

Rプログラミング (演習)

○関数定義

問題 1 :

(1) あるライブコンサートのチケットの料金は次の通りである。

S席：10000円， A席：8000円， B席：5000円

このとき，各チケットの枚数を変数とし，出力を売上額とする関数 `rieki` を定義しなさい。

出力例：

`rieki(2,3,4) = 64000`

(2) (ヘロンの公式) 3つの正の数値(a, b, c)を入力して，その長さを三辺とする三角形の面積を求める関数 `heron01`を作成しなさい。

出力例：

`heron01(2,2,2)=1.732051`

○if 文

問題2

(1) AとBが買い物に行くとする。Aがある書籍を購入したい。

もしもAの所持金で足りればそのまま支払う。

もしもAの所持金で足りない場合、Bから借りて支払う。

もしもAの所持金で足りない場合、Bから借りても支払うことができなければ購入しない。

A, Bの所持金と書籍の値段を入力したとき、Bからの借入金額を出力する関数 `loan` を定義しなさい。

出力例：

$$\text{loan}(200,300,150) = 0$$

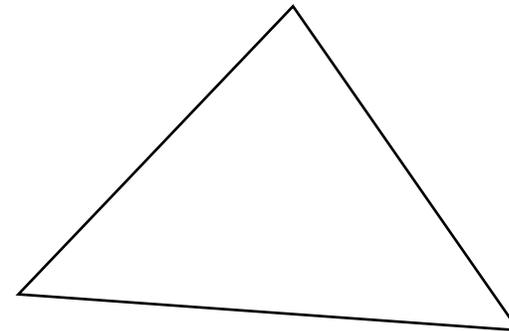
$$\text{loan}(200,300,400) = 200$$

$$\text{loan}(200,300,900) = \text{"too expensive"}$$

○ if文（ヘロンの公式）

(2) 3つの正の数値(a, b, c)を小さい順に入力して, その長さを三辺とする三角形の面積を求める関数を heron02 作成しなさい。ただし, 3つの数値から三角形が出来ない場合は999を出力する。

(3) 3つの正の数値(a, b, c)をランダムに入力して, その長さを三辺とする三角形の面積を求める関数 heron03 を作成しなさい。ただし, 3つの数値から三角形が出来ない場合は999を出力する。



○For 文

問題3 次のような初項から第n項までの和を求める関数 `sowa` を定義しなさい.

(1) $\text{sowa1} = 1 + 2 + 3 + 4 + 5 + \dots$

(2) $\text{sowa2} = 1 + 2 + 4 + 8 + 16 + \dots$

(3) $\text{sowa3} = 1 + 4 + 9 + 16 + \dots$

出力例 : $\text{sowa1}(100) = 5050,$
 $\text{sowa2}(4) = 15,$
 $\text{sowa3}(3) = 14.$

問題4 次のような初項から第n項までの和を求める関数 wa を再帰的に定義しなさい.

ただし, if文と再帰的定義を用いること (実習8を参照:問題3と関数の機能は同じ)

$$(1) wa1 = 1 + 2 + 3 + 4 + 5 + \dots$$

$$(2) wa2 = 1 + 2 + 4 + 8 + 16 + \dots$$

$$(3) wa3 = 1 + 4 + 9 + 16 + \dots$$

出力例 : $wa1(100) = 5050,$
 $wa2(4) = 15,$
 $wa3(3) = 14.$

問題5 フィボナッチ数列：1, 1, 2, 3, 5, 8...を生成する関数 `fib` を再帰的に定義しなさい。

出力例：`fib(6) = 8`

○カウント関数

問題6 n 個の自然数のリストが与えられたとき、その中に偶数がいくつあるかカウントする関数 `count_even` を定義しなさい。

出力例：

```
data <-c (1,2,3,4,5)
count_even(data)=2
```

○線形探索

問題7 すべての値が異なる自然数からなるリストに対し、自然数 n を入力したとき、その数値がリスト（ベクトル）に入っているかを判定する関数 `linear_search` を定義しなさい。
入っていればその位置を、入っていない場合は -1 を出力する。

出力例

```
data<-{1,2,3,4,5,6}  
linear_search(data, 3)=3,  
linear_search(data, 9)=-1,
```

○二分探索

問題8：値が小さい順に並んでいる自然数からなるリストに対し、このリストから特定の値 n を探索したい。「二分探索」のプログラムとはどのような探索アルゴリズムかを調べなさい。

○ピタゴラス数

問題9 自然数 n を入力したとき、 n までのピタゴラス数（3つの対）を出力する関数 `pythagoras` を定義しなさい。（定数倍の重複を許す）

出力例：

`pythagoras (20)`

[1] 1 0 1

[1] 3 4 5

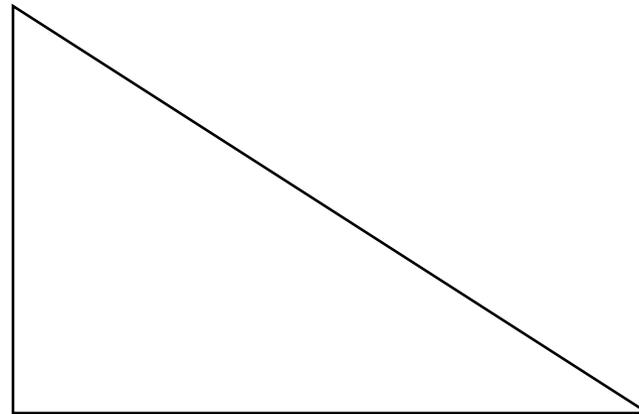
[1] 6 8 10

[1] 5 12 13

[1] 9 12 15

[1] 8 15 17

[1] 12 16 20



問題10 Collatz関数($3n+1$ 問題)

(1) 関数collatz (自然数 n が奇数 $\rightarrow 3n+1$, n が偶数 $\rightarrow n/2$) を定義しなさい.

出力例 : $\text{collatz}(9) = 28$, $\text{collatz}(10) = 5$.

(2) 関数collatzを s 回繰り返す2変数関数 $\text{collatzlist}(n,s)$ を定義しなさい.

出力例 : $\text{collatzlist}(3, 10)=3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1$.

(3) 関数collatzlist関数において1が出力されたら処理を終了する関数collatzlist2を定義しなさい.

出力例 : $\text{collatzlist2}(3)=3, 10, 5, 16, 8, 4, 2, 1$

問題10 (2) (ヒント)

collatz関数をs回繰り返す関数 collatzlist を作成しなさい。

```
collatzlist <- function(n,s){
```

```
  for (  ){
```

```
    if(n %% 2 == 0 ){  
      n <- n/2
```

```
    }
```

```
    else{
```

```
      n <- 3*n+1
```

```
    }
```

```
    print(n)
```

```
  }
```

```
}
```

変数をiにして1からsまで繰り返す

問題10(3) ヒント

collatzlist関数で1が出力されたら処理を終了

```
collatzlist2 <- function(n){  
  x <- n  
  while (  ){  
    # nが1になるまで繰り返す  
    n <- collatz2(n)  
    x <- c(x, n)  
  }  
  print(x)  
}
```

問題11(発展) 1から100までで一番長い経路をだどって1になる自然数を求めよう。

- (1) 1から100までのcollatz数で1に至るまでのリスト数を定義するcollatzlist 3を作成しなさい。
- (2) 1から100までで1に至るまで一番長いcollatz数の経路をグラフ化しなさい。