

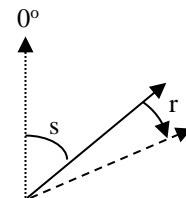
## Sem. 2 2014/2015

### Soal 1. Ekspresi Dasar dan Rekursif

- a. Fungsi **ubahArah**, menerima masukan berupa dua buah integer. Integer pertama (**s**) bernilai antara 0 hingga 359, merepresentasikan arah pergerakan sebuah benda (dalam satuan derajat). Integer kedua (**r**) bernilai antara -359 hingga 359 menyatakan besar perubahan arah pergerakan benda tersebut (dalam satuan derajat): nilai positif menyatakan perubahan arah ke kanan (searah jarum jam), sedangkan nilai negatif menyatakan perubahan arah ke kiri (berlawanan arah jarum jam). Fungsi akan mengembalikan bilangan integer antara 0 sampai 359, yang menyatakan arah pergerakan yang baru setelah diubah sebesar **a** derajat. Karena sudut paling besar adalah 360 derajat, maka 360 derajat sama dengan 0 derajat.

Contoh

Masukan (s)	Masukan (r)	Keluaran
50	100	150
350	100	90
10	-100	270
358	-359	359



#### -- DEFINISI DAN SPESIFIKASI FUNGSI

**ubahArah :: Int -> Int -> Int**

*{- ubahArah s r adalah fungsi yang menerima masukan dua buah integer s dan r yang merepresentasikan arah pergerakan dan besar perubahan arah yang akan dilakukan (dalam satu derajat). Fungsi mengembalikan arah pergerakan yang baru dalam range 0 s.d. 359, setelah s diputar sebesar r. -}*

Buatlah realisasi dan contoh aplikasi berikut hasilnya (minimum 2 buah) dari fungsi **ubahArah** (definisi dan spesifikasi untuk fungsi tidak perlu ditulis ulang).

- b. Fungsi **pangkat** menerima masukan berupa dua buah integer, a dan b, dan mengembalikan hasil berupa  $a^b$  (a pangkat b). Masukan diasumsikan selalu valid, yaitu  $a > 0$  dan  $b \geq 0$ .

Buatlah definisi, spesifikasi, realisasi dengan menggunakan ekspresi rekursif dan contoh aplikasi berikut hasilnya (minimum 2 buah) dari fungsi **pangkat**.

Untuk soal 1, tidak diperkenankan membuat dan menggunakan fungsi antara baru.

### Soal 2. List

Diketahui definisi primitif tipe **list of integer** berikut ini. Semua primitif list dapat langsung digunakan (tidak perlu dibuat realisasinya).

#### -- DEFINISI DAN SPESIFIKASI PRIMITIF LIST

##### -- KONSTRUKTOR

**konso :: Int -> [Int] -> [Int]**

*{- konso e li menghasilkan sebuah list of integer dari e dan li dengan e sebagai elemen pertama -}*

**konsDot :: [Int] -> Int -> [Int]**

*{- konsDot li e menghasilkan sebuah list of integer dari li dan e dengan e sebagai elemen terakhir -}*

##### -- SELEKTOR

**-- head :: [Int] -> Int**

*{- head l menghasilkan elemen pertama list l, l tidak kosong -}*

**-- tail :: [Int] -> [Int]**

*{- tail l menghasilkan list tanpa elemen pertama list l, l tidak kosong -}*

**-- last :: [Int] -> Int**

*{- last l menghasilkan elemen terakhir list l, l tidak kosong -}*

**-- init :: [Int] -> [Int]**

*{- init l menghasilkan list tanpa elemen terakhir list l, l tidak kosong -}*

##### -- PREDIKAT

**isEmpty :: [Int] -> Bool**

*{- isEmpty l true jika list of elemen l kosong, false jika tidak -}*

**isOneElmt :: [Int] -> Bool**

*{- isOneElmt l true jika list of integer l hanya mempunyai satu elemen, false jika tidak -}*

Tuliskan **realisasi** dari keempat fungsi berikut (definisi dan spesifikasi fungsi tidak perlu ditulis ulang).

