took you " + numOfGuesses + " guesses");
```

## הצפנת אותיות

הבעיה עוסקת בהצפנת הודעות. הודעה היא סדרת מילים, כך שכל מילה היא רצף אותיות מן הא"ב האנגלי ובין כל שתי מילים מופיע סימן קריאה (!). הצפנת הודעה מתבצעת באופן הבא: בכל מילה אי-זוגית (כלומר המילה הראשונה, השלישית, החמישית וכו') מוחלפת כל אות באות העוקבת לה בא"ב. בכל מילה זוגית (כלומר המילה השנייה, הרביעית, השישית וכו') מוחלפת כל אות באות הקודמת לה בא"ב. סימני הקריאה נותרים במקומם ללא שינוי.

כלומר תחילה יש להצפין כל אות נתונה לאות העוקבת לה בא"ב ולאחר כל קריאה של סימן קריאה יש להפוך את "כיוון ההצפנה" (מעוקבת בא"ב לקודמת, או מקודמת בא"ב לעוקבת). למשל, הצפנת ההודעה DING!DING!DONG! תהיה: EJOH!CHMF!EPOH. לשם הפשטות נניח שהאותיות A ו-Z אינן נכללות בהודעה.

### האלגוריתם:

אתחל את  $i$  ב-0

1. קלוט מספר שלם לתוך length

2. כל עוד  $i$  קטן מ length בצע

2.1 קלוט תו לתוך letter

- 2.2 אם letter שונה מ '!' ו check שווה true
  - 2.2.1 הגדל את letter ב 1
  - 2.2.2 הדפס את letter
- 2.3 אחרת אם letter שונה מ '!' ו check שווה false
  - 2.3.1 הקטן את letter ב 1
  - 2.3.2 הדפס את letter
- 2.4 אחרת אם letter שווה ל '!'
  - 2.4.1 השם! check ב check
  - 2.4.2 הדפס את letter
- 2.5 הגדל את i ב 1

```

int leangth ,i;
i = 0;
char letter;
boolean check = true;
Scanner in = new Scanner(System.in);
System.out.print("Enter size of message: ");
leangth = in.nextInt();
while(i < leangth) {
    letter = in.next().charAt(0);
    if(letter != '!' && check == true ) {
        letter++;
        System.out.print(letter);
    }
    else if(letter != '!' && check == false) {
        letter--;
        System.out.print(letter);
    }
    else if(letter == '!') {
        check = !check;
        System.out.println(letter);
    }
    i++;
}

```

במפעל לייצור נעליים יש עובדים רבים. פתחו אלגוריתם שיקבל כקלט את מספר העובדים במפעל, ואחר כך רשימה של כל המשכורות של השנה האחרונה (12 משכורות) לכל אחד מעובדי המפעל. האלגוריתם יחשב וידפיס את המשכורת האחרונה של העובד המסכן שסכום משכורותיו כל השנה היה הנמוך ביותר. ישמו את האלגוריתם בשפת Java.

```
public class targil_7_70 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int num,lastSalary,min,salary;

        salary = 0;
        min = 0;
        lastSalary = 0;
        Scanner in=new Scanner(System.in);
        System.out.println("enter a number of employees");
        num = in.nextInt();
        for(int i = 1 ; i <= num ;i++) {
            int sum = 0;

            for(int j =1 ; j <= 3 ;j++) {
                System.out.println("Enter salary");
                salary = in.nextInt();
                sum = sum + salary;

            }
            if(i==1) {
                min = sum;
                lastSalary=salary ;
            }
            else if(sum < min) {
                min = sum;
                lastSalary=salary ;
            }

        }

        System.out.println(lastSalary);

    }

}
```

## פונקציות

### פלינדרום string כולל בדיקת רווחים

```
public class Palindrome {

    public static boolean isPalindrome(String st) {
        int leftChar = 0;
        int rightChar = st.length() - 1;
        boolean ok = true;
        while (leftChar < rightChar && ok) {
            // בדיקת תווים
            if (st.charAt(leftChar) == st.charAt(rightChar))
            {
                rightChar--;
                leftChar++;
            }
            // בדיקת רווחים
            else if(st.charAt(leftChar) == ' ') {
                leftChar++;
            }
            else if(st.charAt(rightChar)==' ') {
                rightChar--;
            }
            else {
                ok =false;
            }
        }
        return ok;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println(isPalindrome("TOP SPOT"));
    }
}
```

## משולש ישר זווית הפוך – הדפסת מספרים

```
3
4 static void oppositeTriangle() {
5     for(int i = 5; i > 0 ; i--) {
6         for(int j = 0 ; j < i ; j++) {
7             System.out.print(i);
8         }
9         System.out.println();
10    }
11 }
12
13 public static void main(String[] args) {
14     // TODO Auto-generated method stub
15     oppositeTriangle();
16 }
```

<terminated> BackShapes [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (1 F

55555  
4444  
333  
22  
1

## פירמידה

```
7 //פעולה המציירת פירמידה בגודל 5 שורות
8 static void drawPyramid() {
9     for(int i = 1; i <= 5 ; i++) {
10
11         for(int j = 5; j > i ; j--) {
12             System.out.print(" ");
13         }
14         for(int j = 1 ; j <=i ;j++) {
15             System.out.print(" *");
16         }
17         System.out.println();
18     }
19 }
20 //--- 5x5 --- פעולה המציירת ריבוע בגודל
21 static void drawSquare()
22 {
```

Console Problems Debug Shell

<terminated> Shapes [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (1 Feb 2020, 20:4

Enter your choice 4

```
[
    *
  * *
 * * *
* * * *
* * * * *
```



## פירמידה הפוכה.

