

Started on	Friday, 4 February 2022, 1:16 PM
State	Finished
Completed on	Friday, 4 February 2022, 3:16 PM
Time taken	2 hours
Grade	560.00 out of 700.00 (80%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

**Nama File:** PersamaanKuadrat.hs

**Header:** module PersamaanKuadrat where

Buatlah definisi, spesifikasi, dan realisasi fungsi **persamaanKuadrat** yang menerima 3 buah bilangan integer (a,b,c) dan sebuah nilai integer x, menghasilkan nilai persamaan kuadrat yang dibentuk oleh ketiga bilangan tersebut untuk nilai x yang diberikan dengan rumus:  $ax^2 + bx + c$

Contoh aplikasi fungsi dan hasilnya:

```
> persamaanKuadrat 1 2 1 (-1)
0
```

Haskell

 [PersamaanKuadrat.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 2.99 MB
2	20	Accepted	0.00 sec, 2.95 MB
3	20	Accepted	0.00 sec, 2.95 MB
4	20	Accepted	0.00 sec, 2.89 MB
5	20	Accepted	0.00 sec, 2.88 MB

Question **2**  
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, 3.08 MB
2	10	Accepted	0.00 sec, 3.06 MB
3	10	Accepted	0.00 sec, 3.23 MB

No	Score	Verdict	Description
----	-------	---------	-------------

4	10	Accepted	0.01 sec, 2.88 MB
5	10	Accepted	0.00 sec, 3.04 MB
6	10	Accepted	0.00 sec, 3.27 MB
7	10	Accepted	0.00 sec, 3.13 MB
8	10	Accepted	0.00 sec, 3.12 MB
9	10	Accepted	0.00 sec, 3.24 MB
10	10	Accepted	0.00 sec, 3.23 MB

Question **3**

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, 2.92 MB
2	6	Accepted	0.00 sec, 2.84 MB
3	6	Accepted	0.00 sec, 2.92 MB
4	6	Accepted	0.00 sec, 2.86 MB
5	6	Accepted	0.00 sec, 2.94 MB
6	6	Accepted	0.03 sec, 2.97 MB
7	6	Accepted	0.00 sec, 2.85 MB
8	6	Accepted	0.01 sec, 2.92 MB

No	Score	Verdict	Description
			0.00 sec, 2.93 MB

10	6	Accepted	0.00 sec, 2.82 MB
11	6	Accepted	0.02 sec, 2.84 MB
12	6	Accepted	0.00 sec, 2.92 MB
13	6	Accepted	0.00 sec, 2.93 MB
14	6	Accepted	0.00 sec, 2.94 MB
15	6	Accepted	0.00 sec, 2.85 MB
16	10	Accepted	0.00 sec, 2.95 MB

Question **4**  
Partially correct  
Mark 60.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

**Nama File:** JamTerlambat.hs

**Header:** module JamTerlambat where

Yukirin adalah penyanyi terkenal yang merupakan *center* dari *idol group* ITB48. Minggu depan, Yukirin akan mengadakan konser di Gedung Sanasa Buyada Gashena di Inaugurasi Terang Bulan. Konser akan mulai tepat pada pukul 08.30. Di Inaugurasi Terang Bulan, Anda dikatakan terlambat apabila datang lewat dari waktu mulai acara. Sementara bila Anda datang tepat waktu atau lebih awal, Anda tidak dikatakan terlambat.

Penonton konser tentunya memiliki tingkat kekecewaan tertentu. Untuk satu detik setiap Yukirin terlambat, tingkat kekecewaan bertambah 10 poin. Namun apabila Yukirin tidak terlambat, penonton tidak akan kecewa.

Tuliskan fungsi **jamTerlambat** yang menerima input tiga buah bilangan integer yang merepresentasikan jam (0..23), menit (0..59), detik (0..59). Diberikan waktu mulai konser 08.30. Fungsi tersebut mengembalikan tuple yang terdiri atas :

- Selisih antara waktu input dengan waktu mulai konser berupa nilai selisih dalam jam, menit, dan detik (3 integer).  
Petunjuk: ubah masukan jam, menit, dan detik ke jumlah detik; cari selisih dalam detik; kembalikan selisih ke jam, menit, detik.
- Terlambat atau tidak (true: terlambat, false: tidak terlambat)
- Tingkat kekecewaan penonton, jumlah selisih detik keterlambatan dikali 10 jika terlambat. 0 jika tidak terlambat.

*Asumsi: masukan selalu berupa integer positif dan pasti valid merepresentasikan jam sesuai definisi di atas.*

Contoh aplikasi fungsi dan hasilnya:

```
> jamTerlambat 7 55 0
(0,35,0,False,0)
> jamTerlambat 8 45 20
(0,15,20,True,9200)
> jamTerlambat 10 0 01
(1,30,1,True,54010)
```

Haskell

 [JamTerlambat.hs](#)

Score: 60

Blackbox

Score: 60

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 2.84 MB
2	20	Accepted	0.00 sec, 2.96 MB
3	0	Wrong answer	0.00 sec, 2.97 MB
4	20	Accepted	0.00 sec, 2.95 MB
5	0	Wrong answer	0.00 sec, 2.95 MB

Question **5**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

**Nama File:** SumOfDigits.hs

**Header:** module SumOfDigits where

Buatlah definisi, spesifikasi, dan realisasi dari fungsi **sumOfDigits**, yang merupakan fungsi menghitung hasil penjumlahan dari setiap bilangan tunggal yang terdapat di dalam sebuah bilangan integer positif atau 0.

Misalnya:

sumOfDigits(234) = 2 + 3 + 4 = 9

sumOfDigits(38) = 3 + 8 = 11

sumOfDigits(5) = 5

Fungsi ini **harus** diselesaikan menggunakan pendekatan rekursif.

```
sumOfDigits :: Int -> Int
-- sumOfDigits n menghasilkan penjumlahan setiap bilangan tunggal yang membentuk n
-- prekondisi n >= 0
```

Contoh aplikasi fungsi dan hasilnya:

No	Aplikasi	Hasil	Keterangan
1.	sumOfDigits 123	6	1+2+3 = 6
2.	sumOfDigits 234	9	2+3+4 = 9
3.	sumOfDigits 1234	10	1+2+3+4 = 10

Haskell

 [SumOfDigits.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, 2.83 MB
3	10	Accepted	0.00 sec, 2.95 MB
4	10	Accepted	0.00 sec, 2.94 MB
5	10	Accepted	0.07 sec, 2.96 MB
6	10	Accepted	0.00 sec, 2.95 MB
7	10	Accepted	0.00 sec, 2.96 MB
8	10	Accepted	0.00 sec, 2.84 MB
9	10	Accepted	0.00 sec, 2.96 MB

No	Score	Verdict	Description
10	10	Accepted	0.00 sec, 2.93 MB

Question **6**  
Correct  
Mark 100.00 out of 100.00

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, 2.96 MB
2	20	Accepted	0.00 sec, 2.86 MB
3	20	Accepted	0.00 sec, 2.85 MB
4	20	Accepted	0.00 sec, 2.95 MB
5	20	Accepted	0.00 sec, 2.92 MB



Question **7**  
Not answered  
Marked out of 100.00

Time limit	1 s
Memory limit	64 MB

[2021-2] 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 <= 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

◀ QnA Praktikum 1

Jump to...