**isAllGanjil** :: [Int] -> Bool

{- **isAllGanjil** *l* mengembalikan true apabila seluruh elemen *l* adalah bilangan ganjil.  
Fungsi mengembalikan true jika *l* adalah list kosong -}

**getSmallest** :: [Int] -> Int

{- **getSmallest** *l* mengembalikan elemen terkecil di *l*. Prekondisi: *l* tidak kosong. -}

**delElement** :: Int -> [Int] -> [Int]

{- **delElement** *x l* mengembalikan list *l* dengan elemen *x* yang telah dihapus dari *l*.  
Jika *x* bukan elemen *l*, maka fungsi mengembalikan *l* semula.  
Prekondisi: elemen *l* unik (setiap elemen hanya muncul 1 kali). -}

**sortList** :: [Int] -> [Int]

{- **sortList** *l* mengembalikan hasil pengurutan list *l* hingga elemen-elemennya terurut membesar.  
Prekondisi: *l* tidak kosong dan semua elemennya unik. -}

Contoh:

Masukan (l)	Masukan (x)	isAllGanjil l	getSmallest l	delElement x l	sortList l
[7,3,9,13,15,31,19]	13	True	3	[7,3,9,15,31,19]	[3,7,9,13,15,19,31]
[6,15,4,9]	5	False	4	[6,15,4,9]	[4,6,9,15]
[9,15,27]	27	True	9	[9,15]	[9,15,27]
[ ]	5	True		[ ]	

Catatan: jika sel diberi *shading* artinya fungsi tidak terdefinisi untuk masukan tersebut.

Untuk soal 2, tidak diperkenankan membuat dan menggunakan fungsi antara baru (boleh menggunakan yang sudah tersedia pada soal ini).

### Soal 3. Fungsi dengan Parameter Fungsi

Buatlah definisi, spesifikasi, dan realisasi sebuah fungsi **offsetList** yang menerima masukan dua buah fungsi, misalnya **f** dan **g**, serta dua buah bilangan riil (float), **a** dan **b**. Fungsi **offsetList** akan menghasilkan sebuah **list of float** yang merupakan penerapan fungsi **f** terhadap bilangan float antara **a** dan **b**, dimulai dari **a** dengan *increment* menggunakan fungsi **g**.

Diasumsikan primitif untuk **List of Float** telah terdefinisi dan siap digunakan (lihat primitif pada soal 2, namun untuk tipe data **Float**). Berikut adalah contoh masukan dan hasil dari fungsi:

Masukan				Hasil offsetList f g a b
f	g	a	b	
<b>id</b> :: float -> float -- id x mengirimkan nilai x id x = x	<b>p1</b> :: float -> float -- p1 x mengirimkan nilai x + 1 p1 x = x + 1	1.0	5.0	[1.0, 2.0, 3.0, 4.0, 5.0]
<b>p1</b> :: float -> float -- p1 x mengirimkan nilai x + 1 p1 x = x + 1	<b>p2</b> :: float -> float -- p2 x mengirimkan nilai x + 2 p2 x = x + 2	5.0	10.0	[6.0, 8.0, 10.0]

Berikan pula aplikasi fungsi **offsetList** untuk ketiga kasus di bawah ini dalam bentuk **ekspresi lambda** dan tuliskan pula hasil aplikasinya.

Masukan			
f	g	a	b
Mengirimkan nilai x	Mengirimkan nilai x + 2	1.2	7.1
Jika x < 0, akan mengirimkan nilai -999.0. Jika x ≥ 0 mengirimkan nilai x + 3.2	Mengirimkan nilai x + 0.5	- 1.0	1.0

Berikan pula contoh aplikasi fungsi **offsetList** untuk menghasilkan list [0.0, 1.0, 4.0, 9.0, 16.0, 25.0, 36.0, 49.0, 64.0, 81.0].

Untuk soal 3, tidak diperkenankan membuat dan menggunakan type baru atau fungsi antara baru.

