

Started on	Friday, 15 March 2024, 10:47 PM
State	Finished
Completed on	Friday, 15 March 2024, 11:30 PM
Time taken	43 mins 7 secs
Marks	1100.00/1100.00
Grade	100.00 out of 100.00

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: [ListOfCharacter.hs](#)

Header: module ListOfCharacter where

Salinlah definisi list of character dalam file [ListOfCharacter.hs](#).

Buatlah fungsi **konkat** yang menerima masukan 2 buah list of character, misalnya lc1 dan lc2, yang masing-masing mungkin kosong, dan menghasilkan list baru yang merupakan penggabungan lc1 dengan lc2 (lc1 di awal).

Contoh aplikasi dan hasil:

Aplikasi	Hasil
konkat ['s','a','y'] ['a']	"sayaa"
konkat [] ['a']	"a"
konkat ['a'] []	"a"
konkat [] []	""

Haskell

 [ListOfCharacter.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.92 MB
2	10	Accepted	0.00 sec, 3.03 MB
3	10	Accepted	0.00 sec, 2.91 MB
4	10	Accepted	0.00 sec, 3.02 MB
5	10	Accepted	0.00 sec, 3.05 MB
6	10	Accepted	0.02 sec, 3.02 MB
7	10	Accepted	0.00 sec, 3.01 MB
8	10	Accepted	0.00 sec, 3.01 MB
9	10	Accepted	0.00 sec, 2.96 MB
10	10	Accepted	0.01 sec, 3.04 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: [ListOfCharacter.hs](#)

Header: module ListOfCharacter where

Salinlah definisi list of character dalam file [ListOfCharacter.hs](#).

Buatlah fungsi **isPalindrom** yang menerima masukan sebuah list of character, misalnya lc, dan menghasilkan true jika lc adalah palindrom, yaitu jika dibaca dari awal maupun dari akhir sama. List kosong adalah palindrom. List 1 elemen adalah list palindrom.

Contoh aplikasi dan hasil:

Aplikasi	Hasil
isPalindrom ['s','a','y']	False
isPalindrom ['k','o','k']	True
isPalindrom ['k']	True
isPalindrom []	True

Haskell

 [ListOfCharacter.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.99 MB
2	10	Accepted	0.00 sec, 2.90 MB
3	10	Accepted	0.00 sec, 2.89 MB
4	10	Accepted	0.00 sec, 2.93 MB
5	10	Accepted	0.00 sec, 3.03 MB
6	10	Accepted	0.00 sec, 2.90 MB
7	10	Accepted	0.00 sec, 2.95 MB
8	10	Accepted	0.00 sec, 3.04 MB
9	10	Accepted	0.00 sec, 2.96 MB
10	10	Accepted	0.00 sec, 2.94 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: [ListOfInteger.hs](#)

Header: module ListOfInteger where

Salinlah definisi list of integer dalam file [ListOfInteger.hs](#).

Buatlah fungsi **isMember** yang menerima masukan sebuah list of integer, misalnya l, dan sebuah integer, misalnya x, dan menghasilkan true jika x adalah salah satu member (anggota) dalam list l. Menghasilkan false jika tidak, atau jika list l kosong.

Contoh aplikasi dan hasil:

Aplikasi	Hasil
isMember [] 5	False
isMember [1,2] 5	False
isMember [1,2,3] 1	True

Haskell

 [ListOfInteger.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.98 MB
2	10	Accepted	0.00 sec, 2.99 MB
3	10	Accepted	0.00 sec, 2.97 MB
4	10	Accepted	0.00 sec, 3.05 MB
5	10	Accepted	0.00 sec, 3.08 MB
6	10	Accepted	0.00 sec, 2.93 MB
7	10	Accepted	0.00 sec, 2.93 MB
8	10	Accepted	0.00 sec, 2.95 MB
9	10	Accepted	0.00 sec, 2.88 MB
10	10	Accepted	0.00 sec, 3.05 MB

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: [ListOfInteger.hs](#)

Header: module ListOfInteger where

Salinlah definisi list of integer dalam file [ListOfInteger.hs](#).

