Especificación matemática ejercicios 1-30

José Simón Ramos Sandoval

Universidad Nacional de Colombia, Programación de computadores 2021-1s

1. La granja

En una granja se crían un número de V - Vacas, A - Aves (pollos y gallinas) y E - escorpiones. Las vacas están encerradas en un corral de N × M metros cuadrados, las aves en un galpón y los escorpiones en vitrinas.

Ejercicio 1

Si una vaca necesita M metros cuadrados de pasto para producir X litros de leche, ¿cuántos litros de leche se producen en la granja?.

1. Análisis y especificación del problema
1.1 Objetos conocidos:

- El número de vacas ($V \in \mathbb{N}$)
- Ancho del corral medido en metros ($N \in \mathbb{R}^+$)
- Largo del corral medido en metros ($M \in \mathbb{R}^+$)
- Cantidad de metros cuadrados de pasto necesarios para producir 1 litro de leche ($P \in \mathbb{R}^+$)

1.2 Objetos desconocidos:

- Cantidad de litros de leche que se producen en la granja ($X \in \mathbb{R}^+$)

1.3 Relaciones entre objetos:

- la multiplicacion entre las medidas de largo y ancho del corral dan los metros cuadrados del corral. (N * M)
- De la division entre este valor y la cantidad de metros cuadrados de pasto necesarios para producir un litro de leche se obtiene la cantidad de litros que se pueden producir por vaca. ($\frac{N*M}{P}$)

- La multiplicacion de este valor por la cantidad de vacas da la cantidad de leche producida

2. Diseño y prueba conceptual

Entradas: 4 variables: (V$\in \mathbb{N}$, N $\in \mathbb{R}^+$, M $\in \mathbb{R}^+$, P $\in \mathbb{R}^+$: P ,V, N, M $\neq 0$)

Salidas: Un valor: X $\in \mathbb{R}^+$

Relación:

Litros_producidos: ($\mathbb{N} \times \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$)

litros_producidos$(V, N, M, P) \to \frac{N*M}{P} *V$

3. Codificación

El código en el que se resuelve el ejercicio se encuentra en el archivo «Ejercicio_1_SimonRamos.py»

Ejercicio 2

Si 1/3 de las aves que hay en la granja son gallinas, y la mitad de las gallinas ponen 1 huevo cada 3 días y la otra mitad 1 huevo cada 5 díıas, ¿en un mes cuáantos huevos producen? (1 mes = 30 días).

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Numero de aves (A$\in \mathbb{N}$)
- Numero de gallinas (G) es un tercio de la cantidad de aves ($\frac{A}{3}$)
- La mitad de las gallinas (G_1) ponen un huevo cada 3 dias y la otra mitad (G_2) cada 5
- Un mes tiene 30 días

1.2 Objetos desconocidos:

- Numero de gallinas, que son un tercio del numero de las aves (G $\in \mathbb{N}$)
- Numero de huevos puestos en un mes (H $\in \mathbb{N}$)

1.3 Relación entre objetos:

- Cada gallina del G_1produce 10 huevos al mes ($\frac{30}{3}$), cada gallina del G_2 produce 6 huevos al mes ($\frac{30}{5}$)
- El total de huevos producidos es la multiplicacion de gallinas del G_1por los huevos que estas producen al mes + el producto de la cantidad de gallinas del G_2por la cantidad de huevos que producen al mes

2. Diseño y prueba conceptual

Entradas: Una variable ($A \in \mathbb{N} : A > 3 \wedge A \bmod 6 = 0$)

Salidas: Numero de huevos puestos en un mes $H \in \mathbb{N}$

Relación:

$G = \frac{A}{3}$

$G_1 = \frac{G}{2} \rightarrow G_1 = \frac{\frac{A}{3}}{2} \rightarrow G_1 = \frac{A}{6}$; $G_2 = \frac{G}{2} \rightarrow G_2 = \frac{\frac{A}{3}}{2} \rightarrow G_2 = \frac{A}{6}$

$H = G_1 * 10 + G_2 * 6 \rightarrow H = \frac{A}{6} * 10 + \frac{A}{6} * 6 \rightarrow H = \frac{16A}{6} \rightarrow H = \frac{8A}{3}$

Función de la cantidad de huevos:

Numero_huevos($\mathbb{N}$) $\rightarrow \mathbb{N}$

Numero_huevos(A) $\rightarrow \frac{8A}{3}$

3. Codificación

El código en el que se resuelve el ejercicio se encuentra en el archivo «Ejercicio_2_SimonRamos.py»

Ejerccio 3

Si los escorpiones de la granja se venden a China, y hay escorpiones de tres diferentes tamaños: pequeños (con un peso de 20 gramos), medianos (con un peso 30 gramos) y grandes (con un peso de 50 gramos), ¿cuántos kilos de escorpiones se pueden vender sin que decrezca la población a menos de 2/3?.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Hay escorpiones grandes (50 gramos) $G \in \mathbb{N}$
- Hay escorpiones medianos (30 gramos) $M \in \mathbb{N}$
- Hay escorpiones pequeños (20 gramos) $P \in \mathbb{N}$
- Solo se puede vender 1/3 del total de la población

1.2 Objetos desconocidos:

- Cantidad de kilos $X \in \mathbb{N}$ que se pueden vender sin que la población decrezca a menos de 2/3

1.3 Relación entre los objetos:

- La población total de escorpiones T es igual a $G + M + P$
- La cantidad de escorpiones que se pueden vender sin que decrezca la población a menos de dos tercios es $\llbracket \frac{T}{3} \rrbracket$

- Primero se venderán los escorpiones grandes hasta que se complete la cantidad permitida para vender, después se venderán los medianos y por último los pequeños

2. Diseño y prueba conceptual

Población total $T = G + M + P$

La cantidad de escorpiones que se pueden vender es $Venta = [\![\frac{T}{3}]\!] \to Venta = [\![\frac{G+M+P}{3}]\!]$

Escorpiones que se pueden vender:

Entradas: 3 variables: Cantidad de escorpiones grandes $G \in \mathbb{N}$, cantidad de escorpiones medianos $M \in \mathbb{N}$, cantidad de escorpiones pequeños $P \in \mathbb{N}$

