دانشکده مهندسی و علوم کامپیوتر

برنامه نویسی پیشرفته

وحیدی اصل

آرایه ها -۱

انگیزه

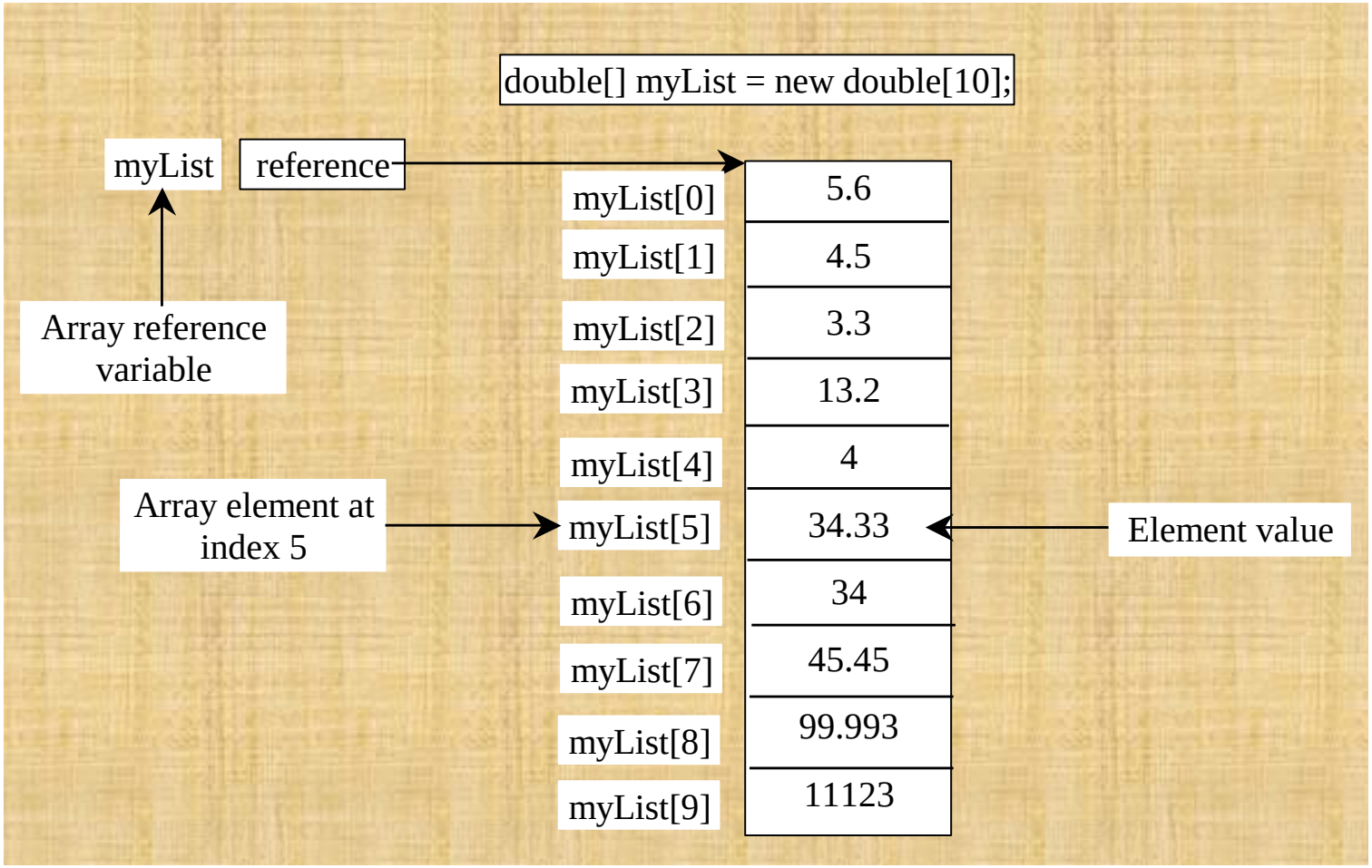
- ۱۰۰۰ عدد را از کاربر بگیرید، میانگین آنها را حساب کرده و بگویید چه تعدادی از آنها از میانگین بزرگتر هستند؟

راه حل

```
public class AnalyzeNumbers {
  public static void main(String[] args) {
    final int NUMBER_OF_ELEMENTS = 10;
    double[] numbers = new double[NUMBER_OF_ELEMENTS];
    double sum = 0;

    java.util.Scanner input = new java.util.Scanner(System.in);
    for (int i = 0; i < NUMBER_OF_ELEMENTS; i++) {
      System.out.print("Enter a new number: ");
      numbers[i] = input.nextDouble();
      sum += numbers[i];
    }

    double average = sum / NUMBER_OF_ELEMENTS;

    int count = 0; // The number of elements above average
    for (int i = 0; i < NUMBER_OF_ELEMENTS; i++)
      if (numbers[i] > average)
        count++;

    System.out.println("Average is " + average);
    System.out.println("Number of elements above the average "
      + count);
  }
}
```

معرفی آرایه ها

- آرایه ساختمان داده ای است حاوی مجموعه ای از داده های هم نوع



مراحل تعریف آرایه ها

1 Declare an int array variable. An array variable is a remote control to an array object.

```
int[] nums;
```

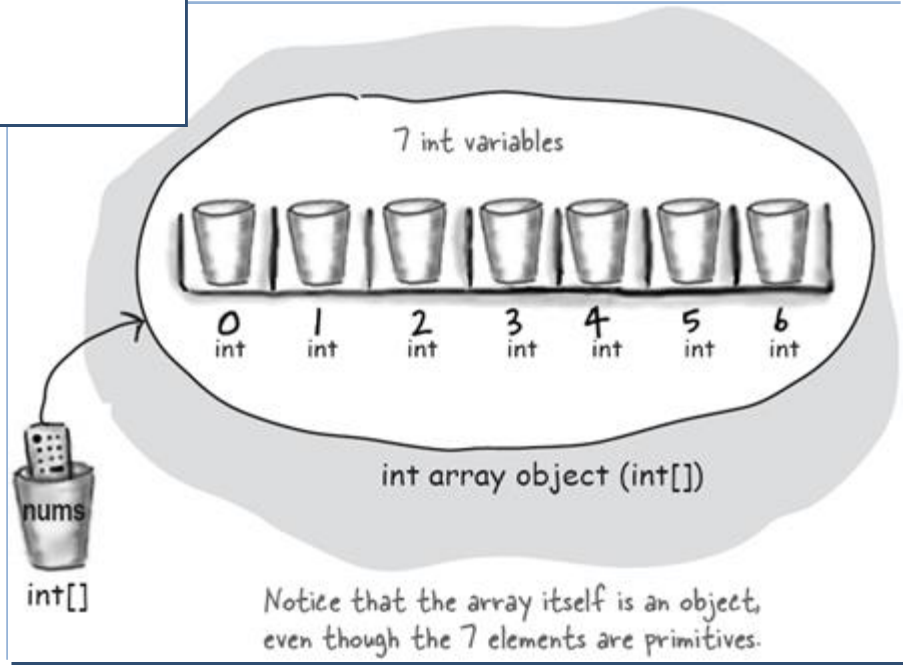
2 Create a new int array with a length of 7, and assign it to the previously-declared `int[]` variable `nums`

```
nums = new int[7];
```

3 Give each element in the array an int value.
Remember, elements in an int *array* are just int *variables*.

7 int variables

```
nums[0] = 6;
nums[1] = 19;
nums[2] = 44;
nums[3] = 42;
nums[4] = 10;
nums[5] = 20;
nums[6] = 1;
```



ایجاد آرایه ها

```
arrayRefVar = new datatype[arraySize];
```

Example:

```
myList = new double[10];
```

`myList[0]` اشاره به اولین عنصر آرایه

`myList[9]` اشاره به آخرین عنصر آرایه

اعلان و ایجاد آرایه ها در یک گام

- ```
  datatype[] arrayRefVar = new
     datatype[arraySize];
  ```

```
double[] myList = new double[10];
```

- ```
  datatype arrayRefVar[] = new
    datatype[arraySize];
  ```

```
double myList[] = new double[10];
```

اندازه آرایه

• یک بار که آرایه ایجاد شود، اندازه آن ثابت می شود. این اندازه دیگر قابل تغییر نمی باشد. اندازه آرایه با دستور زیر محاسبه می شود:

```
arrayRefVar.length
```

•برای مثال اعمال دستور بالا برروی آرایه myList که در اسلاید قبل با ۱۰ عنصر تعریف شده است، مقدار ۱۰ را برمی گرداند.

