Solution to Practice Problems: Loops

Rewrite the following counter-control program to do while loop and for loop, in order to print out the biggest number inputted via keyboard.

```java
Scanner keyboard = new Scanner(System.in);
int max = -1;
int n;
int c = 0;
while (c<10)
{
    n = keyboard.nextInt();
    if (n > max)
    {
        max = n;
    }
    c++;
}
System.out.print(max);
```

Do while loop:

```java
Scanner keyboard = new Scanner(System.in);
int max, n, c;
max = -1;
c = 0;
do
{
    n = keyboard.nextInt();
    if (n > max)
    {
        max = n;
    }
    c++;
} while (c<10);

System.out.print(max);
```
For loop:

```java
Scanner keyboard = new Scanner(System.in);
int max, n, c;
max = -1;
For (c = 0; c < 10; c++)
{
    n = keyboard.nextInt();
    if (n > max)
    {
        max = n;
    }
}
System.out.print(max);
```

Rewrite the following event control program to while loop and for loop.

```java
Scanner keyboard = new Scanner(System.in);
int total = 0;
int c = 1;
int n = keyboard.nextInt();
while (c < 2*n)
{
    System.out.print(" "+c);
    total = total + c;
    c = c * 2;
}
System.out.println(total);
```

While loop:

```java
Scanner keyboard = new Scanner(System.in);
int total, c, n;
total = 0;
c = 1;
n = keyboard.nextInt();
do
{
    System.out.print(" "+c);
    total = total + c;
    c = c * 2;
} while (c < 2*n);
System.out.print(total);
```
For loop:

```java
Scanner keyboard = new Scanner(System.in);
int total, c, n;
total = 0;
n = keyboard.nextInt();
for (c = 1; c < 2 * n; c = c * 2)
{
    System.out.print(" "+c);
    total = total + c;
}
System.out.print(total);
```

Provide the value printed on the screenshot:

```java
public class While1 {
    public static void main(String [] args) {
        int x = 1;
        while(x < 10) {
            System.out.println("x = " + x);
            x++;
        }
    }
}
```

```
Screen:
x = 1
x = 2
x = 3
x = 4
x = 5
...
x = 8
x = 9
```
public class While2 {
    public static void main(String[] args) {
        int x = 1;
        while(x < 100) {
            System.out.println("x = " + x);
            if(x % 2 == 0)
                x++;
            else
                x *= 2;
        }
    }
}
x: starts at 1, then 2, then 3, then 6, then 7, then 14, then 15, then 30, then 31, then 62, then 63, then 126

screen
x = 1
x = 2
x = 3
x = 6
x = 7
...
x = 62
x = 63

public class For1 {
    public static void main(String[] args) {
        for(int i = 1; i < 10; i*=i) {
            System.out.println("i = " + i);
        }
    }
}

screen
x = 1
...
(infinite)

public class For2 {
    public static void main(String[] args) {
        for(int i = 1; i < 5; i++)
            System.out.println("i = " + i);
    }
}

screen
x = 1
x = 2
x = 3
x = 4
public class For3 {
    public static void main(String [] args) {
        for(int i = 1; i < 5; i++);
            System.out.println("i = " + i);
    }
}

Program has error and cannot be run.

public class For4 {
    public static void main(String [] args) {
        for(int i = 1; i < 0; i++) {
            System.out.println("i = " + i);
        }
    }
}

screen
(Nothing)

public class Do_while1 {
    public static void main(String [] args) {
        int i = 1;
        do{
            System.out.println("i = " + i);
            i++;
        }while (i < 0);
    }
}

screen
x = 1

1. **Understanding code**

Draw a representation of what the computer's memory looks like at the end of each of these programs:

public class Simple-While {
    public static void main(String [] args) {
        int x = 1;
        while(x < 10) {
            System.out.println("x = " + x);
            x++;
        }
    }
}

x: starts at 1, changes to 2, then 3, then 4, then 5, ..., then 9, then 10
public class Simple-While2 {
    public static void main(String [] args) {
        int number = 1;
        while (number <= 200) {
            System.out.print(number + " ");
            number *= 2;
        }
    }
}

number: starts at 1, changes to 2, then 4, then 8, then 16, ..., then 128, then 256

screen
1 2 4 8 16 32 64 128

public class Infinite-While {
    public static void main(String [] args) {
        boolean b = true;
        while(b) {
            System.out.println("are we there yet?");
        }
    }
}

b: true

screen
are we there yet?
are we there yet?
are we there yet?
are we there yet?
are we there yet?
are we there yet?
... (ad infinitum)
public class Complex-Update {
    public static void main(String[] args) {
        int x = 2;
        while (x < 1000) {
            System.out.println("x = "+ x);
            x = x * x;
        }
    }
}

x: starts at 2, then 4, then 16, then 256, then something really big, much bigger than 1000 (specifically, 256 * 256 = 65536)

screen
x = 2
x = 4
x = 16
x = 256
(Note: this program does NOT display x = 65536 on the screen.)

public class Infinite-For {
    public static void main(String[] args) {
        for( ; true; ) {
            System.out.println("are we there yet?");
        }
    }
}

screen
are we there yet?
... (ad infinitum)

public class Complex-Update-For {
    public static void main(String[] args) {
        for(int i = 2; i < 1000; i = i * i) {
            System.out.println("i = "+ i);
        }
    }
}

i: starts at 2, then 4, then 16, then 256, then 65536
screen: displays "i = " plus each of the above values EXCEPT 65536
x = 2
x = 4
x = 16
x = 256

memory: i is invalid after the loop.
public class Factor {
    public static void main(String[] args) {
        int number = 13;
        int factor = 2;
        while (number % factor != 0) {
            factor++;
        }
        System.out.println("First factor: "+ factor);
    }
}

number: 2
factor: starts at 2, then 3, then 4, ..., then 12, then 13
screen: First factor: 13

public class Sum {
    public static void main(String[] args) {
        int sum = 0, n = 12345;
        while (n > 0) {
            sum += n % 10;  // add last digit to sum
            n = n / 10;     // remove last digit
        }
        System.out.println("sum = " + sum);
    }
}

n: starts at 12345, then 1234, then 123, then 12, then 1, then 0
sum: starts at 5, then 9, then 12, then 14, then 15
screen: sum = 15