

# Loops And Arrays

# Loops

# for and while loop

- לולאה הינה טכניקה שמאפשרת חזרה על קטע קוד מספר פעמים
- לולאה נכונה מקיימת את התנאים הבאים :

✓בלוק נכון  
✓תנאי עצירה

```
for ( פקודה 1 ; תנאי ; פקודה 2 )  
    פקודה 3 ;
```

```
for ( פקודה 1 ; תנאי ; פקודה 2 ) {  
    פקודה 3 ;  
    פקודה 4 ;  
}
```

```
while (תנאי)  
    פקודה 1 ;
```

loop body

```
while (תנאי){  
    פקודה 1 ;  
    פקודה 2 ;  
    :  
}
```

Loop body can be one statement or a block statement.

# while and do while loop

```
do  
    statement;  
while (condition);
```

≡

```
statement;  
while (condition)  
    statement;
```

```
do{  
    statement1;  
    statement2;  
}  
while (condition);
```

≡

```
statement1;  
statement2;  
while (condition){  
    statement1;  
    statement2;  
}
```

**do while , in contrast to while loop, , guaranteed to execute at least one time.**

# for and while loop

```
for(statement1;condition;statement2){  
    statement3;  
}
```

≡

```
statement1;  
while (condition){  
    statement3;  
    statement2;  
}
```

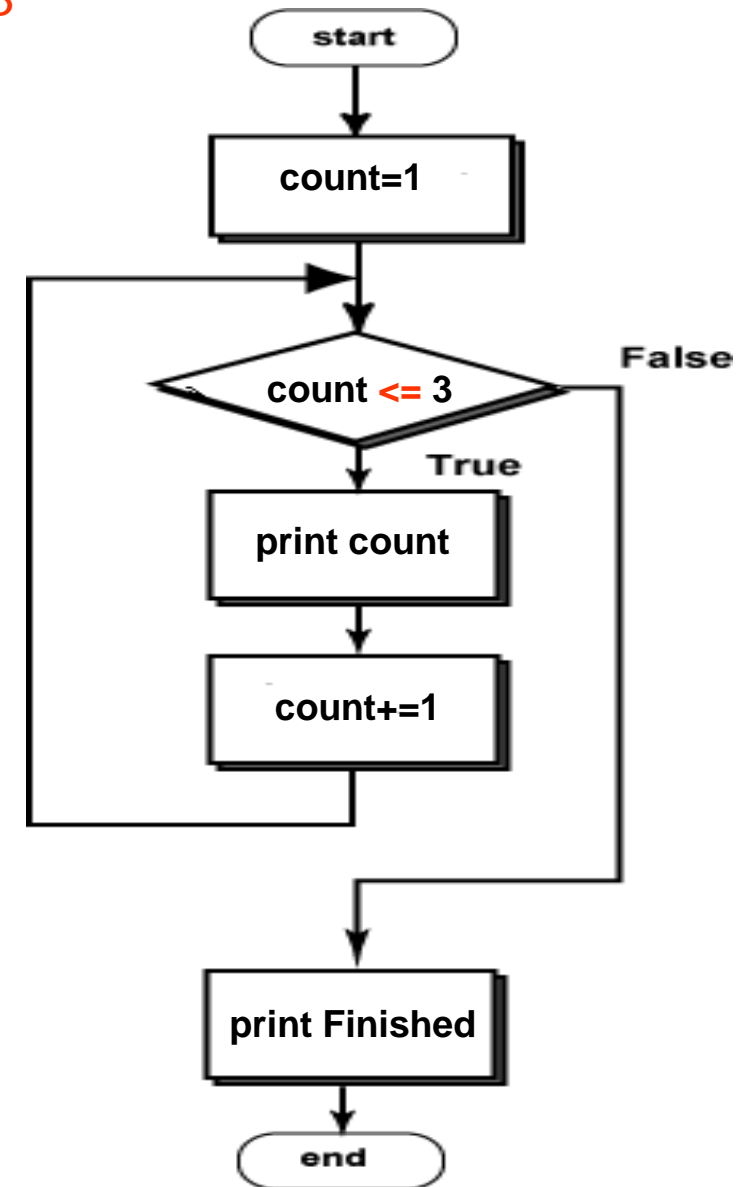
ניתן לפתור את אותה הבעיה עם שני סוגי הלולאות. בדרך כלל לולאת while שימושית כאשר אנחנו לא יודעים מתי התנאי של הלולאה יהפוך מtrue ל false ואין לדעת כמה פעמים תרוץ הלולאה. למשל יש לכתוב תוכנית הקולטת מספרים מהמשתמש עד שהמשתמש מזין את הערך אפס. לעומת זאת בלולאת for משתמשים בדרך כלל כאשר רוצים לבצע את הלולאה מספר פעמים ידוע מראש למשל לרוץ מ 1 עד 100.

count < 4 == count <= 3

```
public class Tester{  
    public static void main(String[] args) {  
        int count;  
        count = 1;  
        while ( count <= 3 ) {  
            System.out.println(count);  
            count += 1;  
        }  
        System.out.println("Finished");  
    }  
}
```

Output:

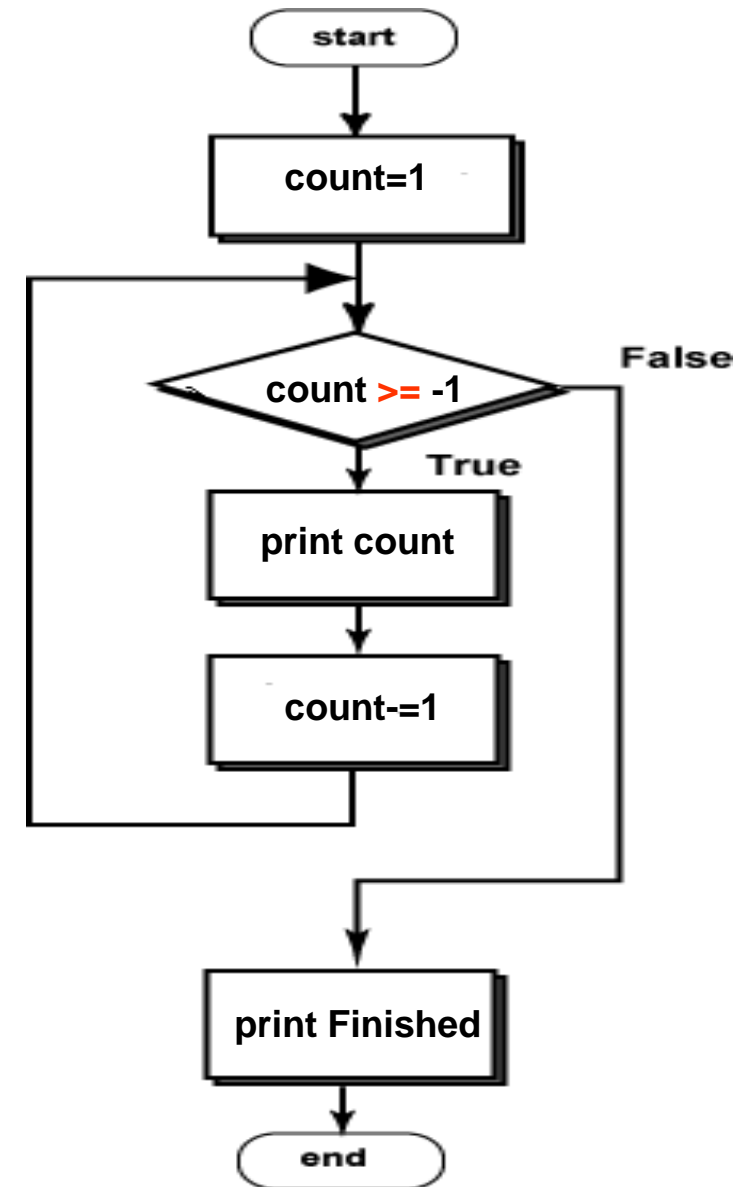
1  
2  
3  
Finished



```
public class Tester{  
    public static void main(String[ ] args){  
        int count;  
        count = 1;  
        while( count >= -1 ) {  
            System.out.println(count);  
            count -= 1;  
        }  
        System.out.println("Finished");  
    }  
}
```

**Output:**

1  
0  
-1  
Finished



```
public static void main(String[] args){
    int counter;
    counter=0;
    while( counter <=3 ) {
        System.out.println(counter);
        counter+=1;
    }
} //0 1 2 3
```

```
public static void main(String[] args){
    int counter;
    counter=0;
    while( (counter += 1) <= 3 ) {
        System.out.println(counter);
    }
} //1 2 3
```

```
public static void main(String[] args){
    int counter;
    counter=0;
    while( ++counter <= 3 ) {
        System.out.println(counter);
    }
} //1 2 3
```

```
public static void main(String[] args){
    int counter;
    counter=0;
    while( counter++ <= 3 ) {
        System.out.println(counter);
    }
} //1 2 3 4
```