Salidas: Una variable: Cantidad de escorpiones que se pueden vender $Y \in \mathbb{N}$

Relaciones:

$Venta : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

$Venta(G, M, P) \to [\![\frac{G+M+P}{3}]\!]$

La cantidad de kilos vendidos (Kilos_ Vendidos):

Entradas: 4 variables: Cantidad de escorpiones grandes $G \in \mathbb{N}$, cantidad de escorpiones medianos $M \in \mathbb{N}$, cantidad de escorpiones pequeños $P \in \mathbb{N}$, cantidad de escorpiones que se pueden vender $Y \in \mathbb{N}$

Salidas: Una variable: Cantidad de kilos que se pueden vender $X \in \mathbb{N}$

Relaciones:

$Kilos\_Vendidos : \mathbb{N}x\mathbb{N}x\mathbb{N}x\mathbb{N} \to \mathbb{R}$

$$(Y, G, M, P) \to \begin{cases} \frac{(Y*50)}{1000}, & si \quad G > Y \\ \frac{G*20+Y*30}{1000}, & si \quad G + M > Y \\ \frac{G*30+M*10+Y*20}{1000} \quad , & En\ otro\ caso \end{cases}$$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo «Ejercicio_3_Simon-Ramos.py».

Ejercicio 4

Al granjero se le daño el corral y no sabe si volver a cercar el corral con madera, alambre de púas o poner reja de metal. Si va a cercar con madera debe poner 4 hileras de tablas, con varilla 8 hileras y con alambre solo 5 hileras,él quiere saber que es lo menos costoso para cercar si sabe que el alambre de púas vale P por metro, las tablas a Q por metro y las varillas S por metro. Dado el tamaño del corral y los precios de los elementos, ¿cuál cerramiento es el más económico?.

1. Análisis y espeficicación del problema

1.1 Objetos conocidos:

- materiales a utilizar: madera, alambre y varillas

- Cantidad de material que necesita por metro: 4 hileras si son tablas, 8 hileras si son varillas y 5 hileras si es alambre
- Precio del material por metro: alambre de púas vale P (P $\in \mathbb{R}^+$), tablas vale Q (Q $\in \mathbb{R}^+$) y las varillas S (S $\in \mathbb{R}^+$)

1.2 Objetos desconocidos:

- Cual es el material mas economico

1.3 Relación entre los objetos:

- El coste total de cada material viene dado por la multiplicación de su precio por la cantidad necesaria (madera = 4Q, alambre = 5P, varillas = 8S)
- El material más economico será el menor de la relación expuesta

2. Diseño y prueba conceptual

Precio de la madera:

Entradas: 1 variable: Precio por hilera de tabla (Q $\in \mathbb{R}^+$: Q$\neq$ 0)

Salidas: 1 variable: Precio total de la madera

Relación:

$precio\_madera : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$precio\_madera(Q) \rightarrow 4 * Q$

Precio del alambre:

Entradas: 1 variable: Precio por hilera de alambre (P $\in \mathbb{R}^+$: P $\neq$ 0)

Salidas: 1 variable: Precio total del alambre

Relación:

$precio\_alambre : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$precio\_alambre(P) \rightarrow 5 * P$

Precio de la varilla:

Entradas: 1 variable: Precio por hilera de la varilla (S $\in \mathbb{R}^+$: S $\neq$ 0)

Salidas: 1 variable: Precio total de la varilla

Relación:

$precio\_varilla : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$precio\_varilla(S) \rightarrow 8 * S$

Material más económico:

Entradas: 3 variables: precio total de la madera (madera $\in \mathbb{R}^+$), precio total del alambre (alambre $\in \mathbb{R}^+$), precio total de las varillas. (varillas $\in \mathbb{R}^+$)

Salida: 1 cadena de caracteres (Material más barato)

Relación:

Material_económico: $\mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{ASCII}^*$

$$(madera, alambre, varilla) \to \begin{cases} madera & , \quad si \quad alambre \geq madera \leq varilla \\ alambre & , \quad si \quad madera \geq alambre \leq varilla \\ varilla & , \quad \text{En otro caso} \end{cases}$$

3. Codificación

El código en donde se resuelve el ejercicio se encuentra en el archivo «Ejercicio_4_SimonRamos.py».

2. Numéricos

Ejercicio 5

Función potencia de un entero elevado a un entero.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Numero entero de la base (B $\in \mathbb{Z}$)
- Numero entero del expeonente (E $\in \mathbb{Z}$)

1.2 Objetos desconocidos:

- Resultado de la potencia (X $\in \mathbb{R}$)

1.3 Relación de los objetos:

- El numero B será multiplicado por si mismo tantas veces lo indica E (B^E= X)
- Si E es un número negativo, entonces el resultado será ($\frac{1}{B^{-E}}$)
- Si E = 0, el resultado será 1. Sin embargo, si E $\leq$ 0 B debe ser $\neq$ 0, ya que de lo contrario sería una indeterminación

2. Diseño y prueba conceptual

Entradas: Dos variables (B$\in \mathbb{Z}$y E $\in \mathbb{Z}$: si E $\leq$ 0 entonces B $\neq$ 0)

Salidas: Una variable (X$\in \mathbb{R}$) como resultado de la potencia

Relación:

$potencia : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R}$

$$\text{potencia } (\mathbf{B{,}E}) \rightarrow \begin{cases} \frac{1}{B^E} & si \quad E < 0 \\ B^E & si \quad E > 0 \\ 1 & , \quad en\, otro\, caso \end{cases}$$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_5_SimonRamos.py»

Ejercicio 6

Una función que determine si un número es divisible por otro.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Dos números reales: X $\in \mathbb{R}$y Y $\in \mathbb{R}$

1.2 Objetos desconocidos:

- Se desconoce si X es divisible por Y

1.3 Realción de los objetos:

- Si X es divisible por Y entonces el módulo da 0

2 Diseño y prueba conceptual

Entradas: Dos variables (X $\in \mathbb{R}$y Y $\in \mathbb{R}$)

Salidas: Valor de verdad: True si X es divisible por Y o False si X no es divisible por Y

