

[Dashboard](#) / [My courses](#) / [ITB IF1210_2_2324](#) / [Responsi 1](#) / [Responsi 1 - Shift 1](#)

Started on	Friday, 1 March 2024, 7:47 AM
State	Finished
Completed on	Friday, 1 March 2024, 8:32 AM
Time taken	44 mins 37 secs
Marks	1300.00/1300.00
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: LuasSegitiga.hs


Header: module LuasSegitiga where

Buatlah definisi, spesifikasi, dan realisasi fungsi **luasSegitiga** yang menerima masukan 2 buah bilangan real (float) a dan t dengan a = alas segitiga dan t = tinggi segitiga (asumsikan: a > 0, t > 0) dan menghasilkan luas segitiga berdasarkan rumus: luas = ½ * a * t

Contoh aplikasi fungsi dan hasilnya:

```
> luasSegitiga 3 4
6.0
```

Haskell ↕

 [LuasSegitiga.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 3.07 MB
2	20	Accepted	0.00 sec, 3.13 MB
3	20	Accepted	0.00 sec, 3.11 MB
4	20	Accepted	0.00 sec, 3.08 MB
5	20	Accepted	0.00 sec, 3.09 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: LuasTrapesium.hs


Header: module LuasTrapesium where

Buatlah definisi, spesifikasi, dan realisasi fungsi **luasTrapesium** yang menerima masukan 3 buah bilangan real (float) t, s1, s2 dengan t = tinggi trapesium, s1 = panjang sisi sejajar 1, dan s2 = panjang sisi sejajar 2 (asumsikan: t > 0, s1 > 0, s2 > 0, dan s1 tidak sama dengan s2) dan menghasilkan luas trapesium berdasarkan rumus: luas = ½ * t * (s1 + s2)

Contoh aplikasi fungsi dan hasilnya:

```
> luasTrapesium 2 4 3
7.0
```

Haskell ↕

 [LuasTrapesium.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	33	Accepted	0.00 sec, 3.13 MB
2	33	Accepted	0.00 sec, 3.13 MB
3	34	Accepted	0.00 sec, 3.21 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: KonversiSuhu.hs

Header: module KonversiSuhu where

Buatlah sebuah fungsi **konversiSuhu** (definisi, spesifikasi, dan realisasi) yang digunakan untuk mengkonversi suhu dari satu satuan Celcius ke satuan suhu yang lain, yaitu Fahrenheit, Reamur, atau Kelvin. Berikut adalah rumus untuk melakukan konversi jika suhu dalam derajat Celcius adalah C:

Suhu Tujuan	Rumus Konversi
Reamur	$4/5 * C$
Fahrenheit	$(9/5 * C) + 32$
Kelvin	$C + 273.15$

Fungsi ini menerima masukan:

- 1 buah nilai bertipe real (float), misalnya t, yang merupakan besaran suhu dalam derajat Celcius.
- 1 buah kode satuan suhu konversi, bertipe karakter, misalnya k, yang diasumsikan bernilai 'R' (Reamur), 'F' (Fahrenheit), atau 'K' (Kelvin).

Fungsi menghasilkan suhu dalam satuan k yang merupakan konversi suhu t derajat Celcius.


Contoh:

konversiSuhu (25, 'R') artinya adalah konversi suhu 25 derajat Celcius ke suhu dalam derajat Reamur.

Contoh aplikasi fungsi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	konversiSuhu 25 'R'	20	$25\text{ }^{\circ}\text{C} = 4/5 * 25^{\circ}\text{R}$ (=) $25^{\circ}\text{C} = 20^{\circ}\text{R}$
2.	konversiSuhu 37 'F'	98.6	$37\text{ }^{\circ}\text{C} = (9/5 * 37^{\circ}\text{F}) + 32^{\circ}\text{F}$ (=) $37^{\circ}\text{C} = 98.6^{\circ}\text{F}$
3.	konversiSuhu (-30) 'K'	243.15	$-30\text{ }^{\circ}\text{C} = -30\text{K} + 273.15\text{K}$ (=) $0^{\circ}\text{C} = 243.15\text{K}$