# כתוב תוכנית אשר קולטת שני מספרים שלמים ומדפיסה את כל המספרים השלמים שבניהם

```
import java.util.Scanner;
public class Tester {
    public static void main(String[] args) {
        int start , end;
        Scanner scan = new Scanner(System.in);
        System.out.println("Insert start value");
        start = scan.nextInt();
        System.out.println("Insert end value");
        end = scan.nextInt();
        while( start <= end ) {
            System.out.println(start);
            start++;
        }
    }
} // input: 1 5 → output : 1 2 3 4 5
```

# כתוב תכנית אשר מדפיסה את המספרים הזוגיים בין 1 ל 100

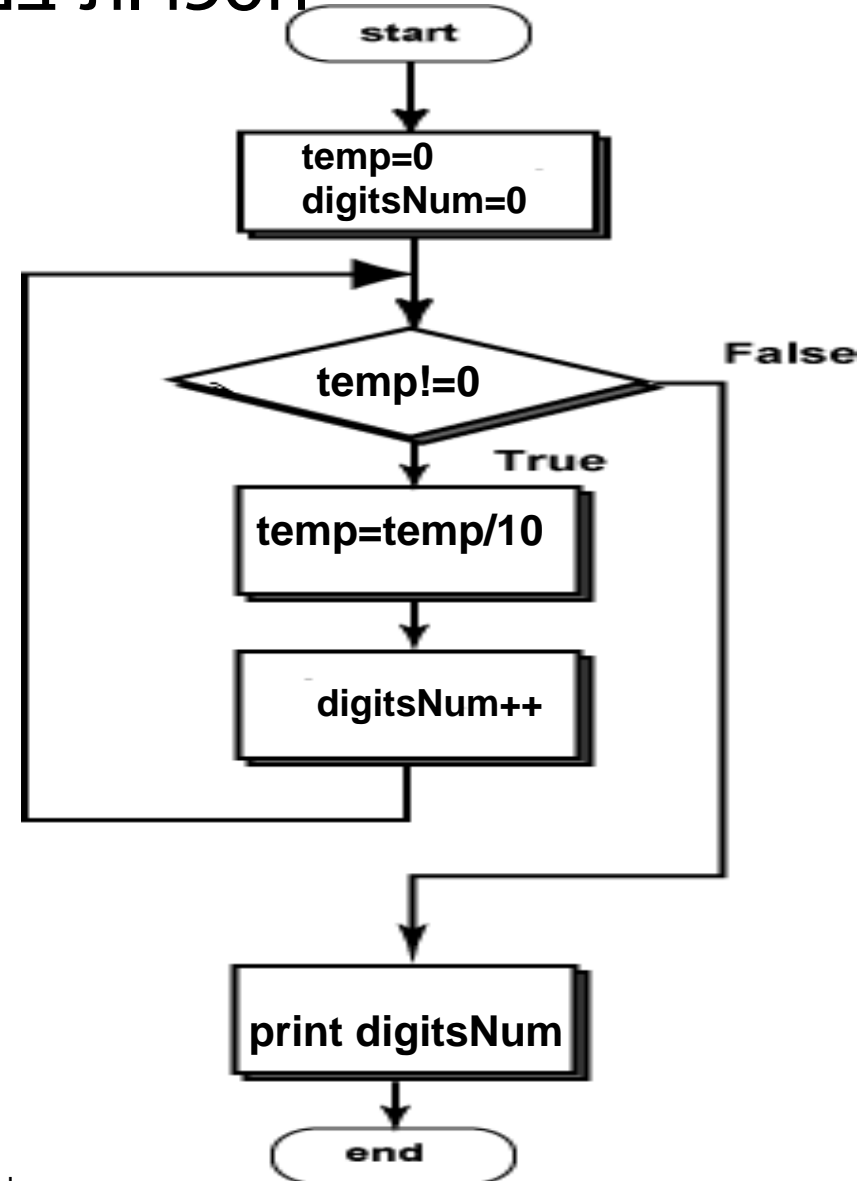
```
public static void main(String[ ] args) {  
    int i=1;  
    while( i <= 100 ) {  
        if( (i % 2) == 0 )  
            System.out.println(i);  
        i++;  
    }  
}
```

מספר זוגי הינו מספר  
שמתחלק ב 2 ללא  
שארית (שארית אפס)

# כתוב תוכנית אשר מחשבת את מספר הספרות במספר שלם כלשהו

```
public static void main(String[ ] args){  
    int digitsNum = 0;  
    int num = 123;  
    int temp = num;  
    while( temp != 0 ) {  
        temp = temp / 10;  
        digitsNum++;  
    }  
    System.out.println(digitsNum);  
}
```

טעות: בדוק כאשר המספר הוא 0???  
פתרון : להכניס עוד תנאי (if) אחרי לופני  
לולאת ה while שבודק זאת או לעבור  
לעמוד הבא



## עוד שתי תוכניות מתוקנות לחישוב מספר הספרות במספר שלם

```
public static void main(String[] args){  
    int digitsNum=0;  
    int num=123;  
    int temp=num;  
  
    temp = temp / 10;  
    digitsNum++;  
    while( temp != 0) {  
        temp = temp / 10;  
        digitsNum++;  
    }  
    System.out.println(digitsNum);  
}
```

```
public static void main(String[] args){  
    int digitsNum=0;  
    int num=123;  
    int temp=num;  
  
    do {  
        temp = temp / 10;  
        digitsNum++;  
    }  
    while( temp != 0 );  
    System.out.println(digitsNum);  
}
```

מספר (או מחרוזת תווים) נקרא Palindrome אם אנו מקבלים אותו מספר (או מחרוזת) בקריאה משני הכיוונים (למשל 1234321 או "abcdcba"). בהינתן מספר שלם חיובי בדוק אם הוא Palindrome

```
public static void main(String[] args){  
    int num=123;  
    int temp=num;  
    int newNum=0;  
    while( temp != 0 ){  
        newNum *= 10;  
        newNum += temp % 10;  
        temp /= 10;  
    }  
    if( newNum == num)  
        System.out.println("yes");  
    else  
        System.out.println("no");  
}
```

בהינתן מספר חיובי num  
מטיפוס byte כתוב תכנית  
אשר מדפיסה את הצורה  
הבינארית של המספר הנתון

```
public static void main(String[] args){
    byte num=4;
    int temp=num, bin=1;
    if( num >= 0 ) {
        while( temp != 0 ) {
            bin *= 10;
            bin += temp % 2;
            temp /= 2;
        }
        temp = bin;
        bin = 0;
        while( temp != 1 ) {
            bin *= 10;
            bin += temp % 10;
            temp /= 10;
        }
        System.out.println(bin);
    }
}
```

גרסה 1

תנאי שמבטיח שהמספר  
הנבדק אינו שלילי

הייצוג הבינארי  
לאחר חלוקה ב 2  
בסדר הפוך  
(העמודה הכחולה)  
שימו לב : הספרה  
השמאלית ביותר  
הינה הספרה 1 והיא  
לא נכללת בחישוב  
שלנו. למה?

הפיכת הסדר של  
הספרות שנשמרו  
במשתנה bin  
מהשלב הקודם

שארית	תוצאת חילוק	חילוק ב 2 עם שארית
0	2	4
0	1	2
1	0	1

האלגוריתם עוצר כאשר  
תוצאת החילוק 0

## גרסה 2 שימוש ב String

```
public static void main (String[] args){
    byte num=4;
    String s1="";

    do {
        s1+=num%2;
        num/=2;
    } while (num!=0);

    int len = s1.length()-1;
    String s2 = "";

    while(len>=0){
        s2 += s1.charAt(len);
        len--;
    }

    System.out.println(s2);
}
```

גרסה 2

בהינתן מספר חיובי num  
מטיפוס byte כתוב תכנית  
אשר מדפיסה את הצורה  
הבינארית של המספר הנתון

כרגע ב s1 יש  
את הייצוג  
הבינארי בסדר  
הפוך

S2 מכיל את  
התוכן של s1  
בסדר הפוך

שארית	תוצאת חילוק	חילוק ב 2 עם שארית
0	2	4
0	1	2
1	0	1

האלגוריתם עוצר כאשר  
תוצאת החילוק 0

האם ניתן לוותר על הלולאה השנייה?

