

Instituto Tecnológico de Veracruz
Programación lógica y funcional. Practica 1.
Programación NO recursiva en Haskell
19/Febrero/2019. Versión 4

Las siguientes funciones deberán ser entregadas individualmente por los estudiantes conforme a los criterios publicados en la plataforma.

NOTAS:

- Como ya saben en la plataforma se publica para su consulta, un método para la creación de programas de computadora bajo el enfoque del paradigma imperativo, sin embargo como ya se explicó en clase, podemos apoyarnos en ella para crear programas usando el enfoque funcional. Les sugiero usar dicho método para realizar los siguientes ejercicios.
- LOS NOMBRES DE LAS FUNCIONES Y PARAMETROS DEBERAN ESTAR EN CONGRUENCIA CON LO ACORDADO POR EL GRUPO Y PUBLICADO EN LA PLATAFORMA DEL CURSO. NO OLVIDE PONER ENFASIS, SI ES POSIBLE, EN DIVIDIR EL PROBLEMA EN PARTES Y CREAR UNA FUNCIÓN PARA SOLUCIONAR CADA PARTE DEL PROBLEMA

=====

=====PROBLEMAS=====

1. Cree una función que reciba como parámetros a longitud y anchura de un rectángulo. La función dará como salida la anchura y longitud del rectángulo además del área, únicamente si el perímetro del rectángulo es mayor que 12. De lo contrario la función dará como salida la frase: "PERIMETRO MENOR O IGUAL A 12", utilice la función predeterminada de Haskell SHOW
2. Construya una función que simplifique un quebrado. El quebrado estará representado por una tupla de dos elementos, el primer elemento es el numerador y el segundo es el denominador. Investigue sobre la función predeterminada de Haskell gcd y úsela en la construcción de la función. Ejemplo:

simplificaQ (4,10) → (2,5)

3. Construya una función que divida dos quebrados. Cada quebrado estará representado por una tupla de dos elementos, el primer elemento es el numerador y el segundo es el denominador. El resultado de la función deberá estar simplificado use la función anterior para lograrlo. Use más de una función.
4. Determine los parámetros de entrada y cree una función de calcule la pendiente y la intercepción con el eje de las ordenadas de una ecuación lineal de la forma $Ax + By + C = 0$. Solo si la pendiente y/o la intercepción con el eje de las ordenadas NO son

Instituto Tecnológico de Veracruz
Programación lógica y funcional. Practica 1.
Programación NO recursiva en Haskell
19/Febrero/2019. Versión 4

enteros exprese el resultado en forma de quebrado. Utilice la función predeterminada de Haskell SHOW. Analice si es posible el uso de MAS DE UNA función.

5. Cree una función que devuelva como salida la ecuación de la línea recta que pasa los puntos A y B los cuales serán dados como parámetro (use tuplas de 2 elementos). Utilice la función SHOW, use más de una función.
6. Realiza una función para resolver el siguiente tipo de problema:

Si la razón en que el punto P divide al segmento P_1P_2 se define como $P_1P/PP_2=-2$ y las coordenadas de sus extremos son $P_1(7,4)$ y $P_2(-2,1)$. ¿Cuáles son las coordenadas del punto P?

Use tuplas de 2 elementos para pasar como parámetro los puntos P_1 y P_2 . También use una tupla de 2 elementos para pasar como parámetro la razón dada ya que, de esta manera, la razón podrá ser un entero o una fracción. Para expresar la salida puede usar una tupla o lo que usted determine. Analice si es posible el uso de MAS DE UNA función y el uso de la función SHOW. Consulte el tema: puntos que dividen segmentos en una razón dada

7. Elabore una función que determine si tres puntos del plano cartesiano forman una línea recta. Investigue lo necesario. Nota: Cada punto estará representado por una tupla de dos elementos, el primer elemento es la abscisa y el segundo es el la ordenada. Analice si es posible el uso de MAS DE UNA función Ejemplo:

palineados (2,1) (4,2) (6,3) → forman una línea recta
palineados (2,5) (4,2) (6,6) → No forman una línea recta

8. **En una empresa manufacturera el jornal de cada obrero se calcula de la siguiente manera:**

Salario neto = Salario Bruto – Deducciones

Determine los parámetros y realice una función que calcule el salario neto de un empleado, de acuerdo con las siguientes definiciones:

Salario Bruto = Salario x hrs trabajadas + Cuota x exceso de unidades producidas + Ayuda de despesa

Salario por horas trabajadas es igual al # de horas trabajadas X la cuota por hora. Si el número de horas trabajadas supera las 40, las horas excedentes serán consideradas horas

Instituto Tecnológico de Veracruz
Programación lógica y funcional. Practica 1.
Programación NO recursiva en Haskell
19/Febrero/2019. Versión 4

extra. Cada hora extra tiene un costo de $2\frac{1}{3}$ veces la cuota por hora. Después de la octava hora extra, el costo de la hora extra es $3\frac{2}{3}$ veces la cuota por hora.

Cuota x exceso de unidades producidas es igual al 20 % del salario por horas trabajadas siempre que se haya excedido en un 10% el mínimo de unidades producidas a la semana

Ayuda de despensa es igual 3% del salario por horas trabajadas multiplicado por el número de hijos

Deducciones

FICA = 5.2% del salario bruto

Retención

Si el empleado tiene más de un hijo

Retención = (salario por horas trabajadas/5) – 40 x (número de hijos-2),

Si el empleado tiene un hijo o ninguno la retención será:

Retención = salario por horas trabajadas / 5

Determine los parámetros de entrada, cree la función y use SHOW

9. Una librería estima los precios de sus libros de la siguiente forma: El precio básico de un libro es de \$5.00 más \$0.20 por página. Sin embargo, si el número de páginas excede de 300, el precio sufrirá un recargo adicional de \$10.00 y 0.75 por página excedente. Además, si el número de páginas excede de 550 el precio se incrementará en otros \$8.50 y \$1.00 por página excedente. Determine cuáles serán los parámetros de la función y cree una función que realice el cálculo descrito. Use mas de una función.

10. Investigue como usar la cláusula LET del lenguaje Haskell. Construya una función que resuelva el problema anterior (inciso 9) usando la cláusula (LET).

11. Definir la función mayor tal que mayor x y w devuelva como resultado el mayor número de tres cifras que puede construirse con los dígitos x, y, w Por ejemplo,

mayor 251 → 521

mayor 125 → 521

mayor 512 → 521

Instituto Tecnológico de Veracruz
Programación lógica y funcional. Practica 1.
Programación NO recursiva en Haskell
19/Febrero/2019. Versión 4

Restricción: los parámetros de entrada deben de ser diferentes.

NOTA: Use: guardas en combinación con if <condición> then ---- else -----

12. Construya una función que convierta los segundos dados a horas, minutos y segundos.

Ejemplo:

```
LuisRiveraLozano> segAhr 9800
"2 Horas 43 minutos 20 segundos"
```

Notas: 1) Utilice 4 funciones definidas por el usuario 2) Use las funciones predeterminadas de haskell div, mod y show

13. Investigue como usar la cláusula WHERE del lenguaje Haskell. Construya una función que resuelva el problema anterior (inciso 12) usando la cláusula mencionada (WHERE). Utilice los mismos datos de prueba.