

Started on Friday, 4 February 2022, 10:59 AM

State Finished

Completed on Friday, 4 February 2022, 12:58 PM

Time taken 1 hour 59 mins

Grade 574.00 out of 700.00 (82%)

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, 3.01 MB
2	20	Accepted	0.00 sec, 3.00 MB
3	20	Accepted	0.00 sec, 2.89 MB
4	20	Accepted	0.00 sec, 3.02 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	$25^{\circ}\text{C} = 4/5 * 25^{\circ}\text{R}$ (=) $25^{\circ}\text{C} = 20^{\circ}\text{R}$
2.	konversiSuhu 37 'F'	98.6	$37^{\circ}\text{C} = (9/5 * 37^{\circ}\text{F}) + 32^{\circ}\text{F}$ (=) $37^{\circ}\text{C} = 98.6^{\circ}\text{F}$
3.	konversiSuhu (-30) 'K'	243.15	$-30^{\circ}\text{C} = -30\text{K} + 273.15\text{K}$ (=) $0^{\circ}\text{C} = 243.15\text{K}$

Haskell

 [KonversiSuhu.hs](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 3.10 MB
2	10	Accepted	0.00 sec, 2.95 MB
3	10	Accepted	0.02 sec, 3.07 MB

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

4	10	Accepted	0.00 sec, 2.88 MB
5	10	Accepted	0.02 sec, 3.11 MB
6	10	Accepted	0.00 sec, 2.96 MB
7	10	Accepted	0.00 sec, 3.10 MB
8	10	Accepted	0.00 sec, 3.13 MB
9	10	Accepted	0.00 sec, 3.12 MB
10	10	Accepted	0.00 sec, 3.13 MB

Question **3**
Partially correct
Mark 94.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File : SeleksiKerja.hs

Nama Modul : SeleksiKerja

Sebuah perusahaan ingin melakukan penerimaan karyawan baru. Terdapat 4 lowongan kerja yang dibuka oleh perusahaan tersebut, yaitu pekerjaan A, B, C, dan D. Untuk melamar pada perusahaan tersebut, dibutuhkan pengalaman managerial dan pengalaman software engineer. Ketentuan untuk melamar pekerjaan adalah sebagai berikut.

1. Jika pelamar memiliki pengalaman managerial ≥ 2 tahun dan memiliki pengalaman menjadi software engineer ≥ 4 tahun, maka ybs bisa mendaftar untuk semua pekerjaan.
2. Jika pelamar memiliki kemampuan managerial < 2 tahun namun memiliki pengalaman menjadi software engineer ≥ 4 tahun, maka ybs bisa mendaftar untuk pekerjaan B. Selain itu, jika pengalaman menjadi software engineer < 4 tahun, maka ybs hanya dapat mendaftar untuk pekerjaan C.
3. Jika pengalaman menjadi software engineer < 4 tahun namun memiliki pengalaman managerial ≥ 2 tahun, maka ybs dapat mendaftar untuk pekerjaan D.
4. Semua pelamar dapat melamar untuk pekerjaan C.

Buatlah sebuah fungsi **seleksi** yang menerima input 2 buah integer, misalnya m, s, dan p, dengan m mewakili lama pengalaman managerial seseorang (dalam tahun, bernilai ≥ 0), s mewakili lama pengalaman menjadi software engineer seseorang (dalam tahun, bernilai ≥ 0), dan p mewakili jenis pekerjaan yang diinginkan (bernilai A..D).

Fungsi akan mengembalikan nilai True jika pelamar dengan pengalaman managerial m dan pengalaman menjadi software engineer s melamar untuk pekerjaan p. Jika kondisi tidak terpenuhi, maka akan dikembalikan nilai False.

Contoh:

```
> seleksi 0 1 'A'
```

```
False
```

```
> seleksi 0 0 'C'
```

```
True
```

```
> seleksi 2 5 'B'
```

```
True
```

Haskell

 [SeleksiKerja.hs](#)

Score: 94

Blackbox

Score: 94

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	6	Accepted	0.00 sec, 2.84 MB
2	6	Accepted	0.00 sec, 2.96 MB
3	0	Wrong answer	0.00 sec, 2.83 MB
4	6	Accepted	0.00 sec, 2.93 MB
5	6	Accepted	0.00 sec, 2.96 MB
6	6	Accepted	0.00 sec, 2.86 MB
7	6	Accepted	0.00 sec, 2.85 MB
8	6	Accepted	0.00 sec, 2.92 MB

No	Score	Verdict	Description
	8	Accepted	0.00 sec, 2.83 MB

10	6	Accepted	0.00 sec, 2.92 MB
11	6	Accepted	0.00 sec, 2.91 MB
12	6	Accepted	0.00 sec, 2.94 MB
13	6	Accepted	0.00 sec, 2.94 MB
14	6	Accepted	0.00 sec, 2.95 MB
15	6	Accepted	0.00 sec, 2.92 MB
16	10	Accepted	0.00 sec, 2.82 MB

Question **4**

Correct

Mark 100.00 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: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 3.00 MB
2	20	Accepted	0.00 sec, 3.01 MB
3	20	Accepted	0.00 sec, 2.90 MB
4	20	Accepted	0.00 sec, 2.91 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: 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:

$\text{sumOfDigits}(234) = 2 + 3 + 4 = 9$

$\text{sumOfDigits}(38) = 3 + 8 = 11$

$\text{sumOfDigits}(5) = 5$

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, 2.93 MB
2	10	Accepted	0.00 sec, 2.82 MB
3	10	Accepted	0.00 sec, 2.92 MB
4	10	Accepted	0.00 sec, 2.94 MB
5	10	Accepted	0.00 sec, 2.81 MB
6	10	Accepted	0.00 sec, 2.93 MB
7	10	Accepted	0.00 sec, 2.85 MB
8	10	Accepted	0.00 sec, 2.84 MB
9	10	Accepted	0.00 sec, 2.96 MB

No	Score	Verdict	Description
10	40	Accepted	0.00 sec, 2.81 MB

Question **6**
Partially correct
Mark 80.00 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 $m \leq n$ dan $x \leq n$.

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: 80

Blackbox

Score: 80

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 2.82 MB
2	20	Accepted	0.00 sec, 2.97 MB
3	20	Accepted	0.00 sec, 2.98 MB
4	20	Accepted	0.00 sec, 2.96 MB
5	0	Wrong answer	0.00 sec, 2.97 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 $(11 \cdot 3 + 1) = 34$. 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

◀ QnA Praktikum 1

Jump to...