Relación: Si X mod Y = 0 $\rightarrow$True, Si X mod Y $\neq$ 0 $\rightarrow$ False

$divisible : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B}$

$$divible(X, Y) \rightarrow \begin{cases} False, & si \quad Y = 0 \\ False, & si \quad XmodY \neq 0 \\ True & \text{En otro caso} \end{cases}$$

3. Codificación:

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_6_SimonRamos.py»

Ejercicio 7

Determinar si un número es primo.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Un número X $\in \mathbb{N}$
- Si X solo es divisible por 1 y él mismo es primo

1.2 Objetos desconocidos:

- Se desconoce si el numero dado es primo

1.3 Relación entre los objetos:

- Un número es primo si solo es divisible por 1 y si mismo. Además, el 2 es el único número par primo

2. Diseño y prueba conceptual

Entradas: Una variable (X $\in \mathbb{N}$: X $\neq$0)

Salidas: Un valor de verdad: True si X es un número primo, False si X no es un número primo

Relación:

$primo : \mathbb{N} \rightarrow \mathbb{B}$

$$primo(X) \rightarrow \begin{cases} False, & si \quad x = 1 \\ False & si \quad \forall_{n=2}^{x-1} x \; mod \, n = 0 \\ True, & \text{En otro caso} \end{cases}$$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo «Ejercicio_7_Simon-Ramos.py»

Ejercicio 8

Dados dos naturales, determinar si son primos relativos.

1. Análisis y especificación del problema:

1.1 Objetos conocidos:

- Dos números naturales X $\in \mathbb{N}$ y Y $\in \mathbb{N}$
- Dos numeros son primos relativos si no comparten divisores

1.2 Objetos desconocidos:

- Se desconoce si son primos relativos (True) o no lo son (False)

1.3 Relación entre los objetos:

- Si un número divide a X y también a Y no serán primos relativos (False)

- Si un numero divide a X pero no a Y,o si un numero divide a Y pero no a X si serán primos relativos (True)

2. Diseño y prueba conceptual:

Entradas: Dos variables: (X $\in \mathbb{N}$ y Y $\in \mathbb{N}$: X ,Y$\neq$0)

Salidas: Un valor de verdad: True, si son primos relativos - False, si no son primos relativos

Relaciones:

$relativos : \mathbb{N} \times \mathbb{N} \to \mathbb{B}$

$$relativos(X,Y) \to \begin{cases} False, \quad si & \forall_{n=2}^{x} x \, mod \, n = 0 \wedge y \, mod \, n = 0 \\ True & En \, otro \, caso \end{cases}$$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_8_SimonRamos.py»

Ejercicio 9

Determinar si un número es múltiplo de la suma de otros dos números.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce un número X $\in \mathbb{R}$
- Se conoce una suma, que es el resultado de otros dos números reales (S =Y + Z, tal que Y, Z $\in \mathbb{R}$)

1.2 Objetos desconocidos:

- Se desconoce si X es un múltiplo de S (True) o si no lo es (False)

1.3 Relaciones:

- X es múltiplo de S si al multiplicar S por algún número natural n (n $\in\mathbb{N}$) obtenemos X
- matemáticamente, $n*$ S= X $\to$ n= $\frac{X}{S}$ $\to$ X mod S = 0
- Si S y X tienen signas diferentes no serán múltiplos

2. Diseño y prueba conceptual:

Suma de los dos digitos: ($suma \neq 0$)

Entradas: Dos números reales (Y , Z)

Salidas: Un número real S que corresponde a la suma de Y y Z, $S \neq 0$

$suma : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

$suma(Y,Z) \to Y + Z$

Detecta si son multiplos:

Entradas: Dos números reales (X, S: S $\neq$ 0)

Salidas: Un valor de verdad (True si X es múltiplo de S - False si X no es múltiplo de S)

$multiplo : \mathbb{R} \times \mathbb{R} \to \mathbb{B}$

$$multiplo(X, S) \to \begin{cases} False, & si \quad X \; mod \; S \neq 0 \\ False, & si \quad (X < 0 \wedge S > 0) \vee (S < 0 \wedge X > 0) \\ True, & En \; otro \; caso \end{cases}$$

3. Codificación:

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_9_SimonRamos.py»

Ejercicio 10

Dados los coeficientes de un polinomio de grado dos, evaluar el polinomio en un valor dado.

1. Análisis y especificación del problema

1,1 Objetos conocidos:

- Coeficientes del polinomio (a $\in \mathbb{R}$, b$\in \mathbb{R}$, c $\in \mathbb{R}$)
- Valor en el cual se evaluará el polinomio (x $\in \mathbb{R}$)

1.2 Objetos desconocidos:

- Imagen del polinomio al ser avaluado (y $\in \mathbb{R}$)

1.3 Relación entre los objetos:

- La forma de un polinomio es $ax^2 + bx - c$, por lo que la imagen resultará de reemplazar los respectivos valores en la frómula y operar
- $a \neq 0$, ya que de lo contrario el polinomio sería de grado 1

2. Diseño y prueba conceptual

Entradas: 4 variables (a $\in \mathbb{R}$, b $\in \mathbb{R}$, c $\in \mathbb{R}$,x$\in \mathbb{R}$: a $\neq$0)

Salidas: Una variable: (y $\in \mathbb{R}$)

Relaciones:

$polinomio : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

$polinomio(a, b, c, x) \to ax^2 + bx + c$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_10_SimonRamos.py»

Ejercicio 11

Dados los coeficientes de un polinomio de grado dos, calcular coeficiente lineal de la derivada.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Coeficientes de un polinomio de segundo grado (a $\in \mathbb{R}$, b $\in \mathbb{R}$, c $\in \mathbb{R}$)
- Forma de un polinomio de segundo grado $ax^2 + bx - c$
- Forma en que se calcula una derivada; f(x) = $x^n \rightarrow$ f '(x) $= n * x^{n-1}$; la derivada de una constante es 0

1.2 Objetos desconocidos:

- Se desconoce el coefieciente lineal de la derivada

1.3 Relación entre los objetos:

- El coeficiente lineal de la derivada se calcula a partir de los coeficientes dados y la forma en que se calcula la derivada
- a $\neq$0 o de lo contrario se convertiría en una polinomio de grado 1