תשובה בעמוד הבא

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

```
    byte num=10;
```

```
    String s1="";
```

```
    do {
```

```
        s1=num%2+s1;
```

```
        num/=2;
```

```
    } while (num!=0);
```

```
    System.out.println(s1);
```

```
}
```

גרסה 3

מקודם השרשור היה לסוף המחרוזת:

$s1+=num\%2 \rightarrow s1=s1+num\%2$

עכשיו השרשור תמיד להתחלה :

$s1=num\%2+s1;$



# בהינתן מספר שלם חיובי n כתוב תוכנית אשר מחשבת את n!

```
public static void main(String[ ] args) {  
    int n=5;  
    int factorial=1;  
  
    for ( int i = n ; i > 0 ; i-- )  
        factorial *= i;  
  
    if( n < 0 )  
        System.out.println("Invalid input");  
    else  
        System.out.println(factorial);  
}
```

# בהינתן שני מספרים שלמים יש לחשב את המחלק המשותף הגדול ביותר GCD

```
public static void main(String[] args){
    int numA = 616;
    int numB = 165;
    int a = numA , b = numB , c;
    while( (a % b) != 0) {
        c = a%b;
        a = b;
        b = c;
    }
    System.out.println(b);
}
```

גרסה 1

```
public static void main(String[] args){
    int numA = 18;
    int numB = 12;
    int a = numA , b = numB , c;
    do{
        c=a%b;
        a = b;
        b = c;
    }
    while(c!=0) ;
    System.out.println(a);
}
```

גרסה 2

Gcd ( 18 , 12 ) = 6  
Gcd ( 616 , 165 ) = 11  
Gcd ( 1071 , 1029 ) = 21

## Euclid's Algorithm for GCD

a%b	a/b	b	a
121	3	165	616
44	1	121	165
33	2	44	121
11	1	33	44
0	3	11	33
-	-	0	11

תנאי העצירה הוא כאשר  
תוצאת השארית 0 ואז b הוא  
המחלק המשותף

בהינתן שני מספרים  $a$  ו- $b$  יש לחשב  
את החזקה  $a^b$  ( $a > 0$  and  $b \geq 0$ )

```
public static void main(String[] args){  
    int a=2;  
    int b=4;  
    int pow=1;  
    for ( int i=1 ; i <= b ; i++ )  
        pow *= a;  
    System.out.println(pow);  
}
```

יש לחשב את סכום הסדרה  $s = 1+2+4+5+7....20$  (ללא  
3,6,9,... – מספרים המתחלקים ב 3 ללא שארית)

```
public static void main(String[] args){  
    int s = 0;  
    for ( int i=1 ; i <= 20 ; i++ ) {  
        if ( (i%3) != 0 )  
            s += i;  
    }  
    System.out.println(s);  
}
```

חשב את סכום הסדרה  $s = \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \dots\dots\dots \frac{1}{10}$

```
public static void main(String[] args){  
    double s=0;  
    for ( int i=2 ; i <= 10 ; i++ ) {  
        s += 1.0/i;  
    }  
    System.out.println(s);  
}
```

$s = 1 + 2 + 3 + 4 + \dots$  : נתונה הסדרה הבאה :  
מצא את מספר האברים  $n$  כך שסכום  $S(n)$  יהיה גדול או שווה ל 100  
בדוגמה לעיל :  $s(14)=105$

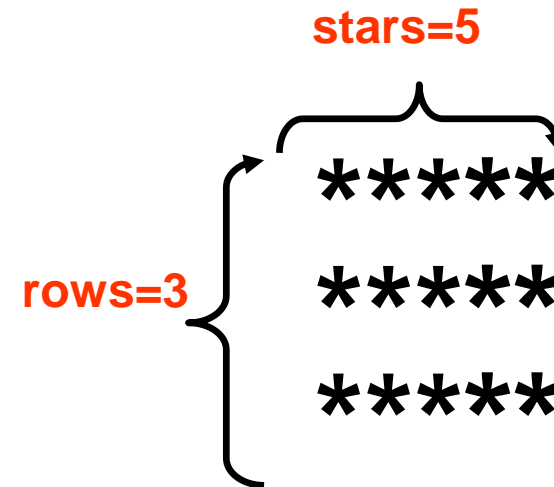
```
public static void main(String[ ] args) {  
    int s=0 , n=0;  
    while( s < 100 ) {  
        n++;  
        s += n;  
    }  
    System.out.println("S(" + n + ")=" + s);  
}
```

# Nested Loops

# Rows of Stars

כתוב תוכנית אשר מדפיסה את הצורה למטה

```
public static void main(String[ ] args) {  
    System.out.println("*****");  
    System.out.println("*****");  
    System.out.println("*****");  
}
```

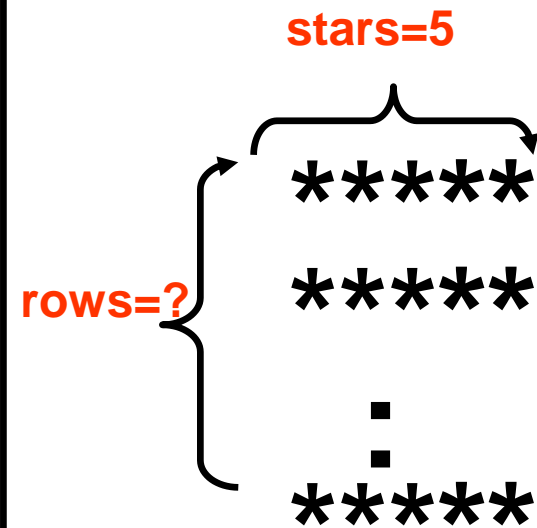




# Rows of Stars

כתוב תוכנית אשר מדפיסה את הצורה למטה

```
public static void main(String[ ] args) {  
    int rows;  
    Scanner input=new Scanner(System.in);  
  
    System.out.print("Insert number of rows:");  
    rows=input.nextInt();  
  
    for (int i=0 ; i < rows ; i++)  
        System.out.println ("*****");  
}
```



# Rows of Stars

כתוב תוכנית אשר מדפיסה את הצורה למטה

```
public static void main(String[ ] args) {  
    System.out.println("***");  
}
```

stars=3

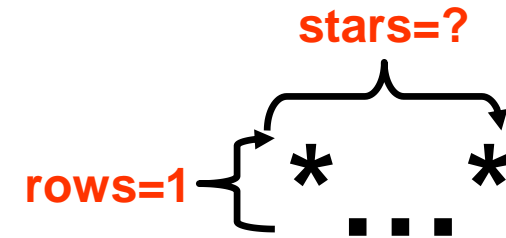
rows=1

\*\*\*

# Rows of Stars

כתוב תוכנית אשר מדפיסה את הצורה למטה

```
public static void main(String[ ] args) {  
    int stars;  
    Scanner input=new Scanner(System.in);  
    System.out.print("Insert number of stars:");  
    stars=input.nextInt();  
    for ( int i=0 ; i < stars ; i++ )  
        System.out.print("*");  
    System.out.println();  
}
```

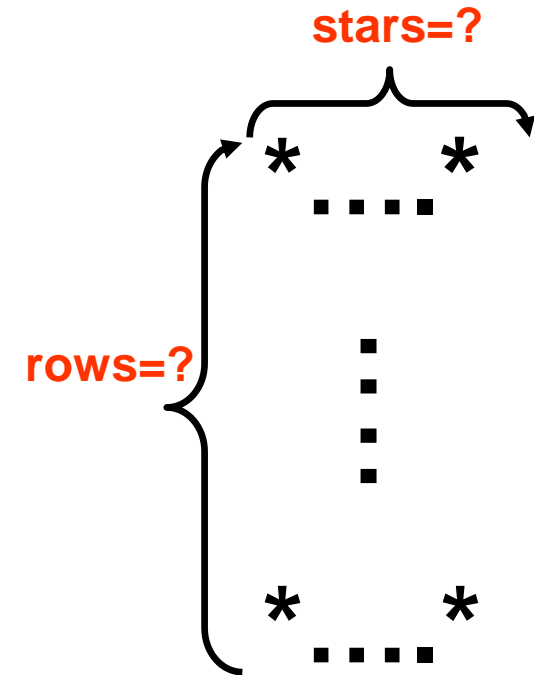


# Rows of Stars

כתוב תוכנית אשר מדפיסה את הצורה למטה

```
public static void main(String[ ] args) {  
    int stars , rows;  
    Scanner input=new Scanner(System.in);  
    System.out.print("Insert number of stars:");  
    stars=input.nextInt();  
    System.out.print("Insert number of rows:");  
    rows=input.nextInt();  
    for ( int i=0 ; i < rows ; i++ ) {  
        for ( int j=0 ; j < stars ; j++ ) {  
            System.out.print("*");  
        }  
        System.out.println();  
    }  
}
```

