Work on project. Stage 1/6: For example

1752 users solved this problem. Latest completion was **about 15 hours ago**.

Project: Numeral System Converter

Medium 3 minutes

Description

In this project, you will be creating your own numeral system converter. If you don't know what a numeral system is, or simply need to brush up on the topic, browse through the relevant Wikipedia page.

Implement a program that outputs a number in two systems: the first is decimal, and the second one is binary. This must be **the same** number represented in different systems; feel free to choose any number you like.

This stage is auto-graded. The grader will check that:

- you output a single line;
- there are two numbers;
- the first number is decimal;
- the second number is binary (it starts with `0b` and contains only `0` and `1`);
- the first number is equal to the second one.

Note that you do not need to enter anything into the program, you should print a line prepared in advance. A single String is expected to be printed.

7 / 7 Prerequisites

- ✓ Introduction to Java 6 — Stage 1
- ✓ Basic literals 6 — Stage 1
- ✓ Overview of the basic program 6 — Stage 1
- ✓ Printing data 6 — Stage 1
- ✓ Binary numbers 6

Show all

Examples

Example 1:

```
10 = 0b1010
```

Example 2:

```
2 is equal to 0b10
```

Report a typo

See hint

Write a program

Code Editor IDE

Java

```
package converter;

public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}

```

Run Continue Solutions (56)

Comments (13) Hints (4) Useful links (0) Solutions (56) Show discussion

Work on project. Stage 2/6: Almost an octopus

1544 users solved this problem. Latest completion was **about 15 hours ago**.

Project: Numeral System Converter

Medium 6 minutes

Description

Let's try to convert something! Implement a program that calculates **the last digit** of the given number converted to base 8. You don't need to convert the whole number, just print the last digit of the result.

This stage is auto-graded. The grader will **input a number** in base 10, and then check that your program **output** matches the correct answer.

9 / 9 Prerequisites

- Types and variables 5 ★ — Stage 1
- Comments 5 ★ — Stage 1
- Coding style conventions 5 ★ — Stage 1
- Naming variables 5 ★ — Stage 1
- Scanning the input 5 ★ — Stage 1

Show all

Example

Example 1

Input:

```
8
```

Output:

```
0
```

Example 2

Input:

```
9
```

Output:

```
1
```

Report a typo

See hint

Write a program

Code Editor IDE

Java

```
package converter;

public class Main {
    public static void main(String[] args) {
        System.out.println("3 = 0b11");
    }
}

```

Continue | This problem has been changed. Reload? | Solutions (73)

Comments (30) Hints (6) Useful links (1) Solutions (73) Show discussion

Work on project. Stage 3/6: Convert decimals

1430 users solved this problem. Latest completion was **about 14 hours ago**.

Project: Numeral System Converter

Medium 9 minutes

Description

Now let's implement a simple converter. It will convert the given decimal number to the given radix. You should support three radices with prefixes:

- binary (`0b`);
- octal (`0`);
- hexadecimal (`0x`).

To get a string with the answer, use the `Long.toString(sourceNumber, destinationRadix)` expression. Note that the expected output is a String, because Java implicitly converts 0/0b/0x concatenated numbers to their decimal representation.

13 / 13 Prerequisites

- Boolean and logical operations — Stage 1 — 4
- Relational operators — Stage 1 — 4
- Conditional statement — Stage 1 — 4
- Ternary operator — Stage 1 — 4
- The for-loop — Stage 1 — 4

Show all

This stage is auto-graded. The grader will **input two lines** (a number and a radix) and check that **your output** is the correct number representation in the given radix. Don't forget about the prefix!

Example

Example 1:

Input:

```
8
16
```

Output:

```
0x8
```

Example 2:

Input:

```
101
2
```

Output:

```
0b1100101
```

Example 3:

Input:

```
103
8
```

Output:

```
0147
```

Report a typo

See hint

Write a program

Code Editor IDE

Java

```
package converter; // use?
import java.util.Scanner;

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

Work on project. Stage 4/6: Any which radix

1227 users solved this problem. Latest completion was **about 14 hours ago**.

Project: Numeral System Converter

Medium 23 minutes

Description

In this stage, you should add support for reading the source number in the given base and converting it to another given base.

8 / 8 Prerequisites

- ✓ Units of information 3★ — Stage 1
- ✓ Sizes and ranges 3★ — Stage 1
- ✓ Type casting 3★ — Stage 1
- ✓ Calling a method 3★ — Stage 1
- ✓ Primitive and reference types 3★ — Stage 1

Show all

Algorithm

As there are 26 Latin letters and 10 digits, the maximum base is 26 + 10 = 36.

To convert a number from the source base to the target base, take the following steps:

- If the given number is not decimal, you can convert it to a decimal using the method `Integer.parseInt(number, sourceBase)` that returns a decimal representation of a number.
- After that, you can use the method `Integer.toString(decimalNumber, newBase)` that takes a decimal number and converts it to the target base.

Note that the minimum radix is 1: the number `N` in radix 1 contains the symbol `1` N times. The methods described above don't work if `sourceBase` or `newBase` equal to 1. In that case, you should convert the numbers manually.

For instance:

$111111_1 = 6_{10}$
$3_{10} = 111_1$

This stage is auto-graded. The grader will input three lines:

1. The source radix;
2. The source number;
3. The target radix.

Then, it will check that your output contains the correct number representation in the given radix. This time, don't use prefixes now.

Example

Example 1:

Input:

```
10
11
2
```

Output:

```
1011
```

Example 2:

Input:

```
1
11111
10
```

Output:

```
5
```

Example 3:

Input:

```
10
4
1
```

Output:

```
1111
```

Example 4:

Input:

```
10
1001
36
```

Output:

```
rt
```

Report a typo

See hint

Write a program

Code Editor IDE

Java

```java
package converter; // use?
import java.util.Scanner;

public class Main
{
  public static void main (String[] args)
  {
    Scanner sc = new Scanner (System.in);
    int a = sc.nextInt ();
    int base = sc.nextInt();
    String b = "";

    //By convention, java Objects have capitalized first-letter names (e.g. String), while primitives have lower case names (

    //These three functions return a string
    if (base == 2)
    {
      b = Integer.toBinaryString (a);
      b = "0b" + b;
    }
    else if (base == 8)
    {
      b = Integer.toOctalString (a);
      b = "0" + b;
    }
    else if (base == 16)
    {
      b = Integer.toHexString (a);
      b = "0x"+b;
    }
    else
    {
      System.out.print("Enter valid base");
    }
    System.out.print(b);
  }
}

```

Continue This problem has been changed. Reload? Solutions (111)

Comments (30) Hints (7) Useful links (1) Solutions (111) Show discussion

Work on project. Stage 5/6: Converting fractions

847 users solved this problem. Latest completion was **about 12 hours ago**.

Project: Numeral System Converter

Medium 37 minutes

Description

Fractional numbers can also be converted from one base to another. To convert a fractional number to another base, you should use the algorithm described below.

10 / 10 Prerequisites

- Introduction to OOP 2★ — Stage 1
- Floating-point types 2★
- Write, compile, and run 2★ — Stage 1
- Declaring a method 2★ — Stage 1
- The main method 2★ — Stage 1

Show all

Algorithm

As you know from the previous stage, in order to convert a number from one base to another, first, we need to convert it to decimal if it's not decimal yet, and only then convert it to another base. The same applies to fractional numbers.

Let's imagine you have a fractional number $ab.cdef$ where ab is the integer part, $cdef$ is the fractional part, and a, b, c, d, e, f are some digits or letters, depending on the base of the number. In this case, we have 2 digits (letters) in the integer part and 4 digits (letters) in the fractional part. In other cases, the number of digits (letters) in parts can be different.

To convert the number into decimal, we need to:

- Split the number into two parts: integer and fractional;
- Convert the integer part into decimal using the method from the previous stage;
- Convert the fractional part into decimal using the following formula:

$$decimalValue = \frac{c}{base^1} + \frac{d}{base^2} + \frac{e}{base^3} + \frac{f}{base^4}$$

where $base$ is a base of the number you want to convert into decimal.

The more digits (letters) in the fractional part, the more addends in the formula. If the fractional part has letters, then you should use their number representation: 'a' – 10, 'b' – 11, c – '12', and so on.

As a result, you should get a decimal number no greater than 1. If it is greater than 1, you did something wrong.

To see the decimal representation of the source number, you can sum the decimal integer and the decimal fractional parts.

To convert a decimal fractional number into any other base, we need to:

- Split the decimal number into two parts: integer and fractional;
- Convert the integer part into the new base;
- Convert the fractional part from decimal to any other base.

Let's elaborate on the third step and convert the fractional part of 0.5168 into base 19 as an example.

Multiply the fractional part by the new base: $0.5168 * 19 = 9.8192$. The integer part of the result is the first digit (or letter if the integer part is greater than 9) in the fractional part of a number in the new base. In this case, the first digit in the fractional part is **9**.

Take the fractional part from the result of the multiplication and multiply it by the new base again: $0.8192 * 19 = 15.5648$. For numbers greater than 9, you should use their letter representation: 10 – 'a', 11 – 'b', 12 – 'c', and so on. In this case, the second digit (letter) is **f (15)**.

To calculate the rest of the digits (letters), you should repeat this action. In this stage, your program should output only 5 digits (letters) in the fractional part:

$0.5648 * 19 = 10.7312\ (\mathrm{a})$
$0.7312 * 19 = 13.8928\ (\mathrm{d})$
$0.8928 * 19 = 16.9632\ (\mathrm{g})$

The final result looks like this: $0.5168_{10} = 0.9fadg_{19}$

Example 1:

0.234_{10} to base 7.

1. $0.234 \cdot 7 = 1.638$: the first fraction symbol is 1.
2. Remove integer part. $0.638 \cdot 7 = 4.466$: the second fraction symbol is 4.
3. Remove integer part. $0.466 \cdot 7 = 3.262$: the third fraction symbol is 3.
4. The same for the next symbols.

So, the number in base 7 is $0.143..._7$

Example 2:

0.234_{10} to base 36.

1. $0.234 \cdot 36 = 8.424$: the first fraction symbol is 8.
2. $0.424 \cdot 36 = 15.264$: the second fraction symbol is '15' and 'f' in base 36.
3. $0.264 \cdot 36 = 9.504$: the third fraction symbol is 9.

So, the number in base 36 is $0.8f9..._{36}$

Example 3:

0.234_7 to base 36.

1. Convert the fractional part to base 10 using the formula above: $2/7 + 3/49 + 4/343 = 0.358..._{10}$
2. Convert this number to base 36 like in example 2.

This stage is auto-graded. The grader will input three lines: the source radix, the source number, and the target radix. Then it will check that your output is the correct number representation in the given radix. Don't forget to round up the result to 5 decimal places! If there is no fractional part in the initial number, you don't need to show the fractional part.

Also, numbers in radix 1 cannot have a fractional part so they're tested only as the integer part, like it was in the previous stage.

Examples

Example 1:

Input:

```
10
0.234
7
```

Output:

```
0.14315
```

Example 2:

Input:

```
10
10.234
7
```

Output:

```
13.14315
```

Example 3:

Input:

```
35
af.xy
17
```

Output:

```
148.g88a8
```

Example 4:

Input:

```
16
aaaaa.0
24
```

Output:

```
22df2.00000
```

Example 5:

Work on project. Stage 6/6: Error!

Project: Numeral System Converter

Hard 17 minutes

921 users solved this problem. Latest completion was **about 2 hours ago**.

Description

Let's face it: errors happen. What if someone enters the wrong radix? What if the input number is impossible to convert to the given radix? What if the number is not a number at all?

In this final stage, we will make sure the program can handle errors like that. At this point, you can implement this without the `try catch` construction. Use the following rule of thumb: if you can avoid exception-based logic, avoid it!

So, the goal here is to implement error messages when the input is wrong.

This stage is auto-graded. The grader will input some data. Then it will check that the last line of your program output contains the word "error" if the input is wrong.

If the input is correct, your program should behave as in the previous stage.

11 / 11 Prerequisites

- Errors in programs 1 Stage 2
- What is an exception 1 Stage 2
- Arrays as parameters 1 Stage 1
- Constructor 1 Stage 1
- Instance methods 1 Stage 1

Show all

Report a typo

HINT by Avinal Kumar Viewed hints

Check for negative radix and grater than 36 too.

Was this hint helpful? Yes No Report

Write a program

Code Editor IDE

Java

```
package converter; // use?
/*
Character.getNumericValue('a'); //10
For capital a also value is 10
Character.getNumericValue('12'); //12
and vice versa 10=a, 11=b ...
Character.forDigit(Value, radix)
-----------------------------------
We can have number like: 3e.4fc/3E.4FC
*/

import java.util.Scanner;
import java.lang.Math;

public class Main
{
    public static void main (String[] args)
    {
        boolean hasErrors = false;
        Scanner scanner = new Scanner(System.in);
        int source_radix = 0;
        String source_number = "";
        int target_radix = 0;

        if (!scanner.hasNextInt()) {
            hasErrors = true;
        } else {
            source_radix = scanner.nextInt();
        }

        if (!scanner.hasNext()) {
            hasErrors = true;
        } else {
            source_number = scanner.next();
        }

        if (!scanner.hasNextInt()) {
            hasErrors = true;
        } else {
            target_radix = scanner.nextInt();
        }

        if (source_radix < 1 || source_radix > 36 || target_radix < 1 || target_radix > 36) {
```