Haskell

 [KonversiSuhu.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 3.12 MB

No	Score	Verdict	Description
2	10	Accepted	0.00 sec, 3.11 MB
3	10	Accepted	0.00 sec, 2.96 MB
4	10	Accepted	0.00 sec, 3.02 MB
5	10	Accepted	0.00 sec, 2.95 MB
6	10	Accepted	0.00 sec, 3.07 MB
7	10	Accepted	0.00 sec, 2.97 MB
8	10	Accepted	0.00 sec, 3.09 MB
9	10	Accepted	0.00 sec, 3.06 MB
10	10	Accepted	0.00 sec, 3.01 MB

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: Max3.hs

Header: module Max3 where

Tuliskanlah fungsi **max3** yang menerima 3 buah bilangan integer yang berbeda, dan menuliskan nilai terbesar di antara ketiganya. Tidak boleh membuat fungsi antara. Berikut adalah definisi dan spesifikasi fungsi dalam notasi fungsional:

```
max3 : 3 integer -> integer
{ max3 (a,b,c) mengirimkan nilai yang paling besar di antara a, b, dan c. Asumsi: a, b, c bilangan berbeda }
```

Contoh aplikasi dan hasil:

No	Aplikasi	Hasil	Keterangan
1.	max3 1 3 5	5	Bilangan terbesar adalah 5
2.	max3 (-10) 0 (-5)	0	Bilangan terbesar adalah 0
3.	max3 20 34 33	34	Bilangan terbesar adalah 34

Haskell ↕

 [Max3.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 2.88 MB
2	20	Accepted	0.00 sec, 2.85 MB
3	20	Accepted	0.00 sec, 2.94 MB
4	20	Accepted	0.00 sec, 2.96 MB
5	20	Accepted	0.00 sec, 2.88 MB

Question **5**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File : JamBangun.hs

Nama Modul : JamBangun

Jane adalah seorang yang sangat teratur, ia selalu tidur **tepat** selama 8 Jam, pada hari ini ia tidur pada pukul **23.45.00**.

Namun, teman sekamar Jane tiba tiba melihat bahwa ada pesan masuk pada ponsel Jane dari dosen pembimbingnya. Jane diminta untuk menelepon dosen tersebut besok pada jam tertentu.

Ada kemungkinan bahwa Jane terlambat bangun dan melewati panggilan yang sangat penting ini sehingga teman Jane harus membangunkan Jane jika Jane akan terlambat bangun.

Bantulah teman Jane untuk membuat program yang dapat menerima input 3 Integer berupa jam(0..23) menit(0..59) detik(0.59) yang merupakan jam yang ditentukan oleh dosen pembimbing Jane

Lalu program dapat mengeluarkan output berupa **tuple** yang berisi:

Apakah Jane akan bangun melewati jam yang telah ditentukan (True: ya, False: Tidak)

Waktu selisih antara waktu Jane bangun dan waktu yang ditentukan pak Dosen.


Asumsi: input selalu valid.

Contoh:

> jamBangun 07 15 00

(True,0,30,0)

Haskell ↕

 [JamBangun.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.97 MB
2	10	Accepted	0.00 sec, 2.99 MB
3	10	Accepted	0.00 sec, 2.98 MB
4	10	Accepted	0.00 sec, 2.98 MB
5	10	Accepted	0.00 sec, 2.86 MB
6	10	Accepted	0.00 sec, 2.90 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, 2.99 MB
10	10	Accepted	0.04 sec, 2.99 MB

Question **6**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File : LamaTidur.hs

Nama Modul : LamaTidur

Hari ini para panitia Arkavidia bekerja keras siang malam mempersiapkan acara besok hari. Namun, panitia jugalah seorang manusia yang pastinya butuh tidur yang cukup. Besok acara dimulai pada pukul 05.00 pagi, dan saat mereka selesai mempersiapkan acara besok. Para panitia merasa lelah dan langsung tidur di tempat. Para panitia ingin mengetahui apakah mereka dapat tidur dengan cukup ($>= 6$ jam) dan menghitung lama mereka bisa tidur.