2. Diseño y prueba conceptual

Entrada: Una variable (a $\in \mathbb{R}$: a $\neq$ 0)

Salidas: Una variable y $\in \mathbb{R}$

Relación:

$Derivada\_termino1 : \mathbb{R} \rightarrow \mathbb{R}$,

$Derivada\_termino1(a) \rightarrow 2a$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_11_SimonRamos.py»

Ejercicio 12

Dados los coeficientes de un polinomio de grado dos y un número real, evaluar la derivada del polinomio en ese número.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conocen los coeficiente del polinomio de segundo grado (a $\in \mathbb{R}$, b $\in \mathbb{R}$, c $\in \mathbb{R}$)
- Se conoce el valor en el que se evaluará la derivada (x $\in \mathbb{R}$)
- Forma de un polinomio de segundo grado $ax^2 + bx - c$
- Forma en que se calcula una derivada; f(x) = $x^n \rightarrow$ f '(x) =$n * x^{n-1}$; la derivada de una constante es 0

1.2 Objetos desconocidos:

- El resultado de evaluar la derivada en el valor de x (y$\in \mathbb{R}$)

1.3 Relación entre los objetos:

- Dado el polinomio $ax^2 + bx - c$ su derivada será $2ax + b$. El resultado de la evaluación se dará al reemplazar el valor de x
- Donde a $\neq$0 para que el polinomio sea de segundo grado

2. Diseño y prueba conceptual

Entradas: 3 variables (a $\in \mathbb{R}$, b $\in \mathbb{R}$, x$\in \mathbb{R}$: a $\neq$0)

Salidas: 1 variable y $\in \mathbb{R}$

Relación:

$derivada$:$\mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$;

$derivada'(a, b, x) \rightarrow 2ax + b$

3. Codificación:

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_12_SimonRamos.py»

Ejerciico 13:

Dado un natural, determinar si es un número de Fibonacci o no.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce un numero natural X $\in \mathbb{N}$
- La secuencia de fibonacci se da de la forma $F_n = F_{n-1} + F_{n-2}$

1.2 Objetos desconocidos:

- Se desconoce si X $\in$ Fibonacci

1.3 Relación entre los objetos:

- Si X $\in$ Fibonacci return True, en otro caso return False
- X $\in$ Fibonacci si su valor puede ser determinado con $F_n = F_{n-1} + F_{n-2}$

2. Diseño y prueba conceptual

Entradas: Una variable X $\in \mathbb{N}$

Salidas: Un valor de verdad (True $\rightarrow$X es un número de fibonacci, False$\rightarrow$ X no es un número de fibonacci)

Relación:

$fibo : \mathbb{N} \rightarrow \mathbb{B}$

$$fibo(X) \rightarrow \begin{cases} True. & si \quad X = F_n \text{tal que } \forall_{n=2}^{F_{n-1} \leq X \vee F_{n-2} \leq X} F_n = F_{n-1} + F_{n-2} \text{ Donde } F_1 = 0 \wedge F_2 = 1 \\ False, & En\ otro\ caso \end{cases}$$

3. Codificación:

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_13_SimonRamos.py»

3. Geométricos

Ejercicio 14

Dadas la pendiente y el punto de corte de dos rectas, determinar si son paralelas, perpendiculares o ninguna de las anteriores.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce la pendiente de dos rectas $M_1 \in \mathbb{R}$, $M_2 \in \mathbb{R}$
- Se conoce el punto de corte con el eje y de las dos rectas $B_1 \in \mathbb{R}$ y $B_2 \in \mathbb{R}$
- Se sabe que la ecuación de una recta es $y = mx + b$

1.2 Objetos desconocidos:

- Se desconoce si las rectas son perpendiculares, paralelas o ninguna de las dos

1.3 Relación entre los objetos:

- Si la pendiente de las dos rectas y sus cortes con el eje son iguales las dos rectas se tocarán en puntos infinitos $M_1 = M_2 \wedge B_1 = B_2$ $\rightarrow$No son paralelas ni perpendiculares
- Si la pendiente de las dos rectas es igual y sus cortes con el eje diferentes serán paralelas $M_1 = M_2 \wedge B_1 \neq B_2$ $\rightarrow$Son paralelas

- Si el resultado de la multiplicación de las dos pendientes es -1 son perpendiculares $M_1 * M_2 = -1$ →son perpendiculares

2. Diseño y prueba conceptual

Entradas: 4 variables ($M_1 \in \mathbb{R}, M_2 \in \mathbb{R}$,$B_1 \in \mathbb{R}$, $B_2 \in \mathbb{R}$: $M_1 \wedge B_1 \neq 0, M_2 \wedge B_2 \neq 0$)

Salidas: 3 cadenas de caracteres («son paralelas», «son perpendiculares», «no son ni paralelas ni perpendiculares»)

Relación:

$recta : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{ASCII}^*$

$$recta(M_1, M_2, B_1, B_2) \to \begin{cases} son\ paralelas, & si & M_1 = M_2 \wedge B_1 \neq B_2 \\ son\ perpendiculares, & si & M_1 * M_2 = -1 \\ no\ son\ ni\ paralelas\ ni\ perpendiculares, & & En\ otro\ caso \end{cases}$$

3. Codificacion

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_14_SImonRamos.py»

Ejercicio 15

Dadas la pendiente y el punto de corte de dos rectas, determinar los puntos de intersección.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce la pendiente de dos rectas $M_1 \in \mathbb{R}$, $M_2 \in \mathbb{R}$
- Se conoce el punto de corte con el eje y de las dos rectas $B_1 \in \mathbb{R}$ y $B_2 \in \mathbb{R}$
- Se sabe que la ecuación de una recta es $y = mx + b$

1.2 Objetos desconocidos:

- Se desconoce el punto de intersección entre las dos rectas

1.3 Relación entre los objetos:

- Las rectas no tendrán solución si $M_1 = M_2 \vee (M_1 = M_2 \wedge B_1 = B_2)$
- La intersección será aquellos valores donde X y Y sean solución al sistema de ecuaciones 2x2 generado de las dos rectas