version 1

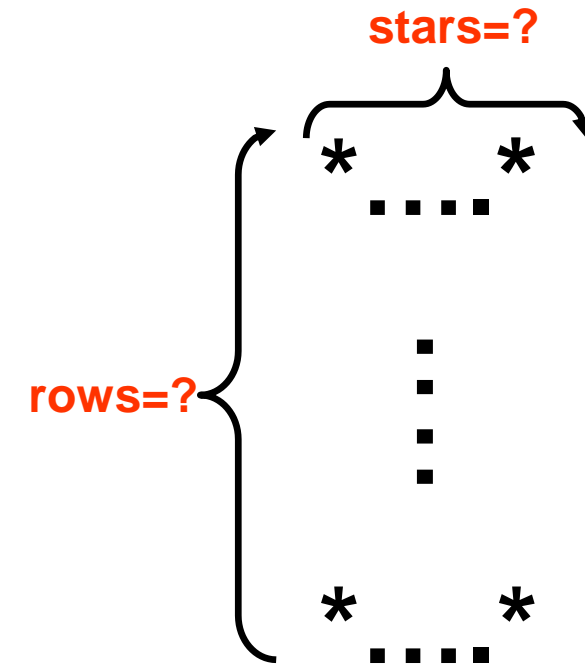


```

public static void main (String [ ] args ) {
    Scanner input = new Scanner( System.in );
    int rows , stars;
    int i , j;
    System.out.print("Insert number of stars:");
    stars=input.nextInt();
    System.out.print("Insert number of rows:");
    rows=input.nextInt();
    i=1;
    while ( i <= rows ) {
        j=1;
        while ( j <= stars ) {
            System.out.print("*");
            j = j + 1;
        }
        System.out.println();
        i = i + 1;
    }
}


```

version 2



# Multiplication Table

i/j	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

 **i\*j**

# Multiplication Table

```
public class Tester {  
    public static void main (String[ ] args ) {  
        int i , j;  
        for( i=1 ; i<10 ; i++) {  
            for(j=1 ; j<10 ; j++) {  
                System.out.print ( i*j + "\t" );  
            }  
            System.out.println();  
        }  
    }  
}
```

חישוב  $S=1!+2!+ \dots n!$

```
public static void main (String [ ] args ) {  
    int n=5;  
    int factorial , s=0;  
    for( int i=1 ; i<=n ; i++) {  
        factorial = 1;  
        for( int j=i ; j>0 ; j-- )  
            factorial *= j;  
        s += factorial;  
    }  
    System.out.println(s);  
}
```

version 1

מה הסיבוכיות ?  
תשובה  $O(n^2)$



חישוב  $S=1!+2!+ \dots n!$

```
public static void main (String[] args ) {  
    int n=4;  
    int factorial=1 , s=0;  
    for( int i=1 ; i<=n ;i++ ) {  
        factorial *= i;  
        s += factorial;  
    }  
    System.out.println(s);  
}
```

version 2

מה הסיבוכיות ?  
תשובה  $O(n)$

# הדפס את כל המספרים n-ספרתיים שהם Palindrome.


```
public static void main (String[ ] args ) {  
    int n=3;  
    int temp , newNum;  
  
    for (int num = (int)Math.pow(10 , n-1) ; num < Math.pow(10,n) ; num++) {  
        temp=num;  
        newNum=0;  
        while( temp !=0 ) {  
            newNum *= 10;  
            newNum += temp%10;  
            temp /= 10;  
        }  
        if ( newNum == num )  
            System.out.println(num);  
    }  
}
```

# התוכנית קולטת מחרוזת ובודקת אם היא Palindrome

## התוכנית תעצור אם המשתמש מקיש תו שונה "y/Y"

```
import java.util.Scanner;
public class Tester{
    public static void main(String[] args){
        String str , ans="y";
        int left , right;
        Scanner scan=new Scanner(System.in);
        while (ans.equalsIgnoreCase("y") ) {
            System.out.print("Enter string to check:");
            str=scan.nextLine();
            left = 0;
            right = str.length() - 1 ;
            while ( left<right && str.charAt(left)==str.charAt(right) ) {
                left++;
                right--;
            }
            if ( left < right )
                System.out.println("Not palindrom");
            else
                System.out.println("Palindrom");
            System.out.print("Check another string(y/n)?:");
            ans=scan.nextLine();
        }
    }
}
```

abcdedcba



left right

כתוב תוכנית הקולטת מחרוזת של תווים באנגלית ומחשבת את מספר האותיות בכתוב גדול וכן בכתוב קטן. התוכנית תעצור כאשר המשתמש מקיש תו השונה מ "Y/y"

```
public static void main(String[ ] args) {  
    Scanner scan=new Scanner(System.in);  
    String str, ans="y";  
    int upper,lower;  
    char ch;  
    while ( ans.equalsIgnoreCase("y") ) {  
        System.out.print("Insert String:");  
        str=scan.nextLine();  
        upper=0 ; lower=0;  
        for ( int i=0 ; i<str.length() ;i++ ) {  
            ch=str.charAt(i);  
            if ( ch >= 'A' && ch <= 'Z' )  
                upper++;  
            else if ( ch >= 'a' && ch <= 'z' )  
                lower++;  
        }  
        System.out.println("Upper=" +upper + " Lower=" + lower);  
        System.out.print("Check another string(y/n)?:");  
        ans=scan.nextLine();  
    }  
}
```

נגדיר מחרוזת כמערך של תווים. בהינתן שני מערכים מסוג char כתוב תוכנית אשר בודקת אם המחרוזת המיוצגת ע"י המערך השני היא תת מחרוזת במערך הראשון

```
public static void main(String[ ] args) {  
    char[ ] str1 = {'H', 'e', 'l', 'l', 'o', 'w', ' ', 'w', 'o', 'r', 'l', 'd'};  
    char[ ] str2 = {'w', 'o'};  
    int i;  
    for ( i=0 ; i<str1.length ; i++) {  
        int start=i;  
        int j;  
        for ( j=0 ; j<str2.length ; j++ , pi++) {  
            if (start< str1.length && str1[start] != str2[j] )  
                break;  
        }  
        if ( j == str2.length ) {  
            System.out.println("Sub string starts at:" + (++i) );  
            break;  
        }  
    }  
    if ( i == str1.length )  
        System.out.println("Sub string not found");  
}
```

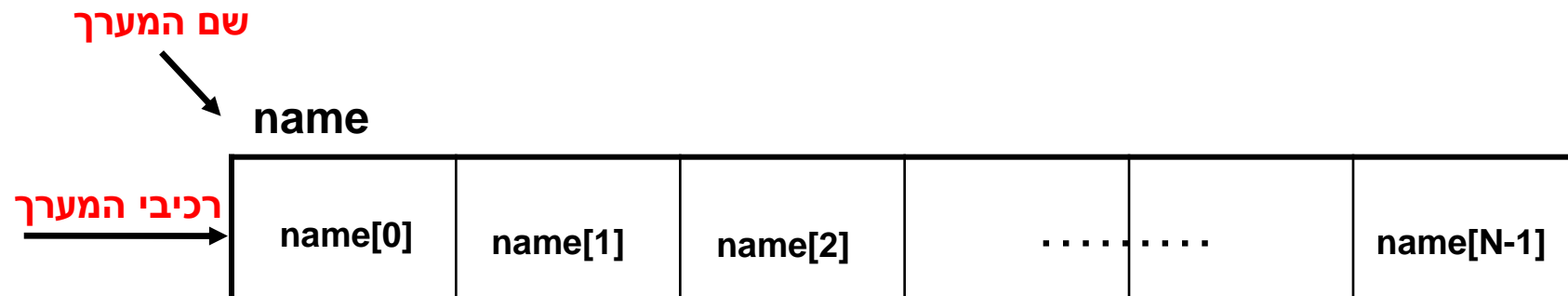
כמו מקודם אבל הפעם צריך להכיל בפתרון  
מחרוזות זהות גם אם הן מכילות תווים גדולים\קטנים זהים

```
public static void main(String[ ] args) {  
    char[ ] str1 = {'H', 'e', 'l', 'l', 'o', 'w', ' ', 'w', 'o', 'r', 'l', 'd'};  
    char[ ] str2 = {'W', 'o'};  
    int i;  
    for ( i=0; i<str1.length; i++) {  
        int start=i;  
        int j;  
        for ( j=0; j<str2.length; j++, start++) {  
            if ( start < str1.length && !( str1[start] == str2[j] ||  
                                             str1[start] + 32 == str2[j] ||  
                                             str1[start] - 32 == str2[j] ) )  
                break;  
        }  
        if ( j==str2.length ) {  
            System.out.println("Sub string starts at:" + (++i) );  
            break;  
        }  
    }  
    if ( i==str1.length )  
        System.out.println("Sub string not found");  
}
```