Bantulah panitia dengan membuatlah sebuah fungsi dengan nama **lamaTidur** yang dapat menerima input 3 integer yang merupakan jam (0..23), menit(0..59), detik(0..59) waktu mereka selesai mempersiapkan acara dan dapat mengeluarkan output tuple berisi lama waktu bisa tidur dalam format jam, menit, detik (selisih waktu input dari pukul 05.00 pagi).

Asumsi: input selalu valid.

Contoh:

```
> lamaTidur 1 0 0
(False, 4,0,0)
```

Haskell

 [LamaTidur.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.88 MB
2	10	Accepted	0.00 sec, 2.89 MB
3	10	Accepted	0.00 sec, 2.96 MB
4	10	Accepted	0.00 sec, 2.89 MB
5	10	Accepted	0.00 sec, 2.98 MB
6	10	Accepted	0.00 sec, 2.89 MB
7	10	Accepted	0.00 sec, 2.99 MB
8	10	Accepted	0.00 sec, 2.87 MB
9	10	Accepted	0.00 sec, 2.98 MB
10	10	Accepted	0.00 sec, 2.89 MB

Question **7**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File : KlasifikasiKomputer.hs

Nama Modul : KlasifikasiKomputer

Sebuah perusahaan pembuat video game ingin melakukan optimisasi pada komputer pengguna yang memiliki game mereka. Optimisasi dilakukan dengan cara mengklasifikasikan komputer pengguna dan mengubah pengaturan game sesuai dengan kelompok kemampuannya. Setiap komputer dapat masuk ke salah satu dari 5 kelompok, yaitu 1, 2, 3, 4, dan 5.

Klasifikasi komputer akan dilakukan berdasarkan kemampuan CPU, kemampuan GPU, dan kemampuan harddisk. Ketentuan klasifikasinya adalah sebagai berikut:

1. Jika kemampuan CPU lebih dari 7, kemampuan GPU lebih dari 7, dan kemampuan harddisk lebih dari 7, maka termasuk kelompok 5
2. Jika setidaknya satu kemampuan bernilai **kurang dari sama dengan** dari 7, maka termasuk kelompok 4
3. Jika nilai kemampuan CPU, GPU, **dan** harddisk kurang dari sama dengan 7, maka termasuk kelompok 3
4. Jika nilai kemampuan CPU **atau** GPU kurang dari 5, maka termasuk kelompok 2
5. Jika ada **salah satu nilai kemampuan yang bernilai kurang dari 2**, maka termasuk kelompok 1 (tidak peduli dengan nilai kemampuan lainnya)


Prioritas kelompok terurut dari nomor kelompok **terkecil** hingga terbesar.

Misalnya, apabila masukan cocok dengan kelompok 2 dan 4, maka yang dianggap benar adalah kelompok 2.

Contoh:

```
> klasifikasi 8 9 4
4
> klasifikasi 4 10 2
2
> klasifikasi 6 10 1
1
```

Haskell

 [KlasifikasiKomputer.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	6	Accepted	0.00 sec, 2.84 MB
2	6	Accepted	0.00 sec, 2.84 MB
3	6	Accepted	0.00 sec, 2.96 MB
4	6	Accepted	0.00 sec, 2.91 MB
5	6	Accepted	0.00 sec, 2.97 MB
6	6	Accepted	0.00 sec, 2.97 MB
7	6	Accepted	0.00 sec, 2.84 MB

3/1/24, 8:32 AM

Responsi 1 - Shift 1

No	Score	Verdict	Description
8	6	Accepted	0.00 sec, 2.95 MB
9	6	Accepted	0.00 sec, 2.87 MB
10	6	Accepted	0.00 sec, 2.95 MB
11	6	Accepted	0.00 sec, 2.86 MB
12	6	Accepted	0.00 sec, 2.95 MB
13	6	Accepted	0.00 sec, 2.83 MB
14	6	Accepted	0.00 sec, 2.95 MB
15	6	Accepted	0.00 sec, 2.97 MB
16	10	Accepted	0.00 sec, 2.96 MB