```
19 }
20 //פירמידה הפוכה
21 static void drawInvertPyramid() {
22     for(int i = 5; i > 0; i--) {
23         //לולאה לטיפול ברווחים
24         for(int j = 0; j < 5 - i; j++) {
25             System.out.print(" ");
26         }
27         //לולאה להדפסת הכוכביות
28         for(int j = 0; j < i; j++) {
29             System.out.print(" *");
30         }
31         //שורה
32         System.out.println();
33     }
```

Console Problems Debug Shell

<terminated> Shapes [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (1 F

Enter your choice 5

```
* * * * *
 * * * *
  * * *
   * *
    *
     *
```

## תוכנית סדרה חשבונית – פיבונצ'י

```
21
22
23 public static void main(String[] args) {
24
25     int n = 10, t1 = 0, t2 = 1;
26     System.out.print("First " + n + " terms: ");
27     for (int i = 1; i <= n; ++i)
28     {
29         System.out.print(t1 + " ");
30         int sum = t1 + t2;
31         t1 = t2;
32         t2 = sum;
33     }
34 }
35 }
```

<

Console Problems Debug Shell

<terminated> fibonacci [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (1 Feb 2020, 21:25:17)

First 10 terms: 0 1 1 2 3 5 8 13 21 34

## פונקציה לבדיקת מספר ראשוני

```
4 public static boolean prime(int num) {
5     boolean prime = true;
6     for (int i = 2; i <= (int) Math.sqrt(num); i++) {
7         if (num % i == 0)
8             return false;
9     }
10    return prime;
11 }
12
13 public static void main(String[] args) {
14
15     System.out.println(prime(13));
16 }
17
18 }
```

Console Problems Debug Shell

<terminated> PrimeNumber [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (1 Feb 2020, 21:29:04)

true

## תוכנית מספר מושלם

מספר N ייקרא מספר מושלם אם הוא שווה לסכום המחלקים שלו. דוגמא : 28 הוא מספר מושלם מכיוון ש  $28=1+2+4+7+14$ .

```
System.out.println("enter a number ");|
int a = reader.nextInt();
int sum = 0;
for(int i=1;i<=(a-1);i++)
{
    if(a%i==0)
    {
        sum = sum + i;
    }
}
if(a==sum)
{
    System.out.println("Is a perfect number");
}
else
{
    System.out.println("Is NOT a perfect number");
}
```

,

**פונקציה שאומרת בידיוק כמה ספרות מונה המספר**

```
public static int number_of_digits_f(int x)
{
    int counter = 0;
    for(counter=0 ; x!=0; counter++)
        x=x/10;
    return(counter);
}
```

**פונקציה להחלפה בין 2 מספרים בעזרת משתנה זמני**

```
private static void swap(int x, int y) {  
    int temp = x;  
    x = y;  
    y = temp;  
  
    System.out.println("After Swapping");  
    System.out.println("Value of x is :" + x);  
    System.out.println("Value of y is :" + y);  
}  
  
//main  
public static void main(String[] args) {  
    int x = 10;  
    int y = 20;  
    System.out.println("Before Swapping");  
    System.out.println("Value of x is :" + x);  
    System.out.println("Value of y is :" + y);  
  
    //swap the value  
    swap(x, y);  
}
```

פונקציה למציאת מספר מקסימלי מתוך קבוצה.

```
public static int max_from_N_numbers_f(int num,int N)
{
Scanner input = new Scanner(System.in);
int max,j=0;
max=num;
while( j<N-1)
{
num=input.nextInt(); //scanner
if(max<num) {
max=num;
}
j++;
}

System.out.println(max);
return(max);
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    max_from_N_numbers_f(100, 4);
}
```

פונקציה למציאת מספר מינימלי מתוך קבוצה

```
public static int min_from_N_numbers_f(int num,int N)
{
Scanner input = new Scanner(System.in);
int min,j=0;
min=num;
while( j<N-1)
{
num=input.nextInt(); //scanner
if(min>num) {
min=num;
}
j++;
}

System.out.println(min);
return(min);
}

public static void main(String[] args) {
    min_from_N_numbers_f(100, 4);
}
```



#### תוכנית שבודקת את כמות התווים במחרוזת