2. Diseño y prueba conceptual

iguales:

Entradas: 4 variables $((M_1 \in \mathbb{R}, M_2 \in \mathbb{R}, B_1 \in \mathbb{R}, B_2 \in \mathbb{R}) : M1 \wedge B1 \neq 0, M2 \wedge B2 \neq 0$)

Salidas: Un valor de verdad (True→si son iguales, False→ si no son iguales)

Relación:

$iguales : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B}$

$$iguales(M1, M2, B1, B2) \rightarrow \begin{cases} True, & si \quad M1 = M2 \wedge B1 = B2 \\ False, & En\ otro\ caso \end{cases}$$

Función hay solución:

Entradas: 2 variables $(M_1 \in \mathbb{R}, M_2 \in \mathbb{R})$

Salidas: Un valor de verdad (True→si hay solución, False→ no hay solución)

Relación:

$haysolucion : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B}$

$$haysolucion(M1, M2) \rightarrow \begin{cases} False, & si \quad M1 = M2 \\ True, & En\ otro\ caso \end{cases}$$

Intersección:

Entradas: 4 variables $(M_1 \in \mathbb{R}, M_2 \in \mathbb{R}, B_1 \in \mathbb{R}, B_2 \in \mathbb{R})$

Salidas: 2 variables $(X \in \mathbb{R}, Y \in \mathbb{R})$

Relación:

$intersección : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \times \mathbb{R}$

$intersección(M1, M2, B1, B2) \rightarrow \frac{B_1 - B_2}{M_2 - M_1}, \frac{M_2 B_1 - M_1 B_2}{M_2 - M_1}$

Función que integra las anteriores:

Entradas: 4 variables $(M_1 \in \mathbb{R}, M_2 \in \mathbb{R}, B_1 \in \mathbb{R}, B_2 \in \mathbb{R})$

Salidas: 1 variable: Una cadena de caracteres ($sol \in \mathbb{ASCII}^*$)

Relación:

$final : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{ASCII}^*$

$$final(M1, M2, B1, B2) \rightarrow \begin{cases} \text{«Las rectas son las mismas e intersectan e infinitos puntos»} & si \quad iguales(M1, M2, B1, B2) = True \\ \text{«El punto de intersección es » + str}(intersecta(M1, M2, B1, B2)) & si \quad haysolucion(M1, M2) = True \\ \text{«No hay punto de intersección»} & \text{En otro caso} \end{cases}$$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_15_SimonRamos.py»

Ejercicio 16

Dado el radio de un círculo, calcular el área del triángulo que circunscribe el circulo (triangulo afuera).

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- El radio de un circulo (R$\in \mathbb{R}^+$:)
- El teorema del baricentro indica que $r = \frac{1}{3}h$, donde r es el radio de la circunferencia y h la altura del triangulo
- Un lado cualquiera de un triangulo equilatero se calcula así: $\frac{2h}{\sqrt{3}}$
- El área de un triangulo es $\frac{base*altura}{2}$

1.2 Objetos desconocidos:

- Se desconoce el área de un triangulo que circunscribe a una circunferencia

1.3 Relación entre los objetos:

- El área de un triangulo se calculo de la forma $\frac{base*altura}{2}$
- La altura de un triangulo que circunscribe a un circulo conociendo el radio (R) de la circunferencia se calcula: $altura = 3 * R$
- la base del triangulo, asumiendo que es un triangulo equilatero, se calcula como $\frac{2*altura}{\sqrt{3}} \rightarrow \frac{2*(3*R)}{\sqrt{3}} \rightarrow \frac{6*R}{\sqrt{3}}$
- Así que el área del triangulo será $\frac{\frac{6*R}{\sqrt{3}}*(3*R)}{2} \rightarrow \frac{\frac{18R^2}{\sqrt{3}}}{2} \rightarrow \frac{18R^2}{2\sqrt{3}}$

2. Diseño y prueba conceptual

Entradas: Una variable: Radio de la circunferencia ($R \in \mathbb{R}^+$)

Salidas: Una variable: Area ($Area \in \mathbb{R}^+ : Area \neq 0$)

Relación:

$Area\, triangulo : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$Area\, triangulo(R) \rightarrow \frac{18*R^2}{2\sqrt{3}}$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo «Ejercicio_16_Simon-Ramos.py»

Ejercicio 17

Dado el radio de un círculo, calcular el área y perímetro del cuadrado, pentágono y hexágono adentro (inscrito en un círculo) y afuera (inscribiendo al círculo).

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- El radio de un circulo (R$\in \mathbb{R}^+$; donde R $\neq 0$)
- El aéra de un cuadrado: S^2, perimetro de un cuadrado: $4 * S$
- El área de un pentagono: $\frac{perimetro * apotema}{2}$, perimetro de un pentagono: $5 * S$
- El área de un hexagono es: $6 * \frac{S^2\sqrt{3}}{4}$, perimetro de un hexagono: $6 * S$

1.2 Objetos desconocidos:

Inscritos al circulo:

- El area de un cuadrado inscrito al circulo ($ACI \in \mathbb{R}^+, ACI \neq 0$), El perimetro de un cuadrado inscrito al circulo($PCI \in \mathbb{R}^+, PCI \neq 0$),
- El area de un pentagono inscrito al circulo ($API \in \mathbb{R}^+, API \neq 0$), El perimetro de un pentagono inscrito al circulo($PPI \in \mathbb{R}^+, PPI \neq 0$),
- El area de un hexagono inscrito al circulo ($AHI \in \mathbb{R}^+, AHI \neq 0$), El perimetro de un hexagono inscrito al circulo($PHI \in \mathbb{R}^+, PHI \neq 0$)

Inscriben al circulo:

- el area de un cuadrado que inscribe al circulo ($ACF \in \mathbb{R}^+, ACF \neq 0$), el perimetro de un cuadrado que inscribe al circulo ($PCF \in \mathbb{R}^+, PCF \neq 0$)
- el area de un pentagono que inscribe al circulo ($APF \in \mathbb{R}^+, APF \neq 0$), el perimetro de un pentagono que inscribe al circulo ($PPF \in \mathbb{R}^+, PPF \neq 0$)
- el area de un hexagono que inscribe al circulo ($AHF \in \mathbb{R}^+, AHF \neq 0$), el perimetro de un hexagono que inscribe al circulo ($PHF \in \mathbb{R}^+, PHF \neq 0$)

1.3 Relación entre los objetos:

Cuadrado inscrito a la circunferencia:

- *Área:* La diagonal del cuadrado = D es $2 * R$, un lado S del cuadrado $S = \frac{D}{\sqrt{2}}$, el área de un cuadrado es S^2
  $S = \frac{2*R}{\sqrt{2}} \rightarrow ACI = (\frac{2*R}{\sqrt{2}})^2$
- *Perimetro:*El perimetro de un cuadrado es $4 * S \rightarrow$ $PCI = 4 * (\frac{2*R}{\sqrt{2}})$

Pentagono inscrito a la circunferencia:

- *Perimetro:* Después de cálculos algebraicos, el lado de un pentagono es $S = \frac{2R*(tan36)*\sqrt{(tan36)^2+1}}{(tan36)^2+1}$, el perimetro es $5 * S \rightarrow PPI = 5 * \frac{2R*(tan36)*\sqrt{(tan36)^2+1}}{(tan36)^2+1}$

- *Área:* Después de cálculos algebraicos, $Apotema = \frac{R\sqrt{(tan36)^2+1}}{(tan36)^2+1}$, el área $API = \frac{perimetro*apotema}{2} \rightarrow \frac{5R^2tan36}{(tan36^2)+1}$

Hexagono inscrito a la circunferencia:

- *Área:* Debido a que un hexagono puede dividirse en 6 triangulos equilateros, $S = R$, el área $AHI = 6 * \frac{R^2\sqrt{3}}{4}$

- *Perimetro:* el perimetro es $PHI = 6 * R$

Cuadrado que inscribe a un circulo:

- *Área:* Un lado del cuadrado es igual al diametro del circulo $S = 2R$, el área es $S^2 \rightarrow ACF = (2R)^2 \rightarrow ACF = 4R^2$

- *Perimetro:* El perimetro de un cuadrado es $4 * S \rightarrow 4 * (2R)$, el perimetro $PCF = 8R$

Pentagono que inscribe a un circulo

- *Perimetro:* El radio corresponde al apotema del pentagono, utilizando cálculos algebraicos se tiene que $S = 2R(tan36)$, el perimetro es $5 * S \rightarrow 5 * 2(R(tan36)) \rightarrow PPF = 10R(tan36)$

- *Área:* El área de un pentagono es $\frac{perimetro*apotema}{2}$, por lo que $APF = \frac{10R(tan36)*R}{2} \rightarrow 5R^2(tan36)$

Hexagono que isncribe a un circulo:

- *Perimetro:* El radio corresponde al apotema del hexagono, utilizando cálculos algebraicos $S = 2R(tan30)$, el perimetro es $6 * S \rightarrow 6 * (2R(tan30)) \rightarrow PHF = 12R(tan30)$

- *Área:* El área de un hexagono es $6*\frac{S^2\sqrt{3}}{4}$, el área es $6*\frac{(2R(tan30))^2\sqrt{3}}{4} \rightarrow AHF = 6R^2(tan30)^2\sqrt{3}$

2. Diseño y prueba conceptual:

FIGURAS INSCRITAS:

perimetro de un cuadrado inscrito:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: $PCI \rightarrow$perimetro de un cuadrado inscrito a una circunferencia, $PCI \in \mathbb{R}^+, PCI \neq 0$

Relación:

$PerimetroCuadradoInscrito : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$PerimetroCuadradoInscrito(R) \rightarrow 4 * (\frac{2*R}{\sqrt{2}})$

área de un cuadrado inscrito:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: ACI $\rightarrow$Área de un cuadrado inscrito a una circunferencia, $ACI \in \mathbb{R}^+, ACI \neq 0$

Relación:

$AreaCuadradoInscrito : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$AreaCuadradoInscrito(R) \rightarrow (\frac{2*R}{\sqrt{2}})^2$

perimetro de un pentagono inscrito:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: PPI $\rightarrow$ perimetro de un pentagono inscrito a una circunferencia, $PPI \in \mathbb{R}^+, PPI \neq 0$

Relación:

$PerimetroPentagonoInscrito : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$PerimetroPentagonoInscrito(R) \rightarrow 5 * \frac{2R*(tan36)*\sqrt{(tan36)^2+1}}{(tan36)^2+1}$

área de un pentagono inscrito:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: API $\rightarrow$ Área de un pentagono inscrito a una circunferencia, $API \in \mathbb{R}^+, API \neq 0$

Relación:

$AreaPentagonoInscrito : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$AreaPentagonoInscrito(R) \rightarrow \frac{5R^2 tan36}{(tan36^2)+1}$

perimetro de un heganono inscrito:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable:PHI $\rightarrow$perimetro de un hexagono inscrito a una circunferencia, $PHI \in \mathbb{R}^+, PHI \neq 0$

Relación:

$PerimetroHexagonoInscrito : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$PerimetroHexagonoInscrito(R) \rightarrow 6 * R$

área de un hexagono inscrito:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable:AHI $\rightarrow$Área de un hexagono inscrito a una circunferencia, $AHI \in \mathbb{R}^+, AHI \neq 0$

Relación:

$AreaHexagonoInscrito : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$AreaHexagonoInscrito = 3 * \frac{R^2\sqrt{3}}{2}$

FIGURAS QUE INSCRIBEN A LA CIRCUNFERENCIA:

perimetro de un cuadrado que inscribe a la circunferencia:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: PCF $\rightarrow$perimetro de un cuadrado que inscribe a un círculo, $PCF \in \mathbb{R}^+, PCF \neq 0$

Relación:

$PerimetroCuadradoFuera : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$PerimetroCuadradoFuera(R) \rightarrow 8R$

área de un cuadrado que inscribe a la circunferencia:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: ACF $\rightarrow$Área de un cuadrado que inscribe a un círculo, $ACF \in \mathbb{R}^+, ACF \neq 0$

