

追加演習

教材の練習問題の追加演習

[前の演習](#)よりも高難易度に設定している。できるところから実施すること。

問題10 素数判定

整数型`n`を引数として受け取り、その数が素数であれば`true`、そうでなければ`false`を戻り値として返す関数`isPrime`を疑似言語で作成せよ。

[プログラム]

```
○論理型:isPrime(整数型:n)  
  //続きを以下に記載すること
```

問題20 フィボナッチ数列

整数型`n`を引数として受け取り、`n`番目のフィボナッチ数を返す関数`fibonacci`を再帰的に疑似言語で作成せよ。

フィボナッチ数とは、次のように定義される数列です。

- 1番目と2番目の値は1、それ以降の値は直前の2つの値の和として求められます。
- 具体的には、数列は次のようになります: 1, 1, 2, 3, 5, 8, 13, ...

上記より、`fibonacci(6)`は8となります。

[プログラム]

```
○整数型: fibonacci(整数型: n)  
  // 続きを以下に記載すること
```

問題30 階乗計算

整数型 n を引数として受け取り、 n の階乗を返す関数`factorial`を疑似言語で作成せよ。

[プログラム]

```
○整数型: factorial(整数型: n)  
  //続きを以下に記載すること
```

問題40 完全数の判定

整数型 n を引数として受け取り、その数が完全数であれば`true`、そうでなければ`false`を返す関数 `isPerfectNumber`を疑似言語で作成せよ。

完全数とは、自分自身を除く正の約数の和がその数自身と等しくなる数のことです。例えば、 6 は 1 、 2 、 3 が約数で、 $1 + 2 + 3 = 6$ となるため完全数です。

[プログラム]

○論理型: `isPerfectNumber(整数型: n)`

// 続きを以下に記載すること