Buatlah fungsi **isEqual** yang menerima masukan 2 buah list of integer, misalnya l1 dan l2, dan menghasilkan true jika l1 sama dengan l2, yaitu jika banyaknya elemen l1 = banyaknya elemen l2 dan jika tiap elemen pada urutan yang sama adalah sama. l1 dan l2 mungkin kosong.

Contoh aplikasi dan hasil:

Aplikasi	Hasil
isEqual [] []	True
isEqual [] [1]	False
isEqual [1] []	False
isEqual [1,2,3] [1,2,3]	True

Haskell

 [ListOfInteger.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.98 MB
2	10	Accepted	0.00 sec, 3.02 MB
3	10	Accepted	0.00 sec, 2.90 MB
4	10	Accepted	0.00 sec, 2.95 MB
5	10	Accepted	0.00 sec, 3.00 MB
6	10	Accepted	0.00 sec, 2.92 MB
7	10	Accepted	0.00 sec, 2.89 MB
8	10	Accepted	0.00 sec, 3.00 MB
9	10	Accepted	0.00 sec, 3.11 MB
10	10	Accepted	0.00 sec, 3.10 MB

Question **5**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: [ListOfInteger.hs](#)

Header: module ListOfInteger where

Salinlah definisi list of integer dalam file [ListOfInteger.hs](#).

Buatlah fungsi **maxNb** yang menerima masukan sebuah list of integer, misalnya l, yang tidak kosong dan menghasilkan pasangan nilai/tuple (max,nbmax) dengan max berisi nilai maksimum elemen l dan nbmax adalah banyaknya kemunculan max dalam l.

Contoh aplikasi dan hasil:

Aplikasi	Hasil
maxNb [1,2,2]	(2,2)
maxNb [3,3,-4,-5,3,3,0]	(3,4)

HINTS

Untuk membantu, dapat membuat fungsi berikut:

-- Definisi dan Spesifikasi

minList :: [Int] -> Int

{- **minList l** mengembalikan nilai minimum dari seluruh elemen list -}

nbX :: Int -> [Int] -> Int

{- **nbX x l** menghasilkan banyaknya kemunculan x pada l -}

Haskell

 [ListOfInteger.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 3.00 MB
2	10	Accepted	0.00 sec, 3.00 MB
3	10	Accepted	0.00 sec, 2.88 MB
4	10	Accepted	0.00 sec, 3.02 MB
5	10	Accepted	0.00 sec, 3.08 MB
6	10	Accepted	0.00 sec, 3.08 MB
7	10	Accepted	0.00 sec, 2.93 MB
8	10	Accepted	0.00 sec, 2.93 MB

No	Score	Verdict	Description
9	10	Accepted	0.00 sec, 3.03 MB
10	10	Accepted	0.00 sec, 3.02 MB

Question **6**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: ListOfInteger

Header: module ListOfInteger where

Salinlah definisi list of integer dalam file [ListOfInteger.hs](#).

Buatlah realisasi fungsi **jmlMin** yaitu fungsi yang menerima sebuah list of integer tidak kosong, dan menghasilkan pasangan bilangan integer (a,b), dengan a berisi nilai minimum dari seluruh elemen list dan b berisi jumlah kemunculan nilai minimum tersebut pada l. Fungsi ini akan memanggil fungsi **minList** yang menghasilkan nilai minimum dari seluruh elemen list, dan fungsi **nbX** yang menghasilkan banyaknya kemunculan sebuah elemen pada list.

```
-- Definisi dan Spesifikasi

minList :: [Int] -> Int

{- minList l mengembalikan nilai minimum dari seluruh elemen list -}

nbX :: Int -> [Int] -> Int

{- nbX x l menghasilkan banyaknya kemunculan x pada l -}

jmlMin :: [Int] -> (Int,Int)

{- jmlMin l menghasilkan tuple (a,b) dengan:
    a adalah nilai minimum dari elemen-elemen l dan
    b adalah jumlah kemunculan a pada l -}
```

Contoh aplikasi fungsi:

```
> jmlMin [2,4,6,8,4,2,2,10]

(2,3)

> jmlMin [5]

(5,1)
```

Haskell

 [ListOfInteger.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.00 sec, 3.07 MB
2	25	Accepted	0.00 sec, 3.13 MB
3	25	Accepted	0.00 sec, 3.10 MB

No	Score	Verdict	Description
4	25	Accepted	0.00 sec, 2.98 MB

Question **7**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama header : module Duel where

Nama file : Duel.hs

Tuan Vin adalah seorang koboi. Saat berduel dengan lawannya, ia harus bersiap-siap mendengar kata "desperado". Apabila ia mendengar kata "desperado", maka ia harus mengatakan "BANG".