Relación:

$AreaCuadradoFuera : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$AreaCuadradoFuera(R) \rightarrow 4R^2$

perimetro de un pentagono que inscribe a la circunferencia:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: $PPF \rightarrow$ perimetro de un pentagono que inscribe a un círculo, $PPF \in \mathbb{R}^+, PPF \neq 0$

Relación:

$PerimetroPentagonoFuera : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$PerimetroPentagonoFuera(R) \rightarrow 10R(tan36)$

área de un pentagono que inscribe a la circunferencia:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: $APF \rightarrow$ Área de un pentagono que inscribe a un círculo, $APF \in \mathbb{R}^+, APF \neq 0$

Relación:

$AreaPentagonoFuera : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$AreaPentagonoFuera(R) \rightarrow 5R^2(tan36)$

perímetro de un hexagono que inscribe a la circunferencia:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: PHF $\rightarrow$perimetro de un hexagono que inscribe a una circunferencia, $PHF \in \mathbb{R}^+, PHF \neq 0$

Relación:

$PerimetroHexagonoFuera : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$PerimetroHexagonoFuera(R) \rightarrow 12R(tan30)$

área de un hexagono que inscribe a la circunferencia:

Entradas: Una variable (R $\in \mathbb{R}^+$: R $\neq 0$)

Salidas: Una variable: AHF $\rightarrow$Área de un hexagono que inscribe a una circunferencia, $AHF \in \mathbb{R}^+, AHF \neq 0$

Relación:

$AreaHexagonoFuera : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$AreaHexagonoFuera(R) \rightarrow 6R^2(tan30)^2\sqrt{3}$

3. Codificación

El código donde se resuelve el ejercicio s eencuentra en el archivo «Ejercicio_17_Simon-Ramos.py»

Ejercicio 18

Si una araña utiliza un patrón de hexágono regular para su telaraña, y cada hexágono está separado del otro por 1cm, y la araña quiere hacer una telaraña πr2, ¿qué cantidad de telaraña requiere la araña?.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce el radio de la telaraña $R \in \mathbb{R}^+ : R \neq 0$
- La telaraña tiene un patrón hexagonal
- Cada hexagono está separado por 1 cm y el hexagono mayor está inscrito en una telaraña de radio R

1.2 Objetos desconocidos:

- Se desconoce la cantidad de telaraña necesaria

1.3 Relación entre los objetos:

- El lado del hexagono mayor es igual al radio $\rightarrow S = R$

- La cantidad de hexagonos será igual a la parte entera de R+1 si R es decimal $\rightarrow H = [\![R]\!]+1$, si R es entero la cantidad de hexagonos será igual a este valor$\rightarrow H = R$
- La cantidad de telaraña usada por cada hexagono será igual a $6 * R$, cada vez R disminuirá en 1
- A la cantidad final de telaraña usada por los hexagonos se le debe sumar tres veces el diametro, pues es la base donde se empieza a tejer la telaraña: $6 * R$

2. Diseño y prueba conceptual

cantidad total de telaraña:

Entradas: Una variable: Radio de la telaraña ($R\in \mathbb{R}^+ : R \neq 0$)

Salidas: Una variable: Cantidad total de telaraña requerida ($TT \in \mathbb{R}^+$)

Relación:

$Total\ telaraña : \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$Total\ telaraña(R) \rightarrow 6 * R + (\sum_{R-[\![R]\!]}^{R}(6 * R))$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo «Ejercicio_18_Simon-Ramos.py»

4. Varios

Ejercicio 19

Si en la UN están podando árboles y cada rama tiene P hojas, y a cada árbol le quitaron K ramas, cuántos árboles se deben podar para obtener T hojas?.

1. Análisis y especificación del problema

1.1 Datos conocidos:

- Cantidad de hojas ($P\in \mathbb{N}$) de cada rama
- Cantidad de ramas ($K\in \mathbb{N}$) de cada árbol
- Cantidad de hojas ($T\in \mathbb{N}$) que se quieren obtener

1.2 Objetos desconocidos

- Cantidad de árboles X ($X\in \mathbb{N}$) que se requieren para obtener T hojas

1.3 Relación entre los objetos:

- La cantidad de hojas que le quitan a cada árbol es $P * K$
- La cantidad de árboles necesarios será $\frac{T}{P*K}$ ó $[\![\frac{T}{P*K}]\!] + 1$dependiendo de la relación entre cantidad de hojas requeridas y hojas sacadas por árbol

2. Diseño y prueba conceptual

Entradas: Tres variables: Cantidad de hojas por rama ($P \in \mathbb{N}, P \neq 0$), Cantidad de ramas por árbol ($K \in \mathbb{N}, K \neq 0$), cantidad de hojas a obtener ($T \in \mathbb{N}$)

Salidas: Una variable: cantidad de arboles ($X \in \mathbb{N}$)

Relación:

$\acute{a}rboles : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

$$\acute{a}rboles(P, K, T) \to \begin{cases} 0, & si & T = 0 \\ \left[\!\left[\frac{T}{P*K}\right]\!\right] + 1 & si & Tmod(P*K) \neq 0 \\ \left[\!\left[\frac{T}{P*K}\right]\!\right] & , & En\ otro\ caso \end{cases}$$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_19_SimonRamos.py»

Ejercicio 20

Si un amigo, no tan amigo, me presta K pesos a i pesos de interés diario, ¿cuánto le pagaré en una semana si el interés es simple?, ¿y cuánto si el interés es compuesto?.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Cantidad de dinero prestada $K \in \mathbb{R}^+$
- Cantidad de pesos de interes diario $i \in \mathbb{R}^+$
- Tiempo del prestamo = 7 días (una semana) $T = 7$

1.2 Objetos desconocidos:

- Cantidad a pagar $X \in \mathbb{R}^+$

1.3 Relación entre los objetos:

Interés simple:

- La tasa de interés es $\frac{i}{K}$diario
- La fórmula del interés simple es $I = K * \frac{i}{K} * 7$
- La cantidad de pesos a pagar es $K + I \to K + (K * \frac{i}{K} * 7) \to K + (i * 7)$