3/1/24, 8:32 AM

Responsi 1 - Shift 1

Question 8

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: UkuranBaju.hs

Header: module UkuranBaju where

Panitia Gemastik ITB ingin mengotomatisasi penentuan ukuran baju panitia dengan memanfaatkan tinggi dalam cm (integer positif) dan berat badan dalam kg (integer positif) dengan ketentuan sebagai berikut:

- Baju ukuran M diberikan kepada mereka yang tingginya ≤ 150 cm (berapa pun berat badannya).
- Baju ukuran XL diberikan ke mereka yang tingginya > 170 cm dan berat badannya > 60 kg tapi masih ≤ 80 kg.
- Jika seseorang tingginya > 150 cm, tapi masih ≤ 170 cm dan berat badannya ≤ 80 kg, maka orang ini mendapatkan baju ukuran L.
- Jika seseorang tingginya > 150 cm, tapi masih ≤ 170 cm dan berat badannya > 80 kg, dia mendapatkan baju ukuran XL.
- Orang yang tingginya > 170 cm dan berat badannya ≤ 60 kg, mendapat baju ukuran L.
- Karena keterbatasan pembuat kaos, tidak ada kaos lain selain M, L, dan XL sehingga untuk yang tidak memenuhi kategori di atas tidak akan mendapatkan kaos. Dalam hal ini, untuk yang bersangkutan diberikan kategori khusus yaitu 4.

Buatlah fungsi **ukuranBaju** yang menerima masukan 2 buah integer positif, misalnya **t** (tinggi badan dalam cm) dan **b** (berat badan dalam kg) dan menghasilkan kode ukuran baju (1 adalah M, 2 adalah L, 3 adalah XL) atau kode 4 adalah untuk yang tidak mendapatkan kaos.

Contoh aplikasi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	ukuranBaju 160 75	2	Ukuran L
2.	ukuranBaju 145 45	1	Ukuran M
3.	ukuranBaju 190 85	4	Tidak mendapatkan kaos

Haskell

UkuranBaju.hs

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.82 MB
2	10	Accepted	0.00 sec, 2.84 MB
3	10	Accepted	0.01 sec, 2.95 MB
4	10	Accepted	0.00 sec, 2.82 MB
5	10	Accepted	0.00 sec, 2.94 MB
6	10	Accepted	0.00 sec, 2.83 MB
7	10	Accepted	0.00 sec, 2.95 MB

3/1/24, 8:32 AM

Responsi 1 - Shift 1

No	Score	Verdict	Description
8	10	Accepted	0.01 sec, 2.85 MB
9	10	Accepted	0.01 sec, 2.84 MB
10	10	Accepted	0.00 sec, 2.83 MB

3/1/24, 8:32 AM

Responsi 1 - Shift 1

Question 9

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: DeretSegitiga.hs

Header: module DeretSegitiga where

Buatlah definisi, spesifikasi, dan realisasi dari fungsi **deretSegitiga** yang merupakan fungsi untuk mencari nilai bilangan ke-n pada deret segitiga. Deret segitiga adalah: 1, 3, 6, 10, 15, ...