## Sem. 2 2015/2016

### Soal 2. List

Diketahui definisi primitif tipe **list of integer** berikut ini. Semua primitif list dapat langsung digunakan (tidak perlu dibuat realisasinya).

```
-- DEFINISI DAN SPESIFIKASI PRIMITIF LIST
-- KONSTRUKTOR
konso :: Int -> [Int] -> [Int]
{- konso e li menghasilkan sebuah list of integer dari e dan li
   dengan e sebagai elemen pertama -}
konsDot :: [Int] -> Int -> [Int]
{- konsDot li e menghasilkan sebuah list of integer dari li dan e
   dengan e sebagai elemen terakhir -}
-- SELEKTOR
-- head :: [Int] -> Int
{- head l menghasilkan elemen pertama list l, l tidak kosong -}
-- tail :: [Int] -> [Int]
{- tail l menghasilkan list tanpa elemen pertama list l, l tidak kosong -}
-- last :: [Int] -> Int
{- last l menghasilkan elemen terakhir list l, l tidak kosong -}
-- init :: [Int] -> [Int]
{- init l menghasilkan list tanpa elemen terakhir list l, l tidak kosong -}
-- PREDIKAT DASAR
isEmpty :: [Int] -> Bool
{- isEmpty l true jika list of elemen l kosong, false jika tidak -}
isOneElmt :: [Int] -> Bool
{- isOneElmt l true jika list of integer l hanya mempunyai satu elemen, false jika tidak -}
```

Tuliskan **realisasi** dari keempat fungsi berikut (definisi dan spesifikasi fungsi tidak perlu ditulis ulang).

```
nilaiEkstrim :: [Int] -> (Int,Int)
{- nilaiEkstrim l mengembalikan pasangan integer (min,max), dengan min adalah nilai terkecil pada
   l dan max adalah nilai terbesar pada l.
   Prekondisi: l tidak kosong
   Contoh: nilaiEkstrim [3,2,6,5,8,0,1,2,12,56,44,3,28] = (0,56) -}

trunc :: [Int] -> Int -> [Int]
{- trunc l i mengembalikan i buah elemen terdepan dari l.
   Apabila i lebih besar dari jumlah elemen pada l, maka fungsi mengirimkan l.
   Prekondisi: i > 0
   Contoh: trunc [3,2,6,5,8] 3 = [3,2,6] -}

splitList :: [Int] -> ([Int],[Int])
{- splitList l mengembalikan dua buah list lfront dan lback, dengan lfront memuat elemen l
   bagian depan dan lback memuat elemen l bagian belakang.
   Jumlah elemen pada lfront sama dengan atau 1 elemen lebih banyak dari pada lback.
   Contoh: splitList [3,2,6,5,8] = ([3,2,6],[5,8]) -}

insSorted :: Int -> [Int] -> [Int]
{- insSorted x l menerima sebuah integer x dan sebuah list l yang terurut membesar serta
   mengembalikan list l yang telah ditambahkan x sedemikian sehingga elemen-elemennya tetap
   terurut membesar.
   Contoh: insSorted 4 [2,3,5,6] = [2,3,4,5,6] -}
```

Untuk soal 2, tidak diperkenankan membuat dan menggunakan fungsi antara baru (tapi boleh menggunakan fungsi yang tersedia dalam soal ini).

### Soal 3. Aspek Fungsi dengan Parameter Fungsi

- a. Buatlah definisi, spesifikasi, dan realisasi sebuah fungsi **sumInteger** yang menerima 2 (dua) buah integer positif ( $>0$ ), misalnya **m** dan **n**, dan sebuah fungsi **f** dan menghasilkan penjumlahan dari semua integer antara m dan n (termasuk m dan n) yang memenuhi f. Jika dalam selang m dan n tidak ada yang memenuhi f, maka hasilnya adalah 0.