# Arrays

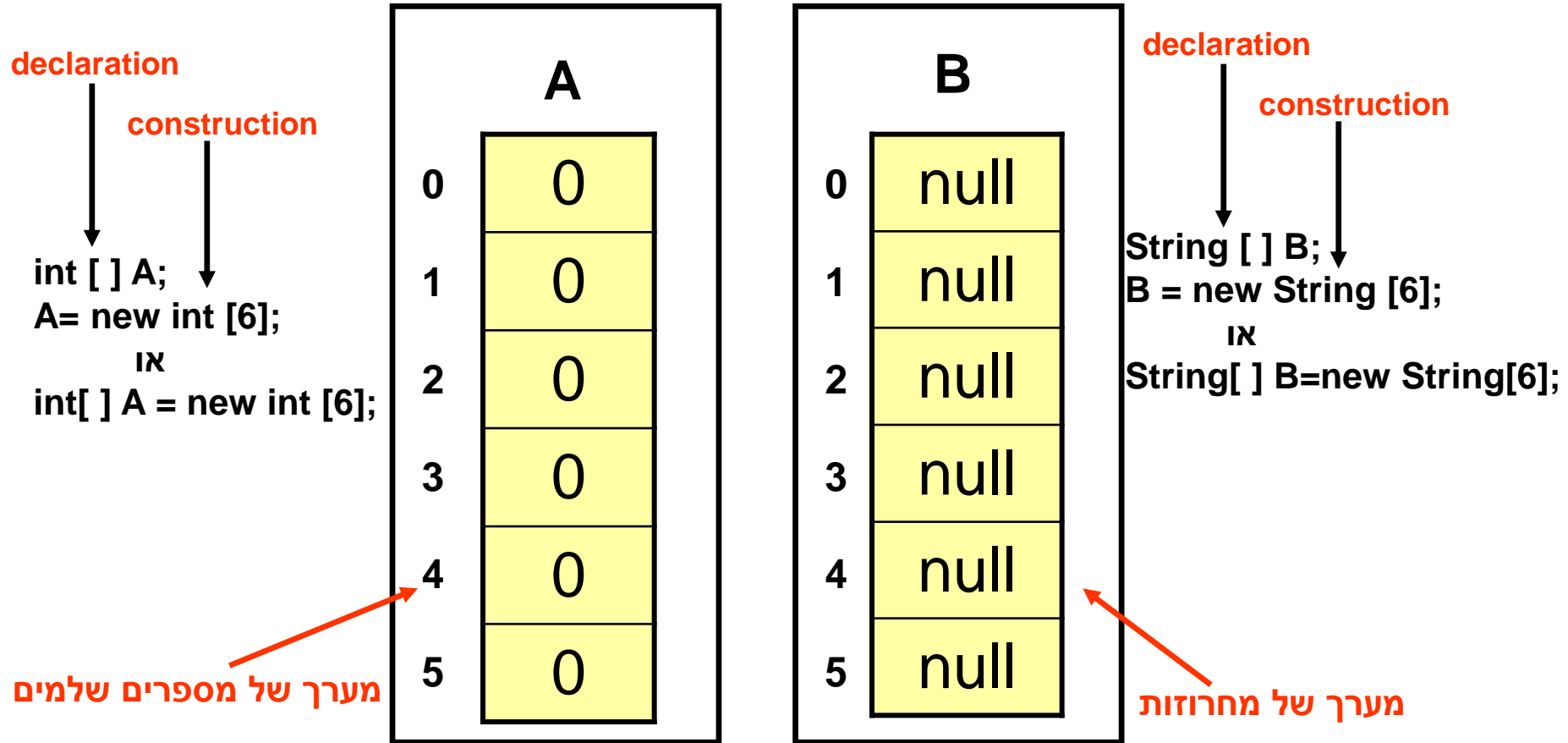
# מערך

- מבנה נתונים פשוט
- רצף זהה של משתנים מאותו סוג
- אוסף פריטים שניתן לגשת אליהם בצורה ישירה באמצעות אינדקס
- סיבוכיות גישה  $O(1)$
- דרישת הזיכרון היא בדיוק הזיכרון הדרוש לנתונים עצמם
- חסרונות: מבנה סטטי





# Declaration And Construction Of Arrays



# Array Assignment

A	
0	3
1	0
2	1
3	0
4	2
5	2

```
public static void main (String[ ] args) {  
    int [ ] A = new int[6];  
    A[0] = 3;  
    A[1] = 0;  
    A[2] = 1;  
    A[3] = 0;  
    A[4] = 2;  
    A[5] = 2;  
}
```

# Declare + Construct + Initialize An Array All In One Statement

A	
0	3
1	0
2	1
3	0
4	2
5	2

```
public static void main ( String [ ] args ) {  
    int [ ] A = {3 , 0 , 1 , 0 , 2 , 2};  
}
```

Index of array  
is always a positive integer

A	
0	0
1	0
2	0
3	0
4	0
5	0

```
A[0] = 9;  
A[1] = 6;  
:  
:
```

This is wrong :  
 $A[-1] = 0;$   
 $A[1.5] = 7;$

A	
0	9
1	6
2	0
3	0
4	0
5	0

# Arithmetic Expressions is allowed while using arrays


A	
0	9
1	6
2	8
3	7
4	12
5	0

```
int x = A[0] + 1;  
A[2] = (x + A[1]) / 2;  
A[3] = (A[2] + A[1]) / 2;  
int i=3;  
A[i+1] = 12;
```

מערך הוא אובייקט לכל דבר (יש לו ייצוג במחשנית וגם ב – heap)  
כאשר אנו מגדירים מערך אנחנו מבצעים את השלבים הבאים :

Declaration  
Creation  
Assignment


```
Int[ ] Arr;  
Arr=new int[3];  
Arr[0]=7; Arr[1]=2; Arr[2]=12;
```



Declaration  
Creation  
Assignment

**או**

```
Int[ ] Arr=new int[3];  
Arr[0]=7; Arr[1]=2; Arr[2]=12;
```



Declaration + Creation  
Assignment

**או**

```
Int[ ] Arr={7,2,12};
```

Declaration + Creation + Initialization

# הדפסת תוכן של מערך ע"י שימוש בלולאה

```
public static void main( String[ ] args ) {  
    int[ ] Arr = {1 , 2 , 5 , 2 , 3};  
    for( int i=0 ; i < Arr.length ; i++)  
        System.out.println( Arr[ i ] );  
}
```

# אתחול מערך ע"י שימוש בלולאה גרסה 1

```
public static void main(String[] args){  
    int[ ] Arr = new int[5];  
    Scanner scan = new Scanner(System.in);  
  
    for( int i=0 ; i < 5 ; i++ ) {  
        System.out.print("Insert num: " + i);  
        Arr[i] = scan.nextInt();  
    }  
}
```

ההודעה insert num תוצג למשתמש 5  
פעמים (לפני כל קליטת נתון)



# אתחול מערך ע"י שימוש בלולאה גרסה 2

```
public static void main(String[] args){  
    int[] Arr = new int[5];  
    Scanner scan = new Scanner(System.in);  
  
    System.out.print("Insert 5 numbers:");  
    for(int i=0 ; i<5 ; i++) {  
        Arr[i] = scan.nextInt();  
    }  
}
```

ההודעה insert num תוצג למשתמש פעם אחת ואז המשתמש  
מכניס את 5 המספרים בשורה אחת כאשר המפריד ביניהם רווח

## כתוב תכנית אשר מגדירה 5 משתנים מסוג int - על התוכנית לקלוט מהמשתמש מספרים ולחשב את ממוצעם

```
public static void main(String[ ] args) {  
    int a0 , a1 , a2 , a3 , a4;  
    float avg;  
    Scanner scan = new Scanner(System.in);  
    System.out.println("Insert 5 integers:");  
    a0=scan.nextInt();  
    a1=scan.nextInt();  
    a2=scan.nextInt();  
    a3=scan.nextInt();  
    a4=scan.nextInt();  
    avg = (a0 + a1 + a2 + a3 + a4) / 5f;  
    System.out.println(avg);  
}
```

