Convert long value to KB,MB,GB,TB

```
//Microsoft Reciprocal License (Ms-RL)
//http://bmcommons.codeplex.com/license
using System;
using System.Globalization;
using System.Text;
using System.Text.RegularExpressions;

namespace BlueMirror.Commons
{

public static class IntExtensions
{
public static string ToRoman(this int number, bool upperCase)
{
return CustomConvert.ToRoman(number, upperCase);
}
}

public static class CustomConvert
{
public static string ToRoman(int number, bool upperCase)
{
if (number < 0) throw new ArgumentOutOfRangeException("number", number, "Liczba musi byæ wiêksza od zera."); string[] romans = new string[] {"I", "IV", "V", "IX", "X", "XL", "L", "XC", "C", "CD", "D", "CM", "M"}; // string[] romansLower = new string[] {"i", "iv", "v", "ix", "x", "xl", "l", "xc", "c", "cd", "d", "cm", "m"}; int[] numbers = new int[] {1, 4, 5, 9, 10, 40, 50, 90, 100, 400, 500, 900, 1000}; int j = 12; string result = ""; // string[] romans = upperCase? romansUpper: romansLower; // za romansUpper i romansLower powstawiac konstruktory tablic - new
```

```
string[] {} while(number != 0) { if(number >= numbers[j])
{
number -= numbers[j];
result += romans[j];
}
else
j–;
}
if (!upperCase)
result = result.ToLower();
return result;
}

public static string ToSay(double number)
{
double floor = Math.Floor(number);
string result = ToSay(System.Convert.ToInt64(floor)) + ” i ” +
ToSay(System.Convert.ToInt64(Math.Round((number – floor) *
100))) + ” setnych”;
return result;
}

public static string ToSay(decimal number, IFormatProvider
format)
{
string result;
NumberFormatInfo nfi;
if(format != null)
nfi =
(NumberFormatInfo)format.GetFormat(typeof(NumberFormatInfo));
else
nfi = CultureInfo.CurrentCulture.NumberFormat;
//if (nfi == null)
//  throw new Exception(“Nie mo¿na uzyskaæ obiektu
NumberFormatInfo.”);
// TODO: double i decimal – wyprostowaæ.
double floor = Math.Floor((double)number);
```

```
//long floor = System.Convert.ToInt64(Math.Floor(number));
result = ToSay(System.Convert.ToInt64(floor)) + ” ” +
nfi.CurrencySymbol + ” ” +
ToSay(System.Convert.ToInt64(Math.Round(((double)number –
floor) * 100)));
return result;
}

public static string ToKB(long bytes) {
string[] suffix = new string[] { “B”, “KB”, “MB”, “GB”, “TB”
};
float byteNumber = bytes;
for (int i = 0; i < suffix.Length; i++) { if (byteNumber <
1000) if(i == 0) return string.Format("{0} {1}", byteNumber,
suffix[i]); else return string.Format("{0:0.#0} {1}",
byteNumber, suffix[i]); else byteNumber /= 1024; } return
string.Format("{0:N} {1}", byteNumber, suffix[suffix.Length -
1]); } public static string ToRegex(string wildcard) { string
result = "^"; foreach (char chin in wildcard) { if (chin ==
'*') result += ".*"; else if (chin == '?') result += "."; else
if (chin == ';') result += "$|^"; else if
("+()^$.{}[]|".IndexOf(chin) != -1) result += "" + chin; else
result += chin; } return result + "$"; } } } [/csharp]
```