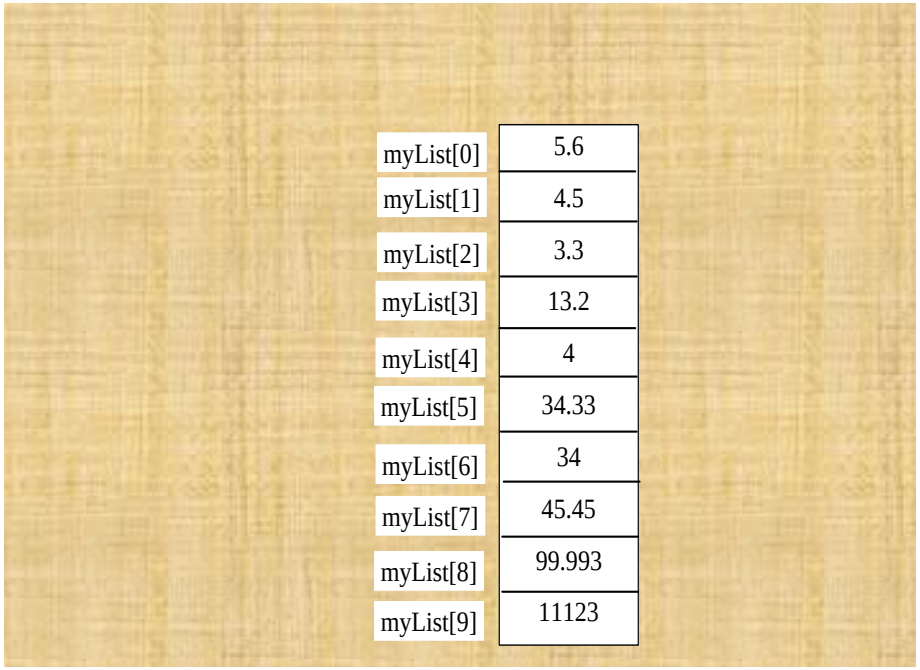
```
myList.length returns 10
```

مقادیر پیش فرض

- هرگاه آرایه ای ایجاد شود، به عناصر آن مقدار اولیه پیش فرض داده می شود:
  - صفر برای انواع داده اصلی عددی
  - '\u0000' برای نوع کاراکتری
  - false برای نوع boolean

متغیرهای اندیس

- عناصر آرایه از طریق اندیس قابل دسترسی هستند.
- اندیسهای آرایه مبتنی بر 0 (*0-based*) هستند:
  - اولین خانه آنها دارای اندیس صفر و خانه آخر دارای اندیس arrayRefVar.length-1 می باشد.
  - مثال نشان داده شده در شکل،آرایه myList را نشان می دهد که ۱۰ مقدار نوع double و اندیسها از صفر تا ۹ هستند.



- دسترسی به هر عنصر آرایه که به آن متغیراندیس دار گفته می شود، با دستور زیر انجام می شود:

```
arrayRefVar[index];
```

- پس از اینکه یک آرایه ایجاد شد، با یک متغیر اندیس دار (هریک از خانه های آرایه) مانند یک متغیر عادی رفتار می شود.
- برای مثال، کد زیر مقدار دو عنصر آرایه (با اندیس ۰) و (با اندیس ۱) را جمع کرده و در عنصر دیگر (با اندیس ۲) قرار می دهد:

```
myList[2] = myList[0] + myList[1];
```

- اعلان، ایجاد، و مقداردهی اولیه آرایه می تواند در یک گام انجام شود:

```
double[] myList = {1.9, 2.9, 3.4, 3.5};
```

- تمامی این اعمال در یک دستور انجام می شود.

```
double[] myList = {1.9, 2.9, 3.4, 3.5};
```

- دستور بالا معادل دستورات زیر است:

```
double[] myList = new double[4];
myList[0] = 1.9;
myList[1] = 2.9;
myList[2] = 3.4;
myList[3] = 3.5;
```

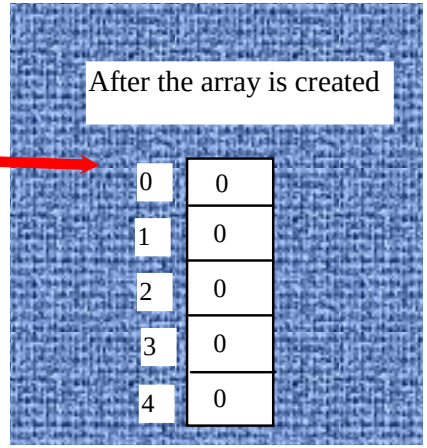
دقت کنید!

- استفاده از یک دستور برای مراحل اعلان، ایجاد و مقداردهی آرایه در بیشتر مواقع توصیه می شود.
- توزیع این مراحل در چند دستور باید با دقت انجام شود. دستور زیر نادرست است:

```
double[] myList;

myList = {1.9, 2.9, 3.4, 3.5};
```

Declare array variable values, create an array, and assign its reference to values

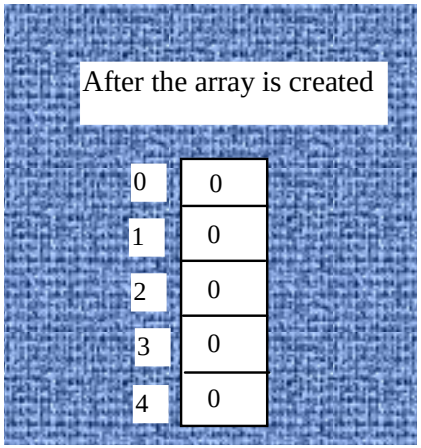
```
public class Test {
  public static void main(Stri[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



ردگیری برنامه حاوی آرایه

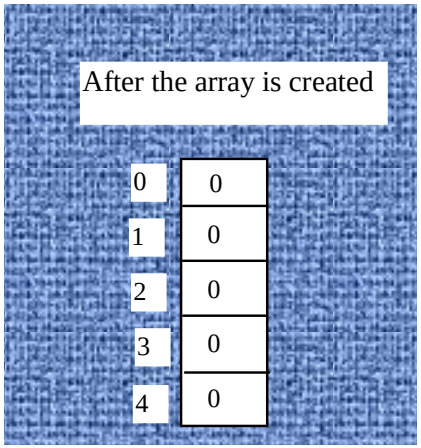
i becomes 1

```
public class Test {
  public static void main(String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



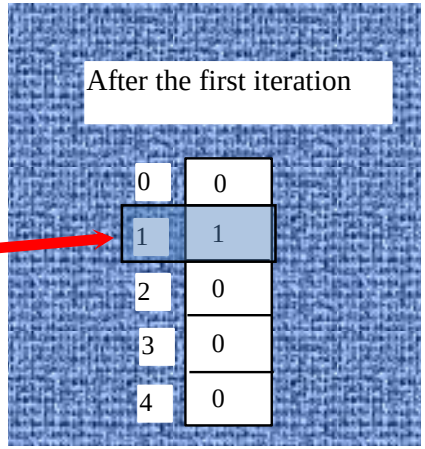
i (=1) is less than 5

```
public class Test {
  public static void ma (String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



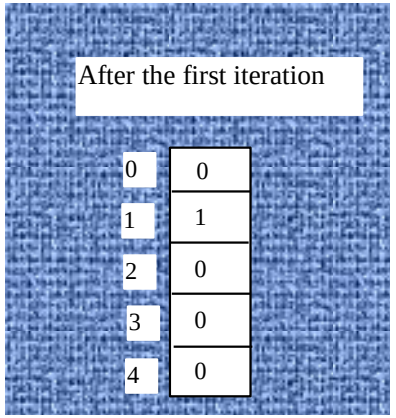
After this line is executed, value[1] is 1

