Tracing Methods

For each of the program, trace the memory and output.

1.

```
// The "UseSwap" class.
import java.awt.*;

public class UseSwap
{

    public static void main (String[] args)
    {
        int a = 10;
        int b = 20;

        System.out.println ("Before swap: a = " + a + " b = " + b);
        swap (a, b);
        System.out.println ("After swap: a = " + a + " b = " + b);
    } // main method


    public static void swap (int num1, int num2)
    {
        int temp;
        System.out.println ("Beginning of procedure:  num1 = " + num1 + " num2 = " +
num2);
        temp = num2;
        num2 = num1;
        num1 = temp;
        System.out.println ("End of procedure:  num1 = " + num1 + " num2 = " + num2);
    }
} // UseSwap class
```

Output:

```
Before swap: a = 10 b = 20
Beginning of procedure: num1 = 10 num2 = 20
End of procedure: num1 = 20 num2 = 10
After swap: a = 10 b = 20
>
```

Memory:
main

a	10
b	20

swap

temp	20
num1	~~10~~ 20
num2	~~20~~ 10

2.

```
// The "UseSwapArray" class.
import java.awt.*;

public class UseSwapArray
{

    public static void main (String[] args)
    {
        int num[] = {3, 10};
        System.out.println ("Before Procedure:  num[0] = " + num [0] + " num[1] = " + num [1]);
        swapUseArray (num);
        System.out.println ("After Procedure:  num[0] = " + num [0] + " num[1] = " + num [1]);
    } // main method


    public static void swapUseArray (int[] a)
    {
        System.out.println ("Begin Procedure:  a[0] = " + a [0] + " a[1] = " + a [1]);
        int temp = a [0];
        a [0] = a [1];
        a [1] = temp;
        System.out.println ("End Procedure:  a[0] = " + a [0] + " a[1] = " + a [1]);

    }
} // UseSwapArray class
```

Output:

```
Before Procedure: num[0] = 3 num[1] = 10
Begin Procedure: a[0] = 3 a[1] = 10
End Procedure: a[0] = 10 a[1] = 3
After Procedure: num[0] = 10 num[1] = 3
>
```

Memory:
main

num[] (int)	~~3~~ 10	~~10~~ 3

swapUseArray

a[] (int)	
temp (int)	3

3.

```
// The "UseAddOne" class.
import java.awt.*;

public class UseAddOne
{

    public static void main (String[] args)
    {
        int num = 10;
        System.out.println ("Before Procedure:  num = " + num);
        addOne (num);
        System.out.println ("After Procedure:  num = " + num);
    } // main method


    public static void addOne (int a)
    {
        System.out.println ("Begin Procedure:  a = " + a);
        a++;
        System.out.println ("End Procedure : a = " + a);
    }
} // UseAddOne class
```

Output:

```
Before Procedure:  num = 10
Begin Procedure:  a = 10
End Procedure : a = 11
After Procedure:  num = 10
>
```

Memory:
main

num	10

addOne

a	~~10~~ 11

4.

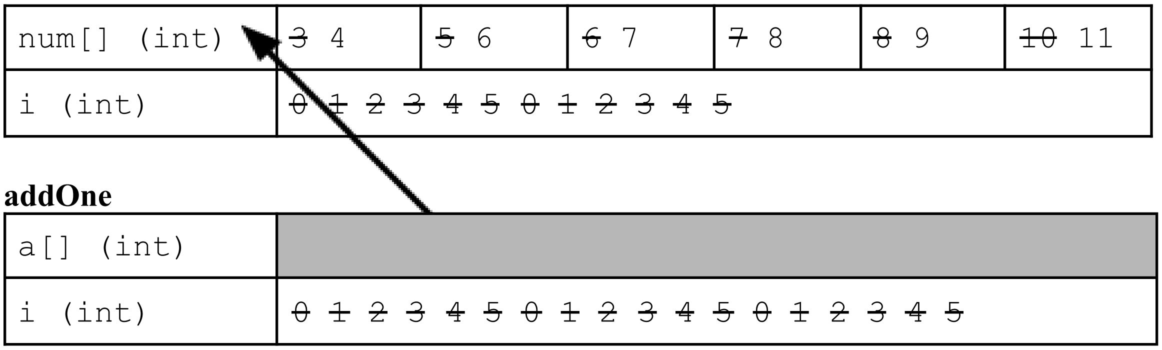
```
 // The "UseAddOneArray" class.
import java.awt.*;

public class UseAddOneArray
{
    public static void main (String[] args)
    {
        int num[] = {3, 5, 6, 7, 8, 10};
        System.out.println ("Before Procedure:  ");
        for (int i = 0 ; i < num.length ; i++)
        {
            System.out.println ("num[" + i + "] = " + num [i]);
        }
        addOneArray (num);
        System.out.println ("After Procedure:  ");
        for (int i = 0 ; i < num.length ; i++)
        {
            System.out.println ("num[" + i + "] = " + num [i]);
        }
    } // main method


    public static void addOneArray (int[] a)
    {
        System.out.println ("Begin Procedure:  ");
        for (int i = 0 ; i < a.length ; i++)
        {
            System.out.println ("a[" + i + "] = " + a [i]);
        }
        for (int i = 0 ; i < a.length ; i++)
        {
            a [i]++;
        }
        System.out.println ("End Procedure:  ");
        for (int i = 0 ; i < a.length ; i++)
        {
            System.out.println ("a[" + i + "] = " + a [i]);
        }
    }
} // UseAddOneArray class
```

Output:

```
Before Procedure:
num[0] = 3
num[1] = 5
num[2] = 6
num[3] = 7
num[4] = 8
num[5] = 10
Begin Procedure:
a[0] = 3
a[1] = 5
a[2] = 6
a[3] = 7
a[4] = 8
a[5] = 10
End Procedure:
a[0] = 4
```

```
a[1] = 6
a[2] = 7
a[3] = 8
a[4] = 9
a[5] = 11
Before Procedure:
num[0] = 4
num[1] = 6
num[2] = 7
num[3] = 8
num[4] = 9
num[5] = 11
>
```

Memory:
main



num[] (int)	~~3~~ 4	~~5~~ 6	~~6~~ 7	~~7~~ 8	~~8~~ 9	~~10~~ 11
i (int)	~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~					

addOne

a[] (int)	▨
i (int)	~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~