Contoh:

Masukan			sumInteger m n f	Keterangan
m	n	f		
2	9	isGenap :: Int -> Bool {- isGenap x menghasilkan true jika x adalah bilangan genap -}	20	Yang memenuhi isGenap dalam selang [2..9] = {2,4,6,8} sehingga $2+4+6+8 = 20$
2	9	gtThan5 :: Int -> Bool {- gtThan5 x menghasilkan true jika $x > 5$ -}	30	Yang memenuhi gtThan5 dalam selang [2..9] = {6,7,8,9} sehingga $6+7+8+9 = 30$
8	5	isGenap :: Int -> Bool {- isGenap x menghasilkan true jika x adalah bilangan genap -}	0	Selang [8..5] tidak terdefinisi

- b. Tuliskan aplikasi fungsi **sumInteger** pada butir a untuk 3 (tiga) masukan di bawah ini dalam bentuk **ekspresi lambda** dan tuliskan pula hasil aplikasinya.

m	n	f
		Menerima masukan sebuah integer x dan ...
1	100	menghasilkan true jika x dapat membagi habis 100.
1	100	menghasilkan true jika x adalah bilangan genap dan dapat dibagi habis oleh 10 atau jika x adalah bilangan ganjil dan dapat dibagi habis oleh 5. Selain itu, menghasilkan false.
25	25	menghasilkan true, jika x lebih kecil dari 10.

Untuk soal 3, tidak diperkenankan membuat dan menggunakan type baru atau fungsi antara baru.

## Sem. 2 2016/2017

### Soal 2. Ekspresi Fungsional

- a. Tuliskan **definisi**, **spesifikasi**, **realisasi**, dan contoh **aplikasi** (berikut hasilnya) dari fungsi **prevSecond** yang menerima masukan 3 integer (j, m, d) merepresentasikan sebuah waktu dan mengembalikan sebuah tuple yang terdiri dari 3 integer (j1,m1,d1) merepresentasikan waktu 1 detik sebelumnya. Realisasi fungsi tidak boleh menggunakan konversi ke detik. Prekondisi: j, m, d dapat membentuk waktu yang valid, yaitu j bernilai [0..23], m bernilai [0..59], dan d bernilai [0..59].
- b. Tuliskan realisasi dari fungsi **prevNSeconds** berikut ini (definisi dan spesifikasi tidak perlu ditulis ulang). Realisasi fungsi harus menggunakan ekspresi rekursif.

```
prevNSeconds :: Int -> Int -> Int -> Int -> (Int,Int,Int)
{- prevNSeconds j m d n menghasilkan tuple (j1,m1,d1) yang merupakan waktu n detik sebelum
   pukul j:m:d.
   Prekondisi: 0<=j<=23, 0<=m<=59, 0<=d<=59, n>=0 -}
```

#### Contoh

Masukan				Hasil Fungsi	
j	M	d	n	prevSecond j m d	prevNSeconds j m d n
11	11	11	10	(11,11,10)	(11,11,1)
15	10	10	15	(15,10,9)	(15,9,55)
0	0	0	3600	(23,59,59)	(23,0,0)

Anda tidak diperkenankan membuat dan menggunakan fungsi antara baru (tapi boleh menggunakan fungsi yang tersedia dalam soal ini).

### Soal 3. List

Diketahui definisi primitif tipe **list of integer** berikut ini. Semua primitif list dapat langsung digunakan (tidak perlu dibuat realisasinya).

```
-- DEFINISI DAN SPESIFIKASI PRIMITIF LIST
-- KONSTRUKTOR
konso :: Int -> [Int] -> [Int]
{- konso e li menghasilkan sebuah list of integer dari e dan li
   dengan e sebagai elemen pertama -}
konsDot :: [Int] -> Int -> [Int]
{- konsDot li e menghasilkan sebuah list of integer dari li dan e
   dengan e sebagai elemen terakhir -}
-- SELEKTOR
-- head :: [Int] -> Int
{- head l menghasilkan elemen pertama list l, l tidak kosong -}
-- tail :: [Int] -> [Int]
{- tail l menghasilkan list tanpa elemen pertama list l, l tidak kosong -}
-- last :: [Int] -> Int
{- last l menghasilkan elemen terakhir list l, l tidak kosong -}
-- init :: [Int] -> [Int]
{- init l menghasilkan list tanpa elemen terakhir list l, l tidak kosong -}
-- PREDIKAT DASAR
isEmpty :: [Int] -> Bool
{- isEmpty l true jika list of elemen l kosong, false jika tidak -}
isOneElmt :: [Int] -> Bool
{- isOneElmt l true jika list of integer l hanya mempunyai satu elemen, false jika tidak -}
```