את המספרים יש לקלוט בשורה  
אחת כאשר המפריד בין כל  
מספר ומספר הינו רווח

# פתרון אלגנטי לבעיה מהעמוד הקודם ע"י שימוש במערך

```
public static void main(String[ ] args) {  
    final int N=5;  
    int[ ] arr = new int[N];  
    float sum= 0;  
    Scanner scan = new Scanner(System.in);  
    System.out.println("Insert "+N+" numbers:");  
    for(int i=0 ; i<N ; i++) {  
        arr[ i ] = scan.nextInt();  
        sum+= arr[i];  
    }  
    System.out.println(sum/N);  
}
```

מערך של N מספרים →

קליטת N מספרים לתוך המערך + חישוב הסכום {

בהינתן מערך arr של מספרים ומספר כלשהו num – כתוב  
תוכנית אשר בודקת אם המספר num נמצא במערך arr  
ע"י שימוש ב Linear Search\*\*\*

```
public static void main(String[] args){  
    int[ ] arr = {1 , 9 , 3 , 77 , 5};  
    int num = 77;  
    int i;  
    for(i=0 ; i < arr.length ; i++) {  
        if( arr[i] == num)  
            break;  
    }  
    if( i < arr.length)  
        System.out.println("The Number " + num + " is at place " + i);  
    else  
        System.out.println("Number doesn't exist");  
}
```

# Modification of Linear Search\*\*\*

```
public static void main(String[ ] args) {  
    int[ ] arr = {1 , 9 , 3 , 77 , 5};  
    int num = 77;  
    int i=0;  
    while(i<arr.length && arr[i] != num)  
        i++;  
    if( i < arr.length)  
        System.out.println("The Number " + num + " is at place " + i);  
    else  
        System.out.println("Number doesn't exist");  
}
```

# Binary Search\*\*\*

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

```
    int[ ] arr = {1,2,3,4,5,6,7};  
    int left =0 , right =arr.length-1;  
    int num=7, mid;
```

```
    while(left<=right) {  
        mid = (left+right)/2;  
        if(arr[mid]==num) {  
            System.out.println("Place:" + mid);  
            break;  
        }  
        else if(num<arr[mid])  
            right=mid-1;  
        else  
            left = mid +1;  
    }  
    if(left>right)  
        System.out.println("Not found");  
}
```

בהינתן מערך arr של מספרים ממויינים  
ומספר כלשהו num – כתוב תוכנית אשר  
בודקת אם המספר num נמצא במערך  
arr ע"י שימוש ב Binary Search

גרסה 1

# Binary Search\*\*\*

```
public static void main(String[ ] args){  
    int[ ] arr = {1 , 9 , 23 , 77 , 205};  
    int num =77;  
    int left=0;  
    int right = arr.length-1;  
  
    int middle=(left+right)/2;  
    while (left<=right && arr[middle] != num) {  
        if(num < arr[middle]){  
            right=middle;  
        } else if(num > arr[middle]){  
            left = middle + 1;  
        }  
        middle = (left+right)/2;  
    }  
    if(left <= right)  
        System.out.println("The Number " + num + " is at place " + middle);  
    else  
        System.out.println("Number doesn't exist");  
}
```

בהינתן מערך arr של מספרים ממויינים  
ומספר כלשהו num – כתוב תוכנית אשר  
בודקת אם המספר num נמצא במערך  
arr ע"י שימוש ב Binary Search

גרסה 2

```

public static void main(String[ ] args) {
    int[ ] arr = {1 , 9 , 23 , 77 , 205};
    int num = 77;
    int left=0;
    int right = arr.length-1;
    int middle;
    do {
        middle = (left+right)/2;
        if(num < arr[middle]){
            right=middle;
        } else if(num > arr[middle]){
            left = middle + 1;
        }
    } while (left<=right && arr[middle] != num);

    if(left <= right)
        System.out.println("The Number " + num + " is at place " + middle);
    else
        System.out.println("Number doesn't exist");
}

```

בהינתן מערך arr של מספרים ממויינים  
 ומספר כלשהו num – כתוב תוכנית אשר  
 בודקת אם המספר num נמצא במערך arr ע"י  
 שימוש ב Binary Search גרסה 3

גרסה 3



# מציאת המספר המקסימאלי במערך של מספרים

```
public static void main(String[ ] args) {  
    int[ ] arr = {3 , 9 , 2 , 7 , 27 , 0 , 9 , 1};  
    int max = arr[0];  
    for(int i=0 ; i < arr.length ; i++) {  
        if( arr[i] > max )  
            max = arr[i];  
    }  
    System.out.println(max);  
}
```

# Bubble Sort

## מיון בועות

מיון מערך  
בסדר עולה

```
public static void main(String[ ] args) {  
  
    int[ ] arr = {3,2,3,1,9,7,2,5};  
  
    for(int i=0;i<arr.length;i++) {  
        for(int j=0;j<arr.length-1-i;j++) {  
            if(arr[j]>arr[j+1]){  
                int tmp=arr[j];  
                arr[j]=arr[j+1];  
                arr[j+1]=tmp;  
            }  
        }  
    }  
}
```

מה הסיבוכיות ?  
תשובה  $O(n^2)$

# Different Members Of Array

```
public static void main (String[ ] args ) {  
    int j;  
    int[] arr={1,1,1,4,2,2,1,2,3,4,2,1,1,5};  
    for ( int i=0 ; i<arr.length ; i++) {  
        for ( j=0 ; j<i ;j++ ) {  
            if ( arr[ j ] == arr[i] )  
                break;  
        }  
        if( j==i )  
            System.out.print( arr[i] + ",");  
    }  
}
```

**Output : 1,4,2,3,5,**

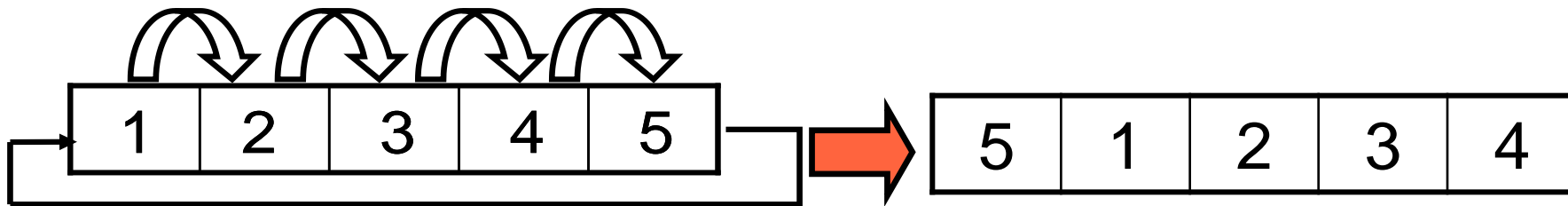
# Unique Members Of Array

```
public static void main (String[ ] args ) {  
    int j;  
    int[ ] arr={1,1,1,4,2,2,1,2,3,4,2,1,1,5};  
    for ( int i=0 ; i<Arr.length ; i++) {  
        for ( j=0 ; j<Arr.length ; j++) {  
            if ( j!=i && Arr[j]==Arr[i])  
                break;  
        }  
        if ( j==Arr.length )  
            System.out.print(Arr[i] + ",");  
    }  
}
```

**Output : 3,5,**

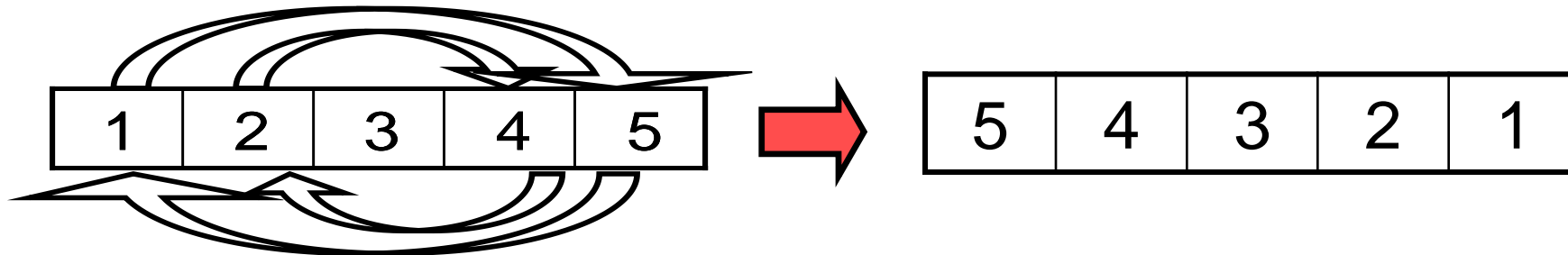
# הזזה לולאית

```
public static void main(String[ ] args) {  
    int[ ] arr = {1 , 2 , 3 , 4 , 5};  
    int temp = arr[ arr.length - 1 ];  
    for(int i=arr.length-1 ; i>0 ; i--) {  
        arr[i] = arr[i-1];  
    }  
    arr[0] = temp;  
}
```