Fungsi ini harus diselesaikan menggunakan pendekatan rekursif.

```
deretSegitiga :: Int -> Int
-- deretSegitiga(n) menghasilkan nilai bilangan ke-n pada deret segitiga
-- prekondisi: n > 0
```

Contoh aplikasi fungsi dan hasilnya:

No	Aplikasi	Hasil
1.	deretSegitiga 1	1
2.	deretSegitiga 5	15
3.	deretSegitiga 100	5050

Haskell

 [DeretSegitiga.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 2.84 MB
2	20	Accepted	0.00 sec, 2.94 MB
3	20	Accepted	0.00 sec, 2.81 MB
4	20	Accepted	0.00 sec, 2.98 MB
5	20	Accepted	0.00 sec, 3.02 MB

Question **10**
Correct
Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File : Fibonacci.hs
Nama Modul : Fibonacci
Buatlah definisi, spesifikasi, dan realisasi fungsi fibonacci yang menerima masukan 1 nilai integer nonnegatif yang merupakan urutan n dalam barisan Fibonacci. Barisan Fibonacci yang dimaksud memiliki suku pertama dan kedua berturut-turut bernilai 1 dan 1.
Realisasi fungsi **fibonacci** menghasilkan nilai dari suku ke n.

Contoh:
> fibonacci 10
55

Haskell

Fibonacci.hs

Score: 100

Blackbox
Score: 100
Verdict: Accepted
Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.93 MB
2	10	Accepted	0.00 sec, 2.93 MB
3	10	Accepted	0.00 sec, 2.94 MB
4	10	Accepted	0.00 sec, 2.84 MB
5	10	Accepted	0.00 sec, 2.82 MB
6	10	Accepted	0.00 sec, 2.84 MB
7	10	Accepted	0.00 sec, 2.96 MB
8	10	Accepted	0.00 sec, 2.81 MB
9	10	Accepted	0.00 sec, 2.95 MB
10	10	Accepted	0.00 sec, 2.81 MB

Question **11**
Correct
Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: NbKelipatanX.hs
Header: module NbKelipatanX where
Buatlah definisi, spesifikasi, dan realisasi fungsi **nbKelipatanX** yang menerima masukan dua buah integer positif (integer > 0), misalnya **m** dan **n**, serta sebuah integer positif lain, yaitu **x**, dan menghasilkan banyaknya bilangan kelipatan x di antara m dan n (m dan n termasuk) dengan menggunakan ekspresi rekursif.

Bilangan y disebut kelipatan bilangan x, jika y habis dibagi dengan x.
Prekondisi/syarat/asumsi yang berlaku adalah m <= n dan x <= n.

Contoh aplikasi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	nbKelipatanX 1 1 1	1	Kelipatan 1 di antara [1..1] adalah 1
2.	nbKelipatanX 1 10 2	5	Kelipatan 2 di antara [1..10] adalah 2, 4, 6, 8, 10
3.	nbKelipatanX 5 14 3	3	Kelipatan 3 di antara [5..14] adalah 6, 9, 12

Haskell

NbKelipatanX.hs

Score: 100

Blackbox
Score: 100
Verdict: Accepted
Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 2.87 MB
2	20	Accepted	0.00 sec, 2.94 MB
3	20	Accepted	0.00 sec, 2.96 MB
4	20	Accepted	0.00 sec, 2.96 MB
5	20	Accepted	0.00 sec, 2.93 MB

Time limit	1 s
Memory limit	64 MB

Nama File: SumKelipatanX.hs

Header: module SumKelipatanX where

Buatlah definisi, spesifikasi, dan realisasi fungsi **sumKelipatanX** yang menerima masukan dua buah integer positif (integer > 0), misalnya **m** dan **n**, serta sebuah integer positif lain, yaitu **x**, dan menghasilkan jumlah total bilangan kelipatan x di antara m dan n (m dan n termasuk) dengan menggunakan ekspresi rekursif.

Bilangan y disebut kelipatan bilangan x, jika y habis dibagi dengan x.

Prekondisi/syarat/asumsi yang berlaku adalah $m \leq n$ dan $x \leq n$.