```
public class Test {
  public static void main(Strin      gs) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



After i++, i becomes 2

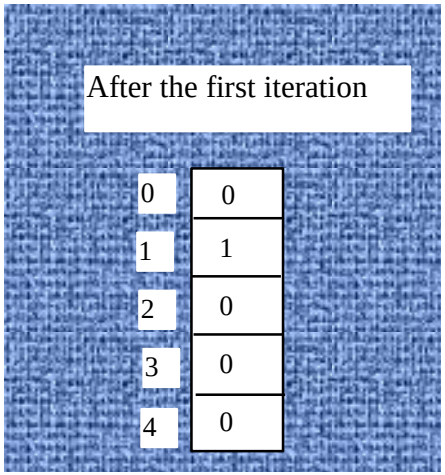
```
public class Test {
  public static void main(String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



ردگیری برنامه حاوی آرایه

```
public class Test {
  public static void main(String[]
      args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] +
      values[4];
  }
}
```

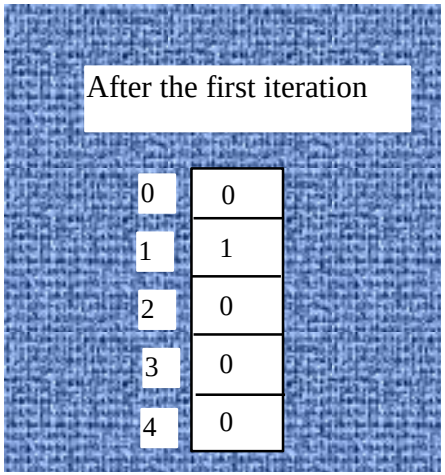
i (= 2) is less than 5



ردگیری برنامه حاوی آرایه

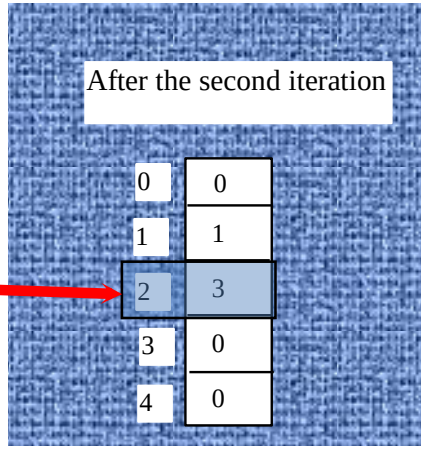
```
public class Test {
  public static void main(String[]
        args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] +
        values[4];
  }
}
```

i (= 2) is less than 5



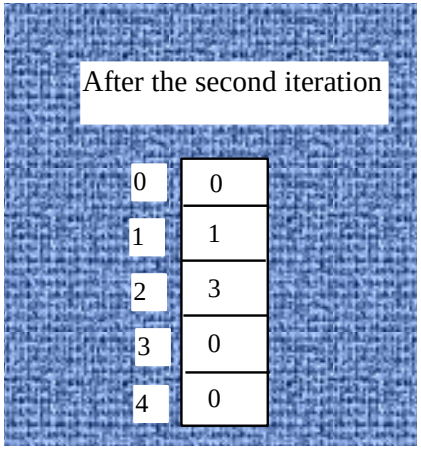
After this line is executed,
values[2] is 3 (2 + 1)

```
public class Test {
  public static void main(Strin      gs) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



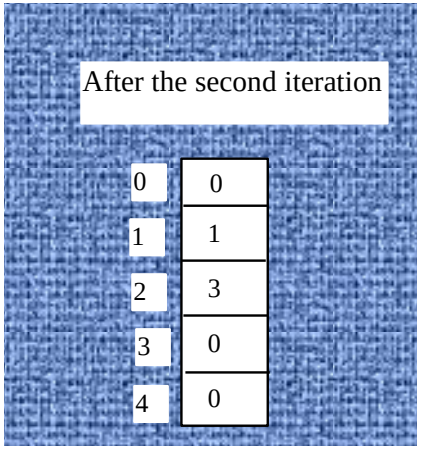
After this, i becomes 3.

```
public class Test {
  public static void main(S    g[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



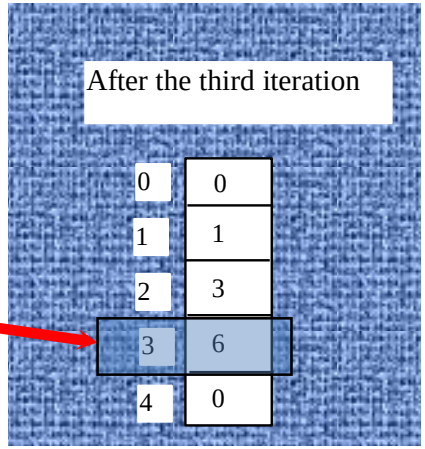
i (=3) is still less than 5.

```
public class Test {
  public static void main(String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



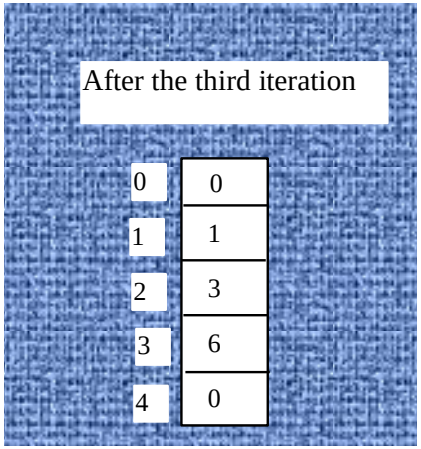
After this line, values[3] becomes 6 (3 + 3)

```
public class Test {
  public static void main(String[] args)
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



After this, i becomes 4

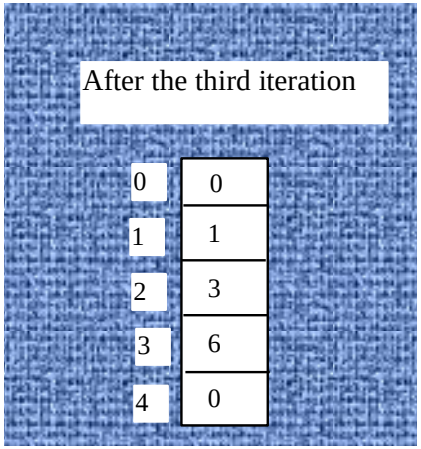
```
public class Test {
  public static void main(String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



ردگیری برنامه حاوی آرایه

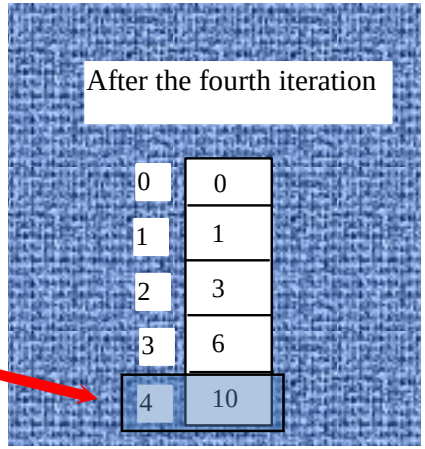
i (=4) is still less than 5

```
public class Test {
  public static void main(String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



After this, values[4] becomes 10 (4 + 6)

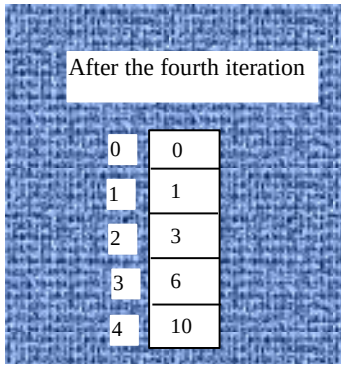
```
public class Test {
  public static void main(Strir  rgs) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