# החלפה סימטרית

```
public static void main(String[ ] args) {  
    int[] arr = {1 , 2 , 3 , 4 , 5};  
    int temp;  
    for(int i=0 ; i<arr.length/2 ; i++) {  
        temp = arr[i];  
        arr[i] = arr[ arr.length - 1 - i];  
        arr[arr.length - 1 - i] = temp;  
    }  
}
```



```

public class Person {
    private String name;
    private String familyName;
    private int id;

    public Person(String n, String fn,int id) { this.id=id; name=n; familyName=fn; }

    public Person(Person other) {
        if(other != null )
        {
            id=other.id; name=other.name; familyName=other.familyName;
        } else {
            id=0; name = "?"; familyName = "?";
        }
    }

    public void setName(String n) { name=n; }
    public void setFamilyName(String fn) { familyName=fn; }
    public void setId(int id) { this.id=id; }

    public String getName() { return name; }
    public String getFamilyName() { return familyName; }
    public int getId() { return id; }

    public boolean equals(Person other) {
        return name.equals(other.name) &&
            familyName.equals(other.familyName) && id == other.id;
    }
    public String toString(){
        return familyName + " " + name + " " + id;
    }
}

```

נתונה המחלקה  
 כמו Person  
 מצד שמתואר  
 שמאל

# תרגיל 1

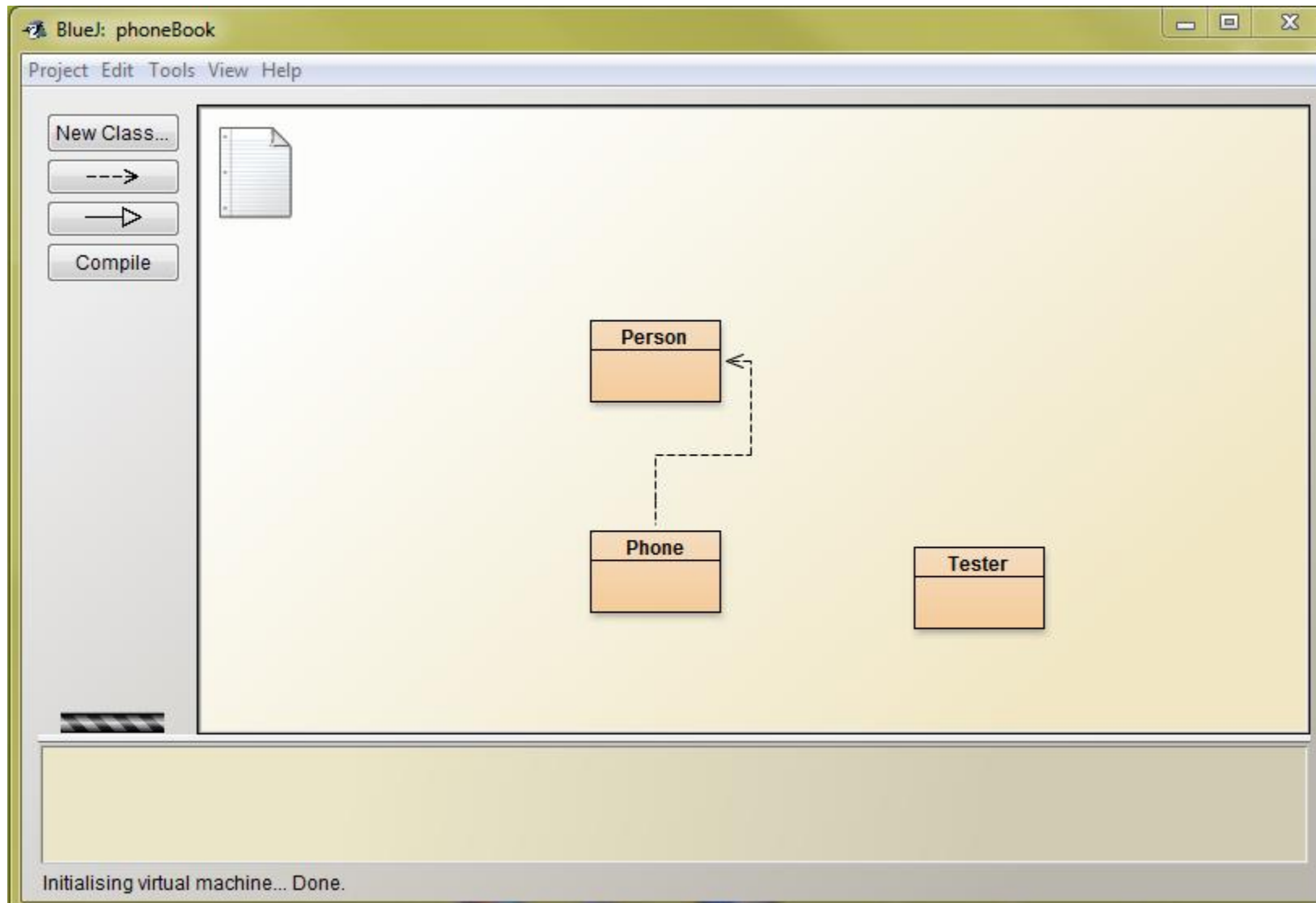
- ינגדיר את המחלקה Phone ע"י שני משתני מופע :  
ק ו- phoneNumber אשר מייצגים בנאדם ומספר הטלפון שלו , כמו שמתואר מצד שמאל.
- כמו כן הוגדרו שלושה בנאים : אחד שמקבל שם, שם משפחה, ת.ז. ומספר טלפון. השני מקבל בנאדם ומספר טלפון. והשלישי בנאי העתקה שמקבל אובייקט מסוג Phone ומשכפל אותו.
- בנוסף הוגדרה השיטה equals אשר מקבלת אובייקט Phone כפרמטר ובודקת אם הוא שווה לאובייקט שהופעלה עליו השיטה.
- בנוסף הוגדרו שיטות ה get ו- set בהתאם ואת השיטה toString.
- עליך להשלים את הבלוק של כל אחת מהשיטות שתיארנו.
- ניתן להיכזר ב Tester שמצד שמאל.

```
public class Phone {  
    private Person p;  
    private String phoneNumber;  
  
    public Phone(String n, String fn, int id, String pn) { .. }  
    public Phone(Person other , String pn){ .. }  
    public Phone(Phone other) { ... }  
  
    public Person getPerson() { .. }  
    public String getPhone() { .. }  
    public void setPerson(Person other) { .. }  
    public void setPhoneNumber(String pn) { .. }  
  
    public boolean equals (Phone other) { .. }  
    public String toString() { .. }  
}
```

```
public class Tester {  
    public static void main(String[ ] args) {  
        Phone ph1 = new Phone ("Alex","Ferro",111,"0545-555000");  
        Person p2 = new Person ("Avi","Dado",222);  
        Phone ph2 = new Phone (p2,"03-6351100");  
        Phone ph3 = new Phone(ph1);  
        System.out.println(ph3);  
        System.out.println(ph3.getPhone());  
        System.out.println(ph1.equals(ph3));  
    }  
}
```

Output:  
Shabo Alex 111 0545-555000  
0545-555000  
true





```
public class Phone {  
    private Person p;  
    private String phoneNumber;  
  
    public Phone(String n , String fn, int id,String pn){  
        p = new Person (n,fn,id);  
        phoneNumber = pn;  
    }  
    public Phone(Person other,String pn){  
        p = new Person (other);  
        phoneNumber = pn;  
    }  
    public Phone(Phone other) {  
        if (other != null ) {  
            p = new Person (other.p);  
            phoneNumber = other.phoneNumber;  
        }  
    }  
  
    public Person getPerson() { return new Person(p); }  
    public String getPhone() { return phoneNumber; }  
    public void setPerson(Person other) { p = new Person(other); }  
    public void setPhoneNumber(String pn) { phoneNumber = pn; }  
  
    public boolean equals (Phone other) {  
        return other != null && p.equals(other.p) && phoneNumber.equals(other.phoneNumber) ;  
    }  
    public String toString() { return p + " " + " " + phoneNumber; }  
}
```