Tuliskan **realisasi** dari keempat fungsi berikut (definisi dan spesifikasi fungsi tidak perlu ditulis ulang).

```
isSortedDown :: [Int] -> Bool
{- isSortedDown l mengembalikan True apabila l terurut mengecil.
   Prekondisi: list tidak kosong
   Contoh: isSortedDown [8,5,2,4] = False; isSortedDown [5,2,0] = True -}

getElTengah :: [Int] -> Int
{- getElTengah l mengembalikan elemen l yang berada di tengah-tengah.
   Apabila elemen l berjumlah genap (misalnya n), maka yang dikembalikan adalah elemen pada
   posisi (n div 2).
   Prekondisi: list tidak kosong
   Contoh: getElTengah [3,2,6,5,8] = 6; getElTengah [1,2,3,4] = 2 -}

elPosGanjil :: [Int] -> [Int]
{- elPosGanjil l mengembalikan sebuah list yang berisi semua elemen l pada posisi ganjil.
   Contoh: elPosGanjil [2,4,6,8,1,3,5,7,9] = [2,6,1,5,9] -}

delAllX :: [Int] -> Int -> ([Int],Int)
{- delAllX l x mengembalikan sebuah list l1 dan integer n, dengan l1 memuat elemen-elemen l
   setelah elemen bernilai x dihapus dan n memuat jumlah kemunculan elemen bernilai x pada l.
   Contoh: delAllX [3,2,6,2,3] 3 = ([2,6,2],2); delAllX [] 3 = ([],0) -}
```

Untuk soal 3, tidak diperkenankan membuat dan menggunakan fungsi antara baru (tapi boleh menggunakan fungsi-fungsi yang tersedia dalam soal ini).

#### Soal 4. Aspek Fungsi dengan Parameter Fungsi

- a. Buatlah definisi, spesifikasi, dan realisasi sebuah fungsi **splitListIF** yang menerima sebuah list of integer, misalnya **li** dan sebuah fungsi **f** (menerima sebuah nilai integer dan menghasilkan nilai boolean), dan menghasilkan dua buah list of integer. List pertama berisi elemen li yang memenuhi f, dan list kedua berisi elemen li yang tidak memenuhi f. List li mungkin kosong.  
Diasumsikan primitif untuk list of integer telah terdefinisi dan siap digunakan (lihat soal 3).

Contoh:

Masukan		spliListIF li f
li	f	
[]	isGenap menerima masukan integer x dan menghasilkan true jika x adalah bilangan genap	([], [])
[1,2,3,4,5,6,7]	gtThan5 menerima masukan integer x dan menghasilkan true jika x > 5	([6,7], [1,2,3,4,5])
[0,1,2,3,4,5]	isPositif menerima masukan integer x dan menghasilkan true jika x adalah bilangan positif atau 0	([0,1,2,3,4,5], [])

- b. Tuliskan aplikasi fungsi **splitListIF** pada butir a untuk 3 (tiga) masukan di bawah ini dalam bentuk ekspresi lambda dan tuliskan pula hasil aplikasinya.

li	f
	Menerima masukan sebuah integer x dan ...
[1,2,3,4,5,6,7,8,9,10]	menghasilkan true jika x habis dibagi 3
[-8,-5,-10,8,10,15,24]	menghasilkan true jika x adalah bilangan positif dan habis dibagi 3 atau jika x adalah bilangan negatif dan habis dibagi 2.
[45,55,76,50,90,50,85,75]	menghasilkan true, jika x lebih besar dari 75.

Untuk soal 4, tidak diperkenankan membuat dan menggunakan type baru atau fungsi antara baru.