ردگیری برنامه حاوی آرایه

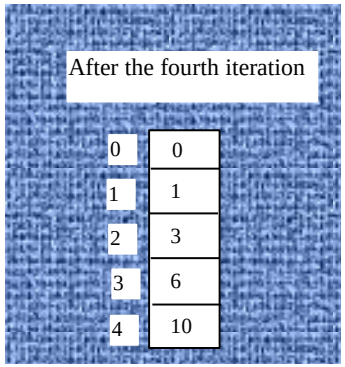
After i++, i becomes 5

```
public class Test {
  public static void main(String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



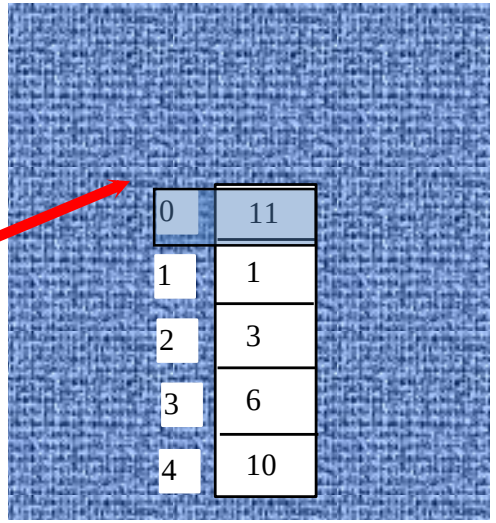
i (=5) < 5 is false. Exit the loop

```
public class Test {
  public static void main(String[] args) {
    int[] values = new int[5];
    for (int i = 1; i < 5; i++) {
      values[i] = i + values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



After this line, values[0] is 11 (1 + 10)

```
public class Test {
  public static void ma      ing[] args) {
    int[] values = nev      [5];
    for (int i = 1; i <      i++) {
      values[i] = i      values[i-1];
    }
    values[0] = values[1] + values[4];
  }
}
```



- در مثالهای نشان داده شده در اسلایدهای بعدی، روشهای مختلف پردازش آرایه ها بیان شده است.

1. مقداردهی اولیه آرایه ها با مقادیر ورودی
2. مقداردهی اولیه آرایه ها با مقادیر تصادفی
3. چاپ آرایه ها
4. جمع کردن همه عناصر آرایه
5. یافتن بزرگترین مقدار
6. یافتن اندیس عنصر حاوی بزرگترین مقدار
7. بُرزدن (شافلینگ) تصادفی
8. شیفت عناصر

مقداردهی اولیه آرایه ها با مقادیر ورودی

```
java.util.Scanner input = new java.util.Scanner(System.in);
System.out.print("Enter " + myList.length + " values: ");
for (int i = 0; i < myList.length; i++)
  myList[i] = input.nextDouble();
```

```
for (int i = 0; i < myList.length; i++) {
  myList[i] = Math.random() * 100;
}
```

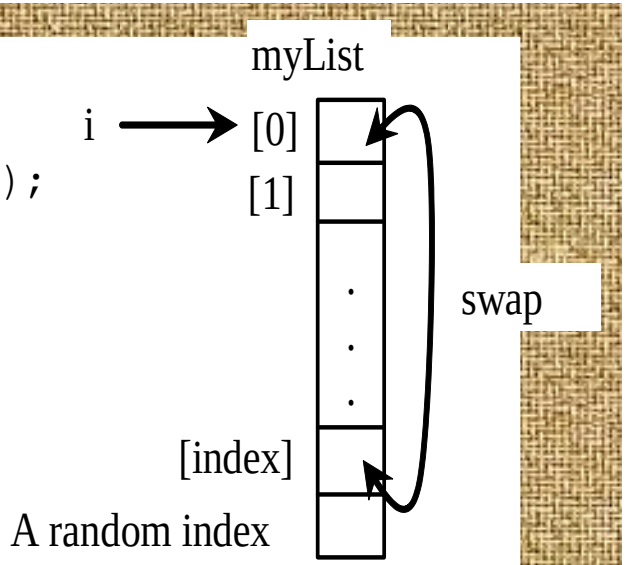
```
for (int i = 0; i < myList.length; i++) {
  System.out.print(myList[i] + " ");
}
```

جمع کردن همه عناصر آرایه

```
double total = 0;
for (int i = 0; i < myList.length; i++) {
  total += myList[i];
}
```

```
double max = myList[0];
for (int i = 1; i < myList.length; i++) {
  if (myList[i] > max) max = myList[i];
}
```

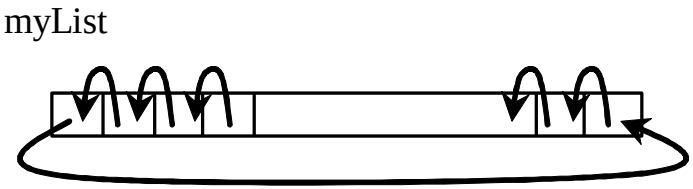
```
for (int i = 0; i < myList.length; i++) {
  // Generate an index j randomly
  int index = (int)(Math.random()* myList.length);

  // Swap myList[i] with myList[j]
  double temp = myList[i];
  myList[i] = myList[index];
  myList[index] = temp;
}
```



شیفت چرخشی عناصر-شیفت چپ

- فرض کنید خانه صفر آرایه، سمپ چپ لیست قرار گرفته است.

```
double temp = myList[0]; // Retain the first element

// Shift elements left
for (int i = 1; i < myList.length; i++) {
  myList[i - 1] = myList[i];
}

// Move the first element to fill in the last position
myList[myList.length - 1] = temp;
```



حلقه for توسعه یافته

- Jdk یک ساختار for معرفی کرده که به شما امکان می دهد، یک آرایه کامل را بدون استفاده از اندیس آرایه، خانه به خانه پیمایش کنید.
- برای مثال، کد زیر تمامی عناصر در آرایه myList را نمایش می دهید:

```
for (double value: myList)
     System.out.println(value);
```

- در حالت کلی قاعده نحوی به فرم زیر است:

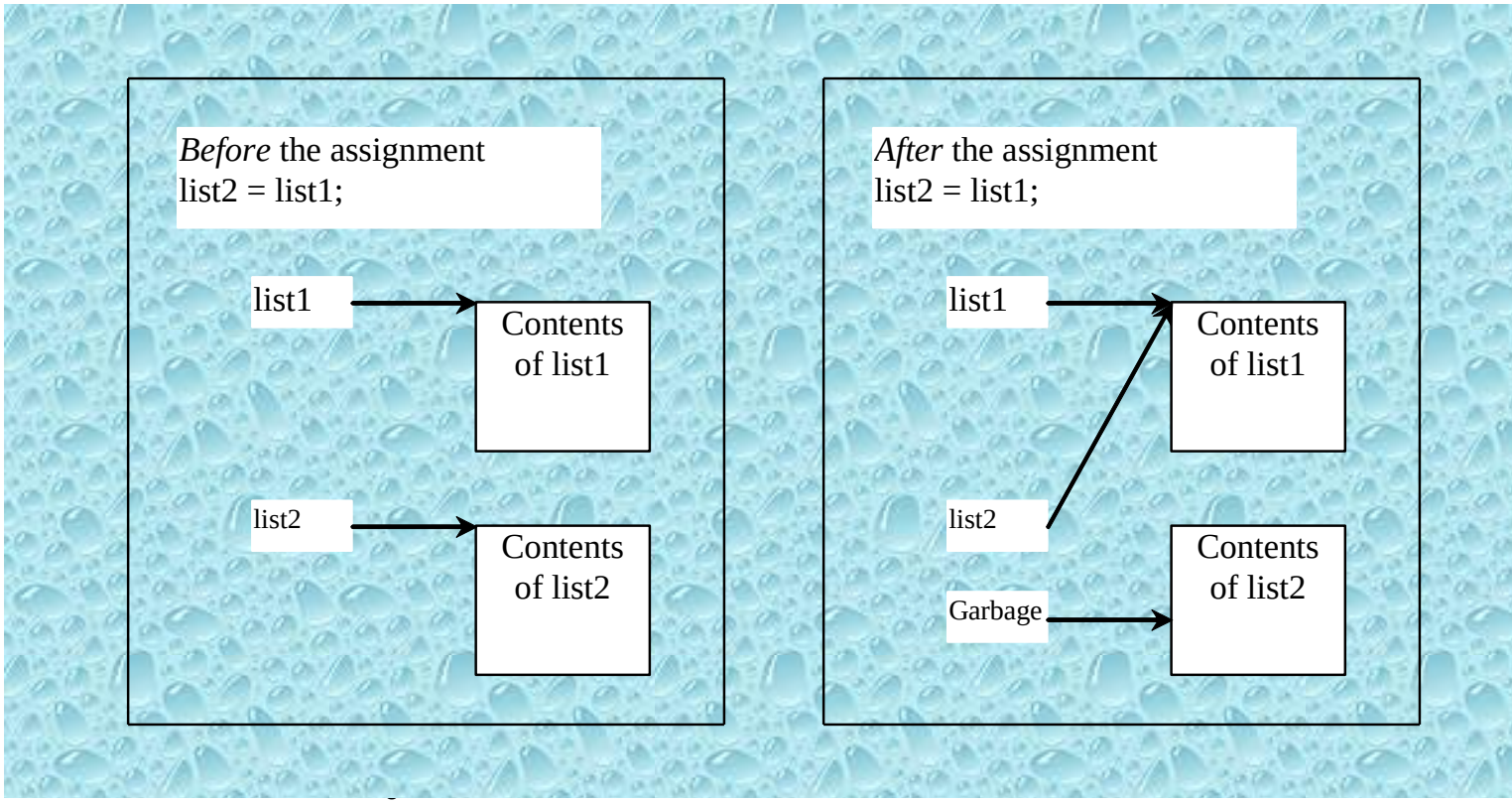
```
for (elementType value: arrayRefVar) {
  // Process the value
}
```

- اگر بخواهید عناصر را به ترتیب دیگری پیمایش کنید یا محتوای عناصر را به شکل دیگری تغییر دهید، باید از اندیس استفاده کنید!

کپی آرایه ها

- اغلب، در یک برنامه می خواهیم محتوای یک آرایه را در یک آرایه دیگر کپی کنیم.
- در این مواقع ممکن است از = به صورت زیر استفاده کنید، که نتیجه ای خلاف انتظار شما خواهد داشت!

```
list2 = list1;
```



کپی آرایه ها

استفاده از حلقه:

```
int[] sourceArray = {2, 3, 1, 5, 10};
int[] targetArray = new
  int[sourceArray.length];

for (int i = 0; i < sourceArrays.length; i++)
   targetArray[i] = sourceArray[i];
```

عملیات arraycopy برای کپی آرایه

```
arraycopy(sourceArray, src_pos,
  targetArray, tar_pos, length);
```

- مثال:

```
System.arraycopy(sourceArray, 0,
  targetArray, 0, sourceArray.length);
```

ارسال آرایه ها به متدها

```
public static void printArray(int[] array) {
  for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
  }
}
```

Invoke the method

```
int[] list = {3, 1, 2, 6, 4, 2};
printArray(list);
```

Invoke the method

```
printArray(new int[]{3, 1, 2, 6, 4, 2});
```

Anonymous array

آرایه بی نام (Anonymous)

- دستور

```
printArray(new int[]{3, 1, 2, 6, 4, 2});
```

- با قاعده نحوی زیر یک آرایه ایجاد می کند:

```
new dataType[]{literal0, literal1, ..., literalk};
```

- در اینجا یک متغیر ارجاعی دارای نام برای آرایه وجود ندارد. به چنین آرایه ای، آرایه بی نام گفته می شود.

ارسال با مقدار

•جاوا از سازوکار ارسال با مقدار برای ارسال پارامترها به یک متد استفاده می کند. تفاوتهای مهمی بین ارسال مقدار متغیرهای انواع اصلی و ارسال آرایه ها وجود دارد.

•برای پارامتری از نوع اولیه، مقداری که ارسال می شود تنها در داخل متد ممکن است تغییر داده شود، اما تاثیری بر متغیری که مقدار آن به متد ارسال شده است، ندارد.

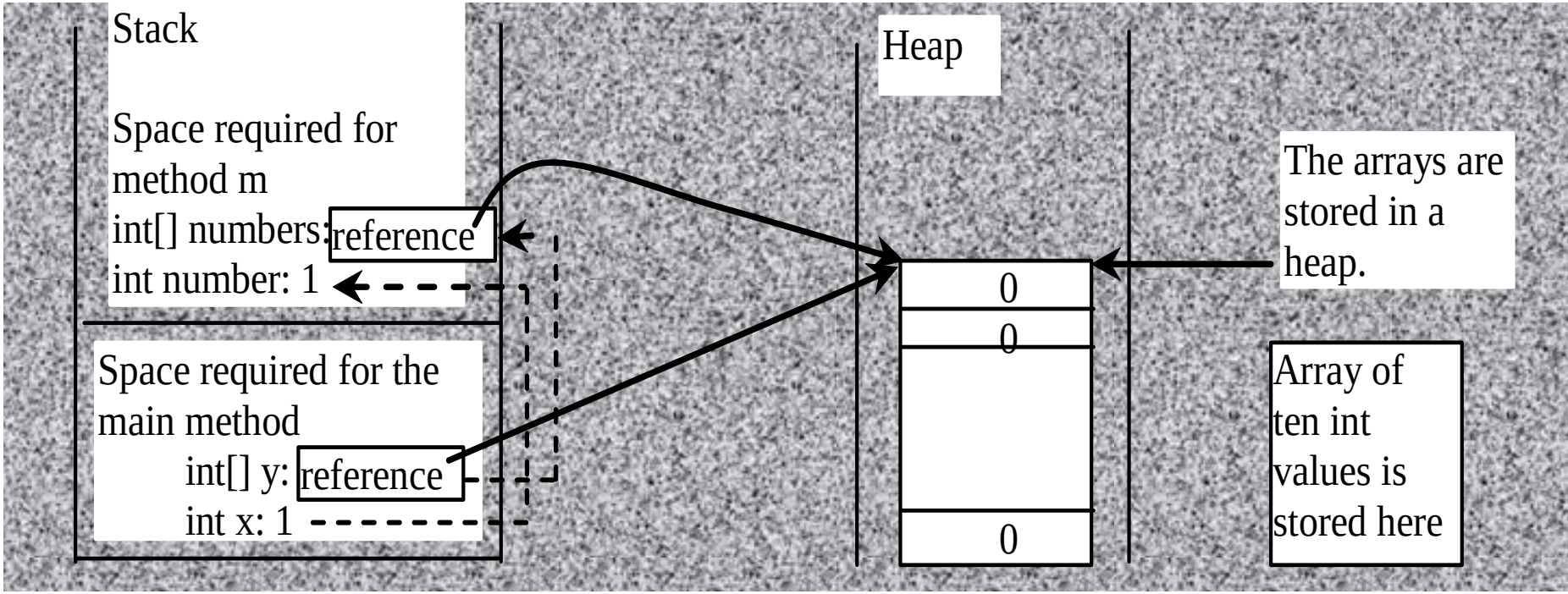
•برای پارامتری از نوع آرایه، مقدار پارامتر ارسال شده، ارجاعی به یک آرایه است (یک کپی از ریموت کنترلی که به آرایه اشاره می کند، به متد ارسال می شود).