Bantulah Tuan Vin untuk mengubah kata "desperado" menjadi "BANG" dari sebuah list agar Tuan Vin menang dengan nama fungsi **duel** yang menerima list bertipe String dan mengeluarkan list bertipe String

Contoh :

li = ["one", "two", "desperado", "cowboy", "guns", "horse", "desperado", "desperado", "desperado"]

Hasil Keluaran = ["one", "two", "BANG", "cowboy", "guns", "horse", "BANG", "BANG", "BANG"]

Hint: Gunakan operator : untuk melakukan konkatenasi String dengan List of String

Haskell

 [Duel.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.95 MB
2	10	Accepted	0.00 sec, 2.91 MB
3	10	Accepted	0.00 sec, 2.92 MB
4	10	Accepted	0.00 sec, 2.98 MB
5	10	Accepted	0.00 sec, 2.98 MB
6	10	Accepted	0.00 sec, 3.04 MB
7	10	Accepted	0.00 sec, 3.03 MB
8	10	Accepted	0.00 sec, 2.91 MB
9	10	Accepted	0.00 sec, 3.00 MB
10	10	Accepted	0.00 sec, 2.94 MB

Time limit	1 s
Memory limit	64 MB

Nama File: [ListOfCharacter.hs](#)

Header: module ListOfCharacter where

Tuan Vin ingin makan di sebuah restoran baru, namun ia uangnya terbatas. Harga makanan yang ditampilkan pada menu restoran belum termasuk pajak dan service.

Bantulah Tuan Vin menghitung makanan mana yang bisa ia beli bila diketahui uangnya 200 rupiah, dan pajak makanan adalah 10%.

Buatlah sebuah fungsi **pajakMakan** yang menerima list nama makanan bertipe char dan list harganya bertipe integer.

Fungsi mengeluarkan list char berisi makanan mana saja yang bisa dibeli Tuan Vin setelah kena pajak.

Keterangan:

- Jumlah menu makanan dan daftar harga pasti sama
- Nama menu makanan pasti unik
- Bila tidak ada makanan yang dapat dibeli, fungsi mengembalikan list kosong

Contoh:

```
> pajakMakan ['A','Z','B'] [200,150,100]
"ZB"

> pajakMakan [] []
""
```

Haskell

 [ListOfCharacter.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 3.02 MB
2	10	Accepted	0.00 sec, 3.05 MB
3	10	Accepted	0.00 sec, 2.94 MB
4	10	Accepted	0.00 sec, 3.07 MB
5	10	Accepted	0.00 sec, 3.07 MB
6	10	Accepted	0.00 sec, 3.02 MB
7	10	Accepted	0.00 sec, 2.93 MB
8	10	Accepted	0.00 sec, 2.96 MB
9	10	Accepted	0.00 sec, 2.96 MB
10	10	Accepted	0.00 sec, 3.05 MB

Time limit	1 s
Memory limit	64 MB

Nama File: [ListOfCharacter.hs](#)

Header: module ListOfCharacter where

Salinlah definisi list of character dalam file [ListOfCharacter.hs](#).

Buatlah fungsi **splitAlternate** berikut ini yang menerima masukan sebuah list of character, misalnya l dan menghasilkan dua buah list of character, misalnya l1 dan l2. l1 berisi semua elemen l pada posisi ganjil, l2 berisi semua elemen l pada posisi genap.

Definisinya adalah sebagai berikut:

```
splitAlternate :: [Char] -> ([Char],[Char])

{- splitAlternate(l) menghasilkan dua buah list, misalnya l1 dan l2. l1 berisi
   semua elemen l pada posisi ganjil, l2 berisi semua elemen l pada posisi genap.
   l mungkin kosong.
-}
```

Contoh aplikasi dan hasil:

Aplikasi	Hasil
splitAlternate ['x','y','z']	("xz","y")
splitAlternate ['k','l','x','z','y']	("kxy","lz")
splitAlternate ['n']	("n","")
splitAlternate ['p','q','r','s']	("pr","qs")
splitAlternate []	("","")

Tips: Perhatikan perbedaan list dengan panjang genap dan ganjil!

