

## Questions and Exercises

*These questions and exercises is an opportunity to see what you've learnt from the lecture as well as practice the new things we've been talking about. In other words, these questions and exercises are completely optional but it's recommended to do them. In the end of the document you will find the answers to the questions as well as possible solutions to the exercises, note that one can solve an exercise in different ways. There will also be some suggestions about what one could code if one want to continue with some more advanced things. These suggestions will not come with a possible solution and might include things that haven't been covered in the lecture.*

### Question 1

What are the following numbers in base 10? Convert them manually.

$210_8$   
 $DEAD_{16}$   
 $1000111110_2$

### Question 2

What are the following numbers in base 16? Convert them manually.

$123_{10}$   
 $42_{10}$   
 $18_{10}$

### Question 3

What are the following numbers in base 2? Convert them manually.

$256_{10}$   
 $123_{10}$   
 $603_{10}$

### Question 4

How many bits can a byte, a short an int and a long store?

### Question 5

What will the following code output?

```
System.out.println((byte)0b10000001)
```

### Exercise

There are built in methods to convert between bases in Java. Write your own one that converts from binary to decimal and from decimal to binary.

**Further explorations**

Continue with the exercise to allow any base to be converted to decimal and decimal to be converted to any base.

## Answers and solutions

**Answer to Question 1**

$210_8 = 136_{10}$   
 $DEAD_{16} = 57005_{10}$   
 $1000111110_2 = 574_{10}$

**Answer to Question 2**

$123_{10} = 7B_{16}$   
 $42_{10} = 2A_{16}$   
 $18_{10} = 12_{16}$

**Answer to Question 3**

$256_{10} = 100000000_2$   
 $123_{10} = 1111011_2$   
 $603_{10} = 1001011011_2$

**Answer to Question 4**

Byte: 8 bits  
Short: 16 bits (2 bytes)  
Int: 32 bits (4 bytes)  
Long: 64 bits (8 bytes)

**Answer to Question 5**

It will output -127. The reason is because the first bit is worth -128 instead of 128. This is because all Java integers are signed.

**Possible solution to Exercise**

```
public class BaseConversion {  
  
    public static void main(String[] args) {  
        System.out.println(toBase10("11001"));  
        System.out.println(toBase2(25));  
    }  
  
    public static int toBase10(String str) {  
        int result = 0;  
  
        for (int i = 0; i < str.length(); i++) {  
            int bitValue = str.charAt(str.length() - 1 - i) != '0' ? 1 : 0;  
            result += bitValue * Math.pow(2, i);  
        }  
    }  
}
```

```
    }  
  
    return result;  
}  
  
public static String toBase2(int num) {  
    String result = "";  
  
    while (num > 0) {  
        char bitValue = num % 2 != 0 ? '1' : '0';  
        result = bitValue + result;  
        num /= 2;  
    }  
  
    return result;  
}  
}
```