بنابراین هر تغییری برروی آرایه در داخل متد، بر آرایه اصلی که به متد ارسال شده، تاثیر خواهد گذاشت.

مثالی ساده

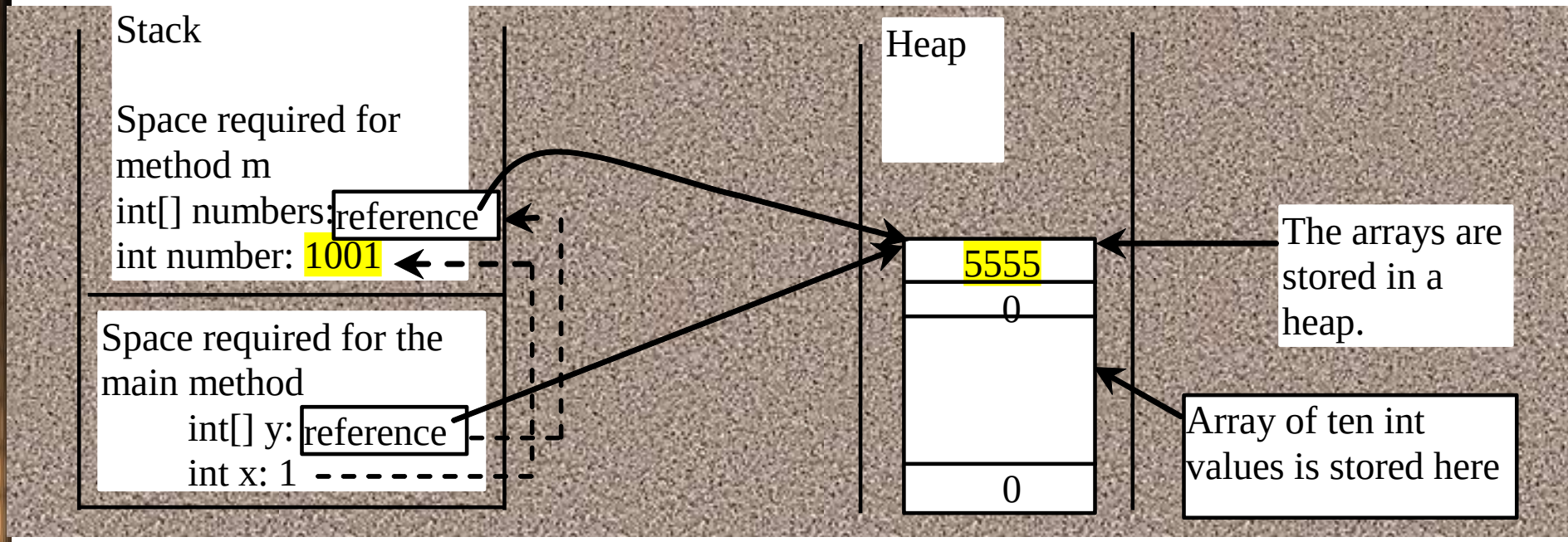
```
public class Test {
  public static void main(String[] args) {
    int x = 1; // x represents an int value
    int[] y = new int[10]; //y represents an array of int values

    m(x, y); // Invoke m with arguments x and y

    System.out.println("x is " + x);
    System.out.println("y[0] is " + y[0]);
  }

  public static void m(int number, int[] numbers) {
    number = 1001; // Assign a new value to number
    numbers[0] = 5555; // Assign a new value to numbers[0]
  }
}
```

پشته فراخوانی مثال قبل

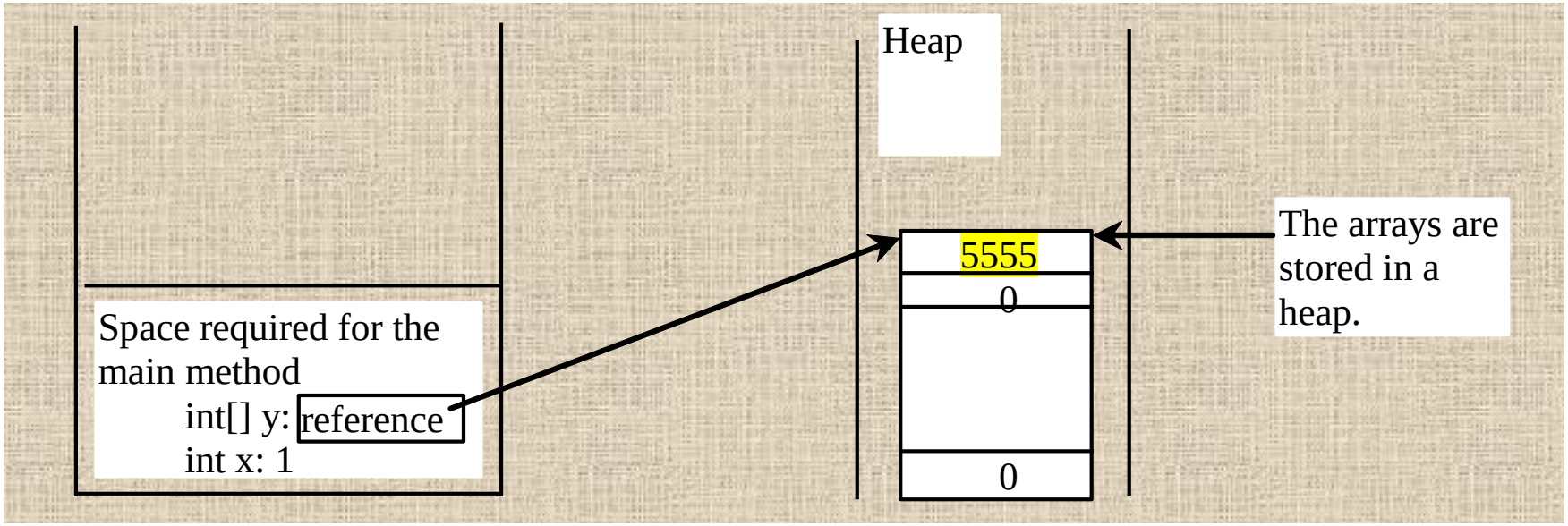


- در هنگام فراخوانی m(x,y)، مقادیر x و y به ترتیب به number و numbers ارسال می شوند. چون y حاوی مقدار ارجاعی به آرایه است، numbers به همان آرایه ای ارجاع می کند که y به آن ارجاع دارد.