Haskell

 [ListOfCharacter.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.00 sec, 3.03 MB
2	25	Accepted	0.00 sec, 2.94 MB
3	25	Accepted	0.00 sec, 2.97 MB
4	25	Accepted	0.00 sec, 2.97 MB

Time limit	1 s
Memory limit	64 MB

Nama File: [ListOfInteger.hs](#)

Header: *module ListOfInteger where*

Salinlah definisi list of integer dalam file [ListOfInteger.hs](#).

Buatlah sebuah fungsi **pecahListPosNeg** (definisi, spesifikasi, dan realisasi) yang menerima masukan sebuah list of integer (l) dan mengembalikan dua buah list of integer (l1, l2). l1 memuat semua elemen l yang bernilai positif atau 0 sedangkan l2 memuat semua elemen l yang bernilai negatif. Urutan kemunculan elemen pada l1 dan l2 tetap sama dengan urutan elemen pada l.

No	Input	Output
1.	pecahListPosNeg [1,3,0,4,-1,4,-9]	([1,3,0,4,4],[-1,-9])
2.	pecahListPosNeg [2,3,4,5]	([2,3,4,5],[])
3.	pecahListPosNeg []	([],[])

Haskell

 [ListOfInteger.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.00 sec, 2.97 MB
2	25	Accepted	0.00 sec, 2.95 MB
3	25	Accepted	0.00 sec, 2.99 MB
4	25	Accepted	0.00 sec, 2.95 MB

Time limit	1 s
Memory limit	64 MB

Soal ini soal bonus. Kerjakan hanya bila soal-soal sebelumnya sudah selesai dikerjakan.

Nama File: AlternateSort.hs

Header: module AlternateSort where

Diberikan sebuah list, Pak Engi memiliki sebuah algoritma prosedural sebagai berikut.

- 1. Urutkan list tersebut
- 2. Bagi list menjadi 2 sama besar, misal l1 dan l2. Jika panjang list ganjil, maka l1 akan memiliki 1 elemen lebih banyak dibanding l2
- 3. Ambil elemen terkecil dari l1, masukkan ke akhir l3.
- 4. Ambil elemen terbesar dari l2, masukkan ke akhir l3.
- 5. Ulangi langkah 3 dan 4 sampai kedua list kosong.

Contohnya, jika list awal adalah [9,10,11,12], maka l3 akan menjadi [9,12,10,11]

Pak Engi telah selesai membuat algoritma prosedural tersebut. Anda, sebagai pemrogram handal, ingin membuat versi fungsional dari kode tersebut. Namun, anda menyadari bahwa langkah prosedural tersebut terlalu kompleks untuk diimplementasikan dalam waktu 2 jam, sehingga anda ingin mencari cara lain untuk mengimplementasikan algoritma tersebut. Buatlah program yang dapat melakukan algoritma tersebut!

Contoh aplikasi fungsi dan hasilnya:

```
> alternateSort[9,10,11,12]
[9,12,10,11]
> alternateSort[5,2,5,2,1]
[1,5,2,5,2]
```

Haskell

 [AlternateSort.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.87 MB
2	10	Accepted	0.00 sec, 3.11 MB
3	10	Accepted	0.00 sec, 3.44 MB
4	10	Accepted	0.00 sec, 3.39 MB
5	10	Accepted	0.00 sec, 2.88 MB

No	Score	Verdict	Description
6	10	Accepted	0.00 sec, 3.07 MB
7	10	Accepted	0.00 sec, 2.91 MB
8	10	Accepted	0.00 sec, 2.99 MB
9	10	Accepted	0.00 sec, 3.43 MB
10	10	Accepted	0.00 sec, 3.34 MB

[◀ Slide Responsi 2 - Shift 1](#)

Jump to...

[Slide Responsi 2 - Shift 2 ▶](#)