```
public static void main(String[] args) {  
    String string = "The best of both worlds";  
    int count = 0;  
    //Counts each character except space  
    for(int i = 0; i < string.length(); i++) {  
        if(string.charAt(i) != ' ')  
            count++;  
    }    System.out.println("Total number of characters in a string: " + count); }
```

#### תוכנית reverse למחרוזת

```
public static void main(String[] args) {  
    String string = "Dream big";  
    String reversedStr = "";  
    for(int i = string.length()-1; i >= 0; i--){  
        reversedStr = reversedStr + string.charAt(i);  
    }  
    System.out.println("Original string: " + string);  
    System.out.println("Reverse of given string: " + reversedStr);  
}
```

### תוכנית להחלפה בין 2 מילים במחרוזת

```
String str1 = "Good ", str2 = "morning ";

    System.out.println("Strings before swapping: " + str1 + " " + str2);

    //Concatenate both the string str1 and str2 and store it in str1
    str1 = str1 + str2;

    //Extract str2 from updated str1
    str2 = str1.substring(0, (str1.length() - str2.length()));

    //Extract str1 from updated str1
    str1 = str1.substring(str2.length());

    System.out.println("Strings after swapping: " + str1 + " " + str2);

//פלט→ morning good
```

### תוכנית שמוצאת את הספרה הכי קטנה במספר נתון

```
public class findMinimum {
    public static void main(String[] args) {

        int num=57830231;
        int min;
        int digit=0;

        min=num;

        while(num!=0){
            digit=num%10;
            if(digit<min){
                min=digit;
            }

            num=num/10;
        }

        System.out.println("min number is = " + min);
    }
}
```

```
public static void main(String[] args)
{ int n=input.Scanner();
    int k=(n*2)-1;
    for(int i=1;i<=k;i++)
    {
        for(int j=1;j<=k;j++)
        {
            if(j==i || j==k-i+1)
                System.out.print("*");
            System.out.print(" ");
        }
        System.out.println();
    }
}
```

## תוכנית להדפסת ריבוע בעזרת כוכביות ע"פ אורך-רוחב נתון

```
public static void main(String[] args)
{
    int number = 7; // כמות הכוכביות
    for (int i = 0; i < number; i++)
    {
        if (i == 0 || i == 6)
        {
            for (int j = 0; j < number; j++)
            {
                System.out.print("*");
            }
            System.out.println();
        }
        if (i >= 1 && i <= 5)
        {
            for (int j = 0; j < number; j++)
            {
                if (j == 0 || j == 6)
                {
                    System.out.print("*");
                }
                else if (j >= 1 && j <= 5)
                {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```

```
*****
*      *
*      *
*      *
*      *
*      *
*      *
*****
```

## משולש ישר זווית הפוך בנוי ממספרים

```
import java.util.Scanner;

public class Pattern1
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of rows: ");

        int rows = sc.nextInt();

        System.out.println("Here is your pattern....!!!");

        for (int i = rows; i >= 1; i--)
        {
            for (int j = i; j >= 1; j--)
            {
                System.out.print(j+" ");
            }

            System.out.println();
        }
    }
}
```

### Output

Enter number of rows: 5

Here is your pattern....!!!

5 4 3 2 1

4 3 2 1

3 2 1

2 1

1

```
public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter number of rows: ");
    int rows = sc.nextInt();
    for (int i = 1; i <= rows; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            System.out.print(j+" ");
        }
        System.out.println();
    }
}
```

output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```



Program 3) Print the following pattern

5 4 3 2 1

5 4 3 2

5 4 3

5 4

5

Program

```
import java.util.Scanner;

public class Pattern3
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of rows: ");

        int rows = sc.nextInt();

        System.out.println("Here is your pattern....!!!");

        for (int i = 1; i <= rows; i++)
        {
            for (int j = rows; j >= i; j--)
            {
                System.out.print(j+" ");
            }

            System.out.println();
        }
    }
}
```

Output

Enter number of rows: 5

Here is your pattern....!!!

5 4 3 2 1

5 4 3 2

5 4 3

5 4

5

**Program 4) Print the following pattern**

1  
10  
101  
1010  
10101

**Program**

```
import java.util.Scanner;

public class Pattern4
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of rows: ");

        int rows = sc.nextInt();

        System.out.println("Here is your pattern.....!!!");

        for (int i = 1; i <= rows; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                if(j%2 == 0)
                {
                    System.out.print(0);
                }
                else
                {
                    System.out.print(1);
                }
            }

            System.out.println();
        }
    }
}
```

```
public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter row for pattern : ");
    int rows = sc.nextInt();
    System.out.println("Here is your pattern....!!!");
    for (int i = rows; i >= 1; i--)
    {
        for (int j = 1; j <= i; j++)
        {
            System.out.print(j+" ");
        }
        System.out.println();
    }
    for (int i = 2; i <= rows; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            System.out.print(j+" ");
        }
        System.out.println();
    }
}
```

Enter row for pattern : 5

Here is your pattern....!!!

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