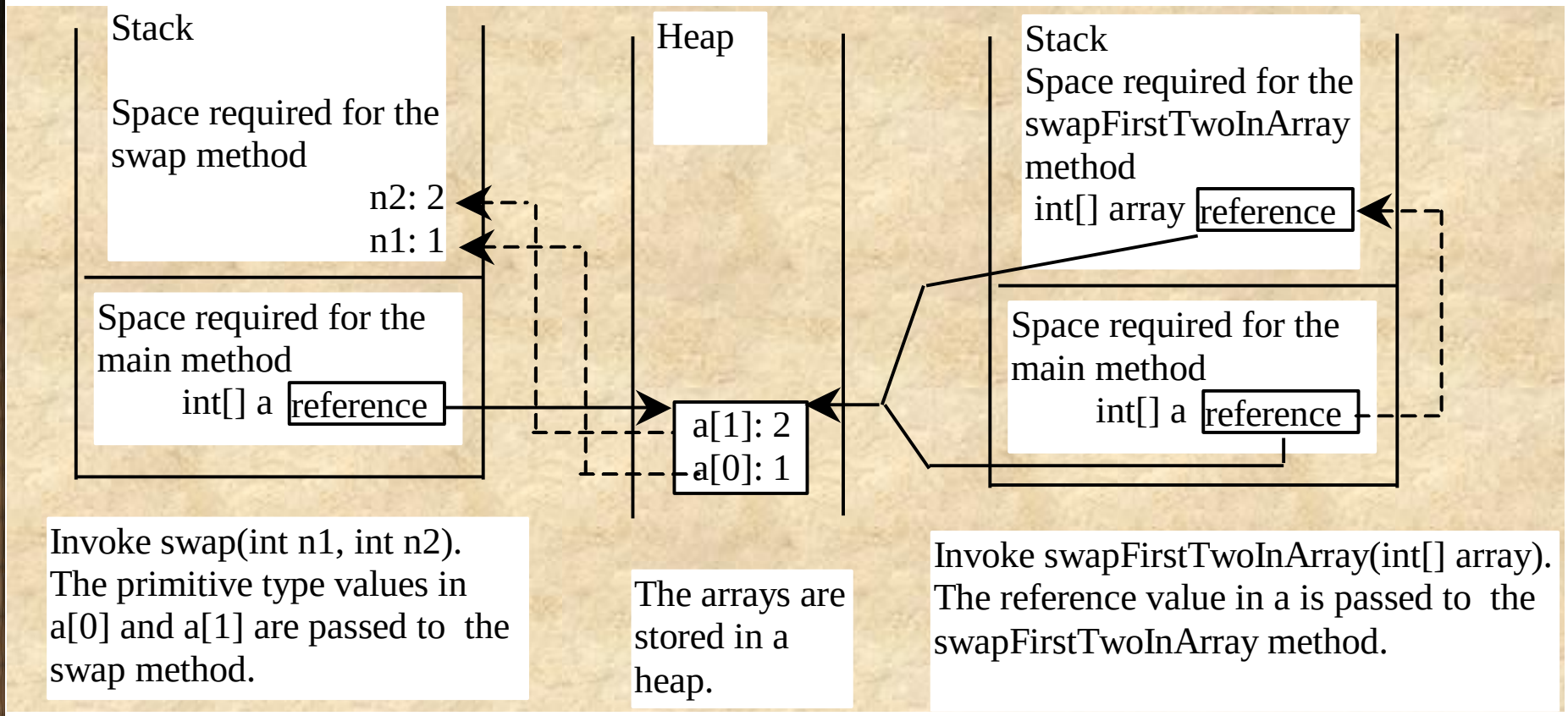
پشته فراخوانی مثال قبل



حافظه Heap- مثال قبل

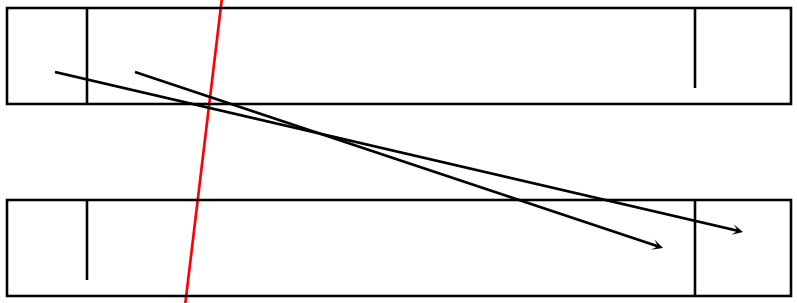


مثال–جابجایی دو عنصر آرایه



برگرداندن آرایه از متد

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```



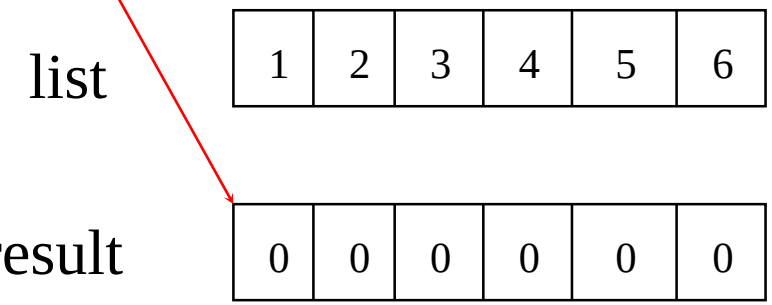

```
int[] list1 = {1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

ردگیری متدمعکوس آرایه

```
int[] list1 = {1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

Declare result and create array

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

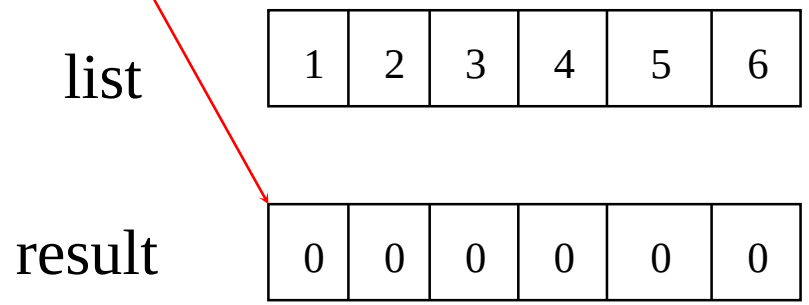


ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i = 0 and j = 5

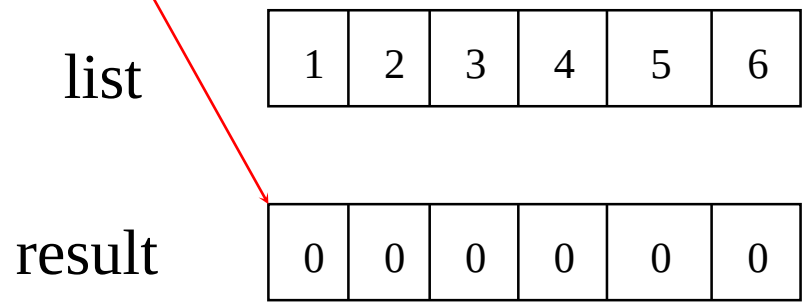


ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i (= 0) is less than 6

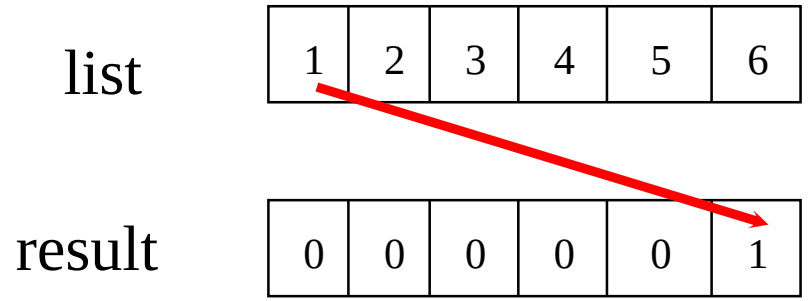


ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i = 0 and j = 5
Assign list[0] to result[5]



ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

After this, i becomes 1 and j becomes 4

list	1	2	3	4	5	6

result	0	0	0	0	0	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i (=1) is less than 6

list

1	2	3	4	5	6

result

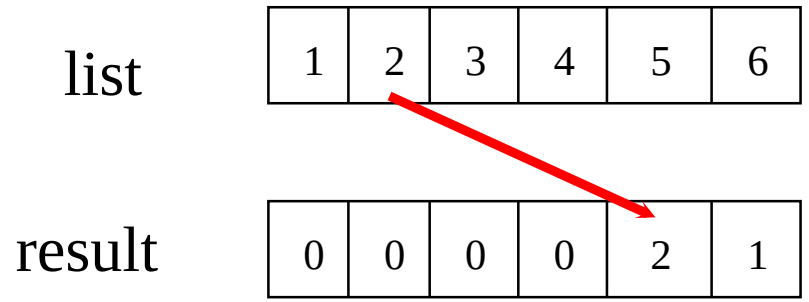
0	0	0	0	0	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i = 1 and j = 4
Assign list[1] to result[4]



ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

After this, i becomes 2 and j becomes 3

list	1	2	3	4	5	6

result	0	0	0	0	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i (=2) is still less than 6

list

1	2	3	4	5	6

result

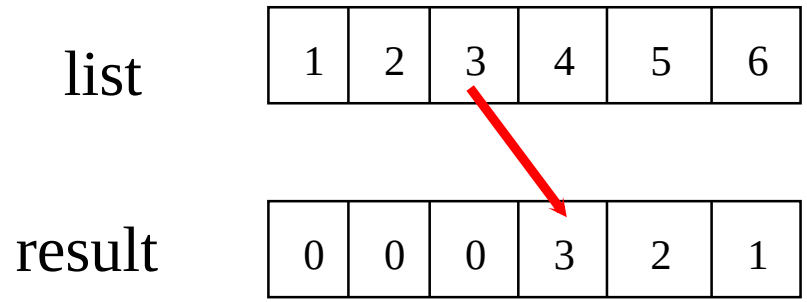
0	0	0	0	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i = 2 and j = 3
Assign list[i] to result[j]



ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

After this, i becomes 3 and j becomes 2

list	1	2	3	4	5	6

result	0	0	0	3	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i (=3) is still less than 6

list	1	2	3	4	5	6

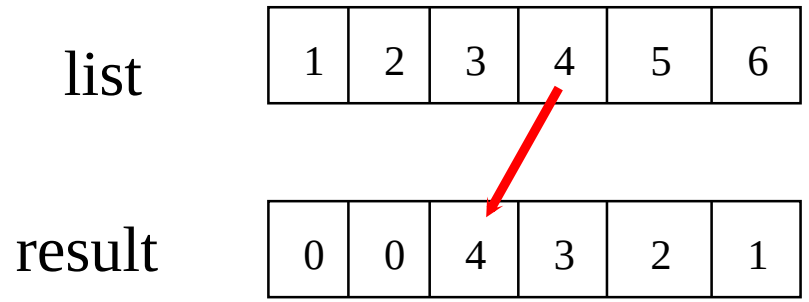
result	0	0	0	3	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i = 3 and j = 2
Assign list[i] to result[j]



ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

After this, i becomes 4 and j becomes 1

list	1	2	3	4	5	6

result	0	0	4	3	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i (=4) is still less than 6

list

1	2	3	4	5	6

result

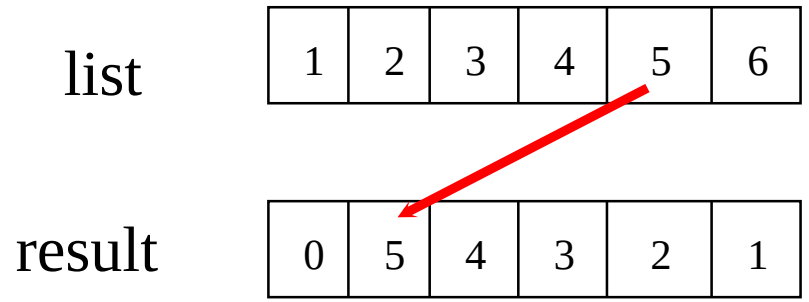
0	0	4	3	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i = 4 and j = 1
Assign list[i] to result[j]



ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

After this, i becomes 5 and j becomes 0

list	1	2	3	4	5	6

result	0	5	4	3	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i (=5) is still less than 6

list

1	2	3	4	5	6

result

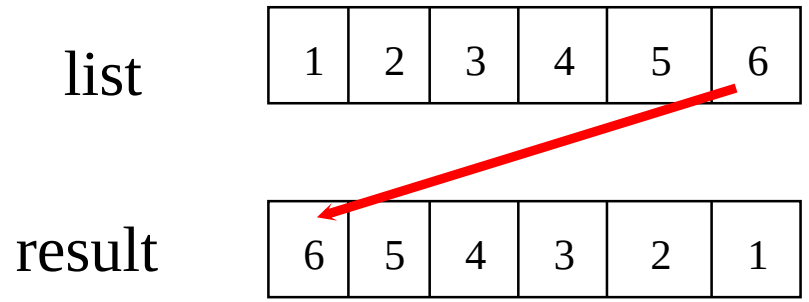
0	5	4	3	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i = 5 and j = 0
Assign list[i] to result[j]



ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

After this, i becomes 6 and j becomes -1

list	1	2	3	4	5	6

result	6	5	4	3	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

i (=6) < 6 is false. So exit the loop.

list	1	2	3	4	5	6

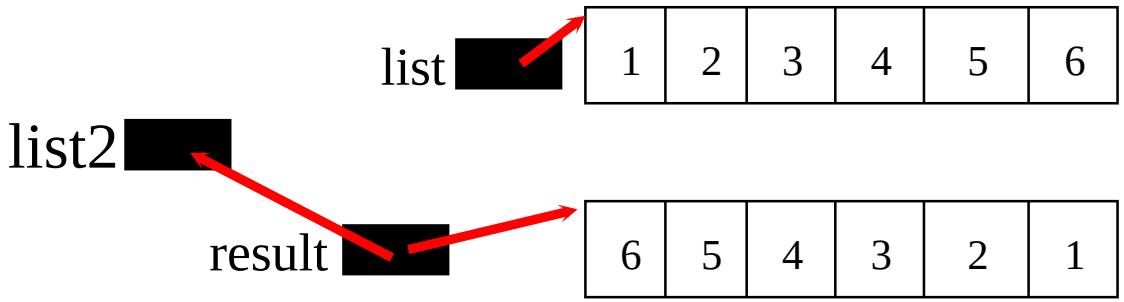
result	6	5	4	3	2	1

ردگیری متدمعکوس آرایه

```
int[] list1 = new int[]{1, 2, 3, 4, 5, 6};
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {
  int[] result = new int[list.length];

  for (int i = 0, j = result.length - 1;
       i < list.length; i++, j--) {
    result[j] = list[i];
  }

  return result;
}
```

Return result



مسئله: شمارش تعداد مشاهدات یک حرف

- ۱۰۰ حرف کوچک به طور تصادفی تولید کنید و آنها را در آرایه ای از کاراکترها قرار دهید.
- تعداد مشاهده هر حرف در آرایه را شمارش کنید.

مسئله: شمارش تعداد مشاهدات یک حرف

```
public class CountLettersInArray {
  /** Main method */
  public static void main(String args[]) {
    // Declare and create an array
 char[] chars
={'t','h','i','s','i','s','n','e','w','s','a','m','p','l','e','o','f','j','a','v','a'};

    // Display the array
    System.out.println("The lowercase letters are:");
    displayArray(chars);

    // Count the occurrences of each letter
    int[] counts = countLetters(chars);

    // Display counts
    System.out.println();
    System.out.println("The occurrences of each letter are:");
    displayCounts(counts);
  }

 /** Display the array of characters */
  public static void displayArray(char[] chars) {
    // Display the characters in the array 20 on each line
    for (int i = 0; i < chars.length; i++) {
      if ((i + 1) % 20 == 0)
        System.out.println(chars[i]);
      else
        System.out.print(chars[i] + " ");
    }
  }
```

```
  /** Count the occurrences of each letter */
  public static int[] countLetters(char[] chars) {
    // Declare and create an array of 26 int
    int[] counts = new int[26];

    // For each lowercase letter in the array, count it
    for (int i = 0; i < chars.length; i++)
      counts[chars[i] - 'a']++;

    return counts;
  }

  /** Display counts */
  public static void displayCounts(int[] counts) {
    for (int i = 0; i < counts.length; i++) {
      if ((i + 1) % 10 == 0)
        System.out.println(counts[i] + " " + (char)(i + 'a'));
      else
        System.out.print(counts[i] + " " + (char)(i + 'a') + " ");
    }
  }
}
```