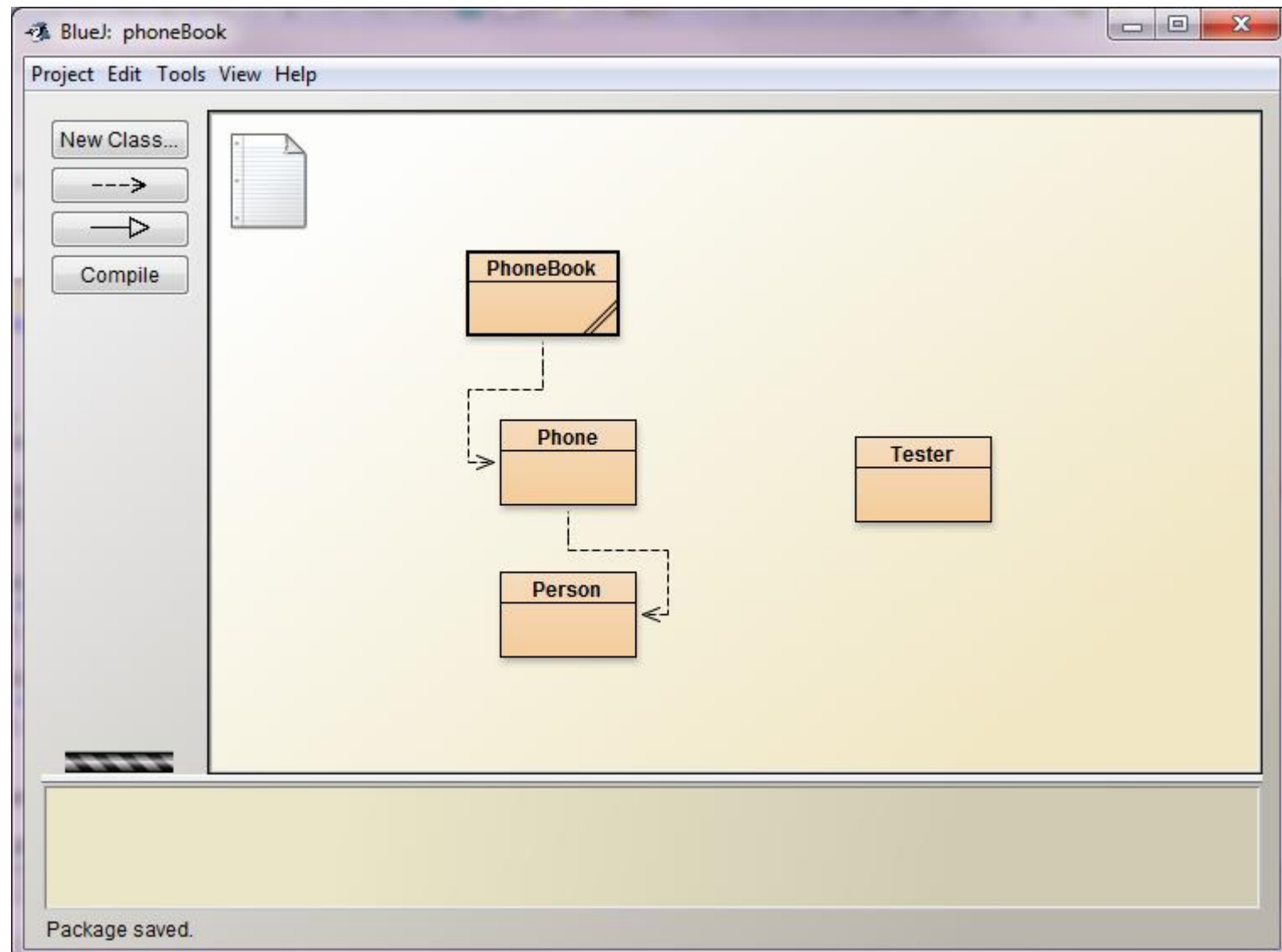
## תרגיל 2

■ נגדיר את המחלקה PhoneBook ע"י משתנה המופע  
phones אשר מייצג מערך של אובייקטים מסוג  
Phone, כמו שמתואר מצד שמאל.  
■ כמו כן הוגדרו 3 הבנאים כפי שמתואר מצד שמאל.  
בנוסף הוגדרו השיטות הבאות:  
■ שתי שיטות add אשר מוסיפות למערך (אם יש  
מקום) אובייקטים מסוג Phone כפי שמתואר.  
■ שתי שיטות delete אשר מוחקות את הייצוג של  
מספר במערך לפי סוג הפרמטר שנשלח (בנאדם או  
שם).  
■ searchPhoneByName אשר מקבלת שם  
ומחזירה את רשימת המספרים העונים לשם.  
■ toString אשר מחזירה מחרוזת שמייצגת את  
הרשימה של כל המספרים בספר הטלפונים.  
■ הערה: ניתן להיעזר בטסטר שמצד שמאל.

```
public class PhoneBook {  
    private Phone[ ] phones;  
    public PhoneBook(int n){..  
    public PhoneBook(Phone[ ] tmp){..  
    public PhoneBook(PhoneBook other) {..  
    public void add(String n,String fn,int id, String ph){..  
    public void add(Phone p){..  
    public void delete(Person p){..  
    public void delete(String name){..  
    public Phone[ ] searchPhoneByName(String name){..  
    public String toString(){..  
}
```

```
public class Tester {  
    public static void main(String[ ] args) {  
        PhoneBook myPhoneBook = new PhoneBook(5);  
        Person p1 = new Person ("Alex","Ferro",111);  
        Phone ph1 = new Phone (p1,"0545-555000");  
        myPhoneBook.add(ph1);  
        myPhoneBook.add("Dan" ,"Dado",222, "03-6351100");  
        myPhoneBook.add(new Phone(new Person("Shabi","Shabo",33),"0522-000000");  
        myPhoneBook.add("Dan" ,"Kabo",333, "03-6351234");  
        System.out.println(myPhoneBook);  
        Phone[ ] list = myPhoneBook.searchPhoneByName("Dan");  
        for(int i=0;list!=null && i<list.length && list[i]!=null;i++)  
            System.out.println(list[i]);  
        myPhoneBook.delete(p1);  
        System.out.println(myPhoneBook);  
        myPhoneBook.delete("Dan");  
        System.out.println(myPhoneBook);  
    }  
}
```

```
Ferro Alex 111 0545-555000  
Dado Dan 222 03-6351100  
Shabo Shabi 33 0522-000000  
Kabo Dan 333 03-6351234  
Dado Dan 222 03-6351100  
Kabo Dan 333 03-6351234  
Dado Dan 222 03-6351100  
Shabo Shabi 33 0522-000000  
Kabo Dan 333 03-6351234  
Shabo Shabi 33 0522-000000
```



```
public class PhoneBook {  
    private Phone [ ] phones;  
  
    public PhoneBook(int n) {  
        phones=new Phone[n];  
    }  
  
    public PhoneBook(Phone[ ] tmp) {  
        if( tmp != null ) {  
            phones = new Phone[tmp.length];  
            for(int i = 0 ; i<tmp.length; i++) {  
                phones[i] = new Phone(tmp[i]);  
            }  
        }  
    }  
  
    public PhoneBook(PhoneBook other) {  
        if( other != null ) {  
            phones = new Phone[other.phones.length];  
            for(int i = 0 ; i<phones.length; i++) {  
                phones[i] = new Phone(other.phones[i]);  
            }  
        }  
    }  
    :  
}
```

```
public void add(String n, String fn, int id, String ph) {  
    int i=0;  
    while(i<phones.length && phones[i]!=null) i++;  
    if(i<phones.length) // check if PhoneBook is not Full  
        phones[i]=new Phone(n, fn, id, ph);  
}  
public void add(Phone p) {  
    if (p != null) {  
        int i=0;  
        while(i<phones.length && phones[i]!=null) i++;  
        if(i<phones.length) // check if PhoneBook is not Full  
            phones[i]=new Phone(p);  
    }  
}  
public void delete(Person p) {  
    for (int i=0;i<phones.length;i++){  
        if(phones[i]!=null && p.equals(phones[i].getPerson())){  
            phones[i]=null;  
        }  
    }  
}  
public void delete(String name) {  
    for (int i=0;i<phones.length;i++){  
        if(phones[i]!=null && name.equals(phones[i].getPerson().getName())){  
            phones[i]=null;  
        }  
    }  
}
```

```
public Phone[ ] searchPhoneByName(String name) {  
    Phone[] list = new Phone[phones.length];  
    int j=0;  
    for(int i=0;i<phones.length;i++){  
        if(phones[i]!=null && name.equals(phones[i].getPerson().getName())){  
            list[j++] = new Phone(phones[i]);  
        }  
    }  
    return list;  
}
```

```
public String toString(){  
    String s="";  
    int i=0;  
    while(i<phones.length){  
        if(phones[i] != null)  
            s = s + phones[i] + "\n";  
        i++;  
    }  
    return s;  
}  
}
```

***END***