Contoh aplikasi fungsi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	sumKelipatanX 1 1 1	1	Kelipatan 1 di antara [1..1] adalah 1
2.	sumKelipatanX 1 10 2	30	Kelipatan 2 di antara [1..10] adalah 2, 4, 6, 8, 10 $2+4+6+8+10 = 30$
3.	sumKelipatanX 5 14 3	27	Kelipatan 3 di antara [5..14] adalah 6, 9, 12 $6+9+12 = 27$

Haskell ▾

 [SumKelipatanX.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 2.97 MB
2	20	Accepted	0.00 sec, 2.93 MB
3	20	Accepted	0.00 sec, 2.97 MB
4	20	Accepted	0.00 sec, 2.94 MB
5	20	Accepted	0.00 sec, 2.92 MB

Time limit	1 s
Memory limit	64 MB

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

Nama File : HitungBensin.hs

Nama Modul : HitungBensin

Setelah setahun tidak pulang kampung, akhirnya Tuan Vin pun memberanikan diri untuk meminta cuti kepada bosnya. Bosnya sebenarnya ingin langsung menyetujui cuti Tuan Vin. Akan tetapi, dia ingin Tuan Vin membereskan pekerjaannya terlebih dahulu. Pekerjaan Tuan Vin sebenarnya cukup mudah. Dia hanya perlu menyiapkan bensin untuk seluruh kendaraan perusahaannya.

Kendaraan perusahaan Tuan Vin memiliki rute yang sangat unik. Awalnya, kendaraan tersebut akan terletak pada posisi X. Kemudian jika X adalah bilangan genap, kendaraan tersebut akan bergerak ke titik X/2. Jika X adalah bilangan ganjil, kendaraan tersebut akan bergerak ke posisi (3X + 1). Hal ini terus dilakukan sampai kendaraan tersebut sampai ke kantor pusat yang terletak pada posisi 1. Untuk setiap perpindahan posisi, kendaraan tersebut akan menghabiskan bensin sebanyak 1 unit. Jika pada awalnya suatu kendaraan terletak pada posisi 11, kendaraan tersebut akan berpindah ke (11*3 + 1) = 34. Kemudian, kendaraan tersebut kemudian berpindah ke posisi 17,52,26,13,40,20,10,5,16,8,4,2 dan berakhir pada posisi 1 sehingga kendaraan tersebut menghabiskan bensin sebanyak 14 unit.

Tiap harinya, akan ada kendaraan yang berangkat dari posisi A sampai dengan posisi B. Tuan Vin kemudian menjumlahkan banyaknya bensin yang dibutuhkan untuk tiap-tiap kendaraan dari posisi A sampai dengan posisi B. Tuan Vin takut dia tidak sempat menyelesaikan kalkulasinya sebelum hari cutinya tiba. Sebagai teman baik Tuan Vin, Anda pun ingin membantu Tuan Vin dengan membuat sebuah fungsi untuk menghitung bensin yang perlu disiapkan. Fungsi **hitungBensin** menerima 2 buah bilangan bulat, A dan B ($A \leq B$). Fungsi ini kemudian mengeluarkan sebuah bilangan bulat yang menunjukkan konsumsi bensin dari tiap-tiap kendaraan dari A sampai B. Tentu saja sebagai programmer yang baik, Anda harus membuat definisi, spesifikasi, dan realisasi fungsi ini.

Contoh :

> hitungBensin 11 11

14


> hitungBensin 1 10

67

Keterangan:

Pada contoh kedua, 67 didapatkan dengan menjumlahkan bensin yang diperlukan untuk mobil yang mulai pada posisi ke-1, posisi ke-2, posisi ke-3, hingga posisi ke-10

Haskell ▾

 [HitungBensin.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.01 sec, 3.39 MB
2	10	Accepted	0.00 sec, 3.00 MB
3	10	Accepted	0.00 sec, 2.96 MB

No	Score	Verdict	Description
4	10	Accepted	0.09 sec, 4.27 MB
5	10	Accepted	0.08 sec, 4.40 MB
6	10	Accepted	0.00 sec, 2.96 MB
7	10	Accepted	0.00 sec, 2.97 MB
8	10	Accepted	0.08 sec, 4.23 MB
9	10	Accepted	0.00 sec, 2.88 MB
10	10	Accepted	0.00 sec, 3.32 MB

◀ [Praktikum 1 - Shift 1](#)

Jump to...

⌵