リスト処理の例

胡　振江

例題1：数をことばに

- 問題：

  0以上100万以下の数 ➔ 通常の英語表現

  例：

  - 308000 ➔ three hundred and eight thousand
  - 369027 ➔ three hundred and sixty-nine thousand and twenty-seven
  - 369401 ➔ three hundred and sixty-nine thousand four hundred and one

解決法

- 簡単な問題から複雑問題へ
  - n<100 の数字を対象に
  - n<1000 の数字を対象に
  - n< 1000,000 の数字を対象に

数の英語名：文字列

```
units = [ "one", "two", "three", "four", "five",
          "six", "seven", "eight", "nine"]

teens = ["ten", "eleven", "twelve", "thirteen",
         "fourteen", "fifteen", "sixteen",
       "seventeen", "eighteen", "nineteen"]

tens = ["twenty", "thirty", "forty", "fifty", "sixty",
        "seventy", "eighty", "ninety"]
```

0<n<100の場合

```
convert2 n = combine2 (digits2 n)

digits2 n = (n `div` 10, n `mod` 10)

combine2 (0,u+1)      = units !! u
combine2 (1,u)        = teens !! u
combine2 (t+2,0)      = tens !! t
combine2 (t+2,u+1)    = tens !! t ++ "-" ++
                        units !! u
```

0<n<1000の場合

```
convert3 n = combine3 (digits3 n)

digits3 n = (n `div` 100, n `mod` 100)

combine3 (0,t+1) = convert2 (t+1)
combine3 (h+1,0) = units !! h ++ " hundred"
combine3 (h+1,t+1) = units !! h ++ " hundred
                     and " ++ convert2 (t+1)
```

0<n<1000,000の場合

```
convert6 n = combine6 (digits6 n)
digits6 n = (n `div` 1000, n `mod` 1000)

combine6 (0,h+1) = convert3 (h+1)
combine6 (m+1,0) = convert3 (m+1) ++ " thousand"
combine6 (m+1,h+1) = convert3 (m+1) ++
                                " thousand“ ++
                                link (h+1) ++
                                convert3 (h+1)

link h | h < 100 = " and "
       | otherwise = " "
```

実行例

Convert> convert6 308000
"three hundred and eight thousand"
(985 reductions, 1350 cells)

Convert> convert6 369027
"three hundred and sixty-nine thousand and twenty-seven"
(1837 reductions, 2547 cells)

Convert> convert6 369401
"three hundred and sixty-nine thousand four hundred and one"
(1851 reductions, 2548 cells)

例題2：カレンダーの印刷

- 問題：calendar 2005 ➔

JANUARY 2005 FEBRUARY 2005 MARCH 2005

Sun 2 9 16 23 30	Sun 6 13 20 27	Sun 6 13 20 27
Mon 3 10 17 24 31	Mon 7 14 21 28	Mon 7 14 21 28
Tue 4 11 18 25	Tue 1 8 15 22	Tue 1 8 15 22 29
Wed 5 12 19 26	Wed 2 9 16 23	Wed 2 9 16 23 30
Thu 6 13 20 27	Thu 3 10 17 24	Thu 3 10 17 24 31
Fri 7 14 21 28	Fri 4 11 18 25	Fri 4 11 18 25
Sat 1 8 15 22 29	Sat 5 12 19 26	Sat 5 12 19 26

APRIL 2005 MAY 2005 JUNE 2005

…

↓

抽象的なカレンダーの構成

↓

カレンダーの印刷

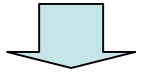
↓

図形の表示

```
type Picture = [[Char]]

height,width :: Picture -> Int
height p = length p
width p = length (head p)
```

```
1 2 3 4
5 6 7 8
```



```
[['1 ','2 ','3 ','4 '],
 ['5 ','6 ','7 ','8 ']]
```

図形の構成

図形qの上に図形pを置く

```
p `above` q | width p == width q = p++q
```

図形pを図形qの左に置く

```
p `beside` q | height p == height q = zipWith (++) p q
```

図形のリストを縦に積む

```
stack = foldr1 above
```

図形リストを横に並べる

```
spread = foldr1 beside
```

特定の高さと幅をもつ空の図形の生成

```
empty (h,w) = copy (copy ' ' w) h
```

図形のgrouping

```
block :: Int -> [Picture] -> Picture
block n = stack . map spread . group n
group n xs = [take n (drop j xs) | j <- [0,n..(length xs-n)]]
```

[G1,G2,G3,G4,G5,G6,G7,G8] ➔ n=2

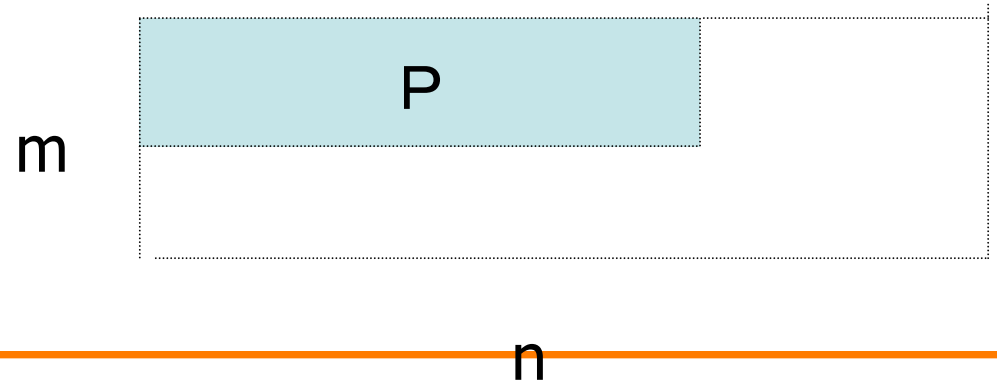
G1 G2
G3 G4
G5 G6
G7 8G

```
blockT :: Int -> [Picture] -> Picture
blockT n = spread . map stack . group n
```

図形の埋め込み

高さm,幅nの大きな図形の左上部に図形pをはめ込む

```
lframe (m,n) p = (p `beside` empty (h,n-w))
                      `above` empty (m-h,n)
    where h = height p
          w = width p
```



カレンダーの表示

```
Month_pic (mn,yr,fd,ml) = title mn yr `above` table fd ml

各月の見出し
title mn yr = lframe (2,25) [mn ++ " " ++ show yr]

table fd ml = lframe (8,25) (daynames `beside` entries fd ml)

daynames = ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"]

entries fd ml = blockT 7 (dates fd ml)
dates fd ml = map (date ml) [(1-fd)..(42-fd)]
date ml d | d<1 || ml < d = [rjustify 3 " "]
          | otherwise     = [rjustify 3 (show d)]
```

カレンダーの作成

```
calendar :: Int -> String
calendar = display . block 3 . map month_pic . months

months yr = zip4 mnames (copy yr 12) (fstdays yr)
                        (mlengths yr)
    where zip4 [] [] [] [] = []
          zip4 (x:xs) (y:ys) (z:zs) (u:us)
               = (x,y,z,u) : zip4 xs ys zs us

display = unline
```

カレンダーの印刷

> putStrLn (calender 2004)

中間レポートの提出について

- 演習問題

  講義で説明したカレンダーのプログラムを次のように変更する。
  - カレンダーの曜日が日曜日からではなくて月曜日から始まる。
  - 縦に並んでいる曜日の名前を横にする。
- 報告書の内容:
  - ソース・実行例・変更点など
  - 講義の出席表
- 締切日:12月27日(月)
- 提出先:胡のポストへ

報告書に名前と学生証番号を忘れずに記入すること