Interés compuesto:

- La tasa de interés es $\frac{i}{K}$diario
- La formula del interés compuesto es $K(1 + \frac{i}{K})^T$

2. Diseño y prueba conceptual

Interés simple:

Entradas: Dos variables($K, i : K, i \in \mathbb{R}^+, K, i \neq 0$)

Salidas: Una variable ($X_S \in \mathbb{R}^+$)

Relación:

$interessimple : \mathbb{R}^+ x \mathbb{R}^+ \to \mathbb{R}^+$

$interessimple(K, i) \to K + (7 * i)$

interés compuesto:

Entradas: Dos variables($K, i : K, i \in \mathbb{R}^+, K, i \neq 0$)

Salidas: Una variable ($X_C \in \mathbb{N}$)

Relación:

$interescompuesto : \mathbb{R}^+ x \mathbb{R}^+ \to \mathbb{R}^+$

$interescompuesto(K, i) \to K * (1 + \frac{i}{K})^7$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_20_SimonRamos.py»

Ejercicio 21

Un niño se la pasó jugando con fichas de lego, tenia dos tipos de fichas de lego, fichas de cuadros de 1 × 1 (rojas) y fichas de cuadros de 2 × 1 (azules), y le dieron una base de 1 × n cuadritos, ¿de cuántas formas distintas puede ubicar las fichas rojas y azules sobre la base?, ¿y si le dan una ficha amarilla de 1 × 3?.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Hay una base de 1 x n cuadros ($n \in \mathbb{N}$)
- Hay fichas rojas de 1x1
- Hay fichas azules de 1x2
- Hay fichas amarillas de 1x3

1.2 Objetos desconocidos:

- Se desconoce la cantidad de formas en las que se puede llenar el espacio ($X \in \mathbb{N}$)

1.3 Relación entre los objetos:

- Si el niño inicia con una ficha roja, el espacio restante será de $n - 1$

- Si el niño inicia con una ficha roja el espacio restante será de $n-2$
- Si el niño inicia con una ficha amarilla el espacio será de $n-3$

2. Diseño y prueba conceptual

fichas rojas y azules:

Entradas: Una variable $(N \in \mathbb{N} : \mathbb{N} \neq 0)$

Salidas: Una variable $X \in \mathbb{N}$

Relación:

$fichas1 : \mathbb{N} \to \mathbb{N}$

$$fichas1(n) \to \begin{cases} 2, & si \quad n = 2 \\ 1, & si \quad n = 1 \\ 0, & si \quad n = 0 \\ fichas1(n-1) + fichas1(n-2), & si \quad n > 2 \end{cases}$$

fichas rojas, azules y amarillas:

Entradas: Una variable $(N \in \mathbb{N} : \mathbb{N} \neq 0)$

Salidas: Una variable $X \in \mathbb{N}$

Relación:

$fichas2 : \mathbb{N} \to \mathbb{N}$

$$fichas2(n) \to \begin{cases} 4, & si \quad n = 3 \\ 2, & si \quad n = 2 \\ 1, & si \quad n = 1 \\ 0, & si \quad n = 0 \\ fichas2(n-1) + fichas2(n-2) + fichas2(n-3), & si \quad n > 3 \end{cases}$$

3. Codificación

El código donde se reuselve el ejercicio se encuentra en el archivo llamado «Ejercicio_21_SimonRamos.py»

5. Arreglos

Ejercicio 22

Implementar la criba de Eratostenes para calcular los números primos en el rango 1 a n, donde n es un número natural dado por el usuario.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce el rango a calcular (1,n); donde $n \in \mathbb{N}$

1.2 Objetos desconocidos:

- Se desconocen los números primos en el rango de (1,n)

1.3 Relación:

- El primer número primo es 2, apartir de ahí se tachan todos los múltiplos de dos. Se pasa al siguiente número sin tachar y se tachan sus multiplos, se repite el procedimiento hasta acabar el rango. FInalmente, los números que no han sido tachados son números primos

2. Diseño y prueba conceptual

Función que crea el arreglo:

Entradas: 2 variables: Una cariable n que indica la cantidad de elementos del arreglo ($n \in \mathbb{N}$:n $\neq 0$); los elementos del arreglo

Salidas: Un arreglo A^* de booleanos, tal que los dos primeros elementos son True, y desde A_2 hasta $A_n$serán True

Relación:

$$\begin{array}{lcr} crea: & \mathbb{N} & \rightarrow & \mathbb{B}^* \\ crea(n) & \rightarrow & (A_0, A_1, A_2, ..., A_n) \end{array}$$ Donde $A_0 = False$, $A_1 = False$ y$\forall_{i=2}^{n} A_i = True$

Función que tacha los múltiplos:

Entradas: Un arreglo de booleanos (A^*) y un natural que indica el tamaño del arreglo ($n \in \mathbb{N} : n \neq 0$)

Salidas: Un arreglo de booleanos (B^*)

Relación:

$$\begin{array}{lcl} tacha : \mathbb{B}^* \times \mathbb{N} & \rightarrow & \mathbb{B}^* \\ tacha(A, n) & \rightarrow & B \text{ tal que } \begin{cases} \forall_{i=2}^{\lfloor \frac{n}{k} \rfloor + 1} B_{i*k} = False & si \quad \exists_{k=2}^{n} A_k = True \\ B_i = A_i & \text{En otro caso} \end{cases} \end{array}$$

función que imprime los números múltiplos:

Entrada: Un arreglo de booleanos (B^*)

Salidas: Un arreglo de números naturales (C^*)

Relación:

$$\begin{array}{lcl} imprime : \mathbb{B}^* & \rightarrow & \mathbb{N}^* \\ imprime(B) & \rightarrow & C \end{array} \text{ tal que si } \forall_{i=0}^{|B|} B_i = True \text{ entonces } i \in C$$

3. Codificación:

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_22_SimonRamos.py»

Ejercicio 23

Desarrollar un algoritmo que calcule la suma de los elementos de un arreglo de números enteros (reales).

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce la cantidad de elementos del arreglo de enteros (n); donde $n \in \mathbb{N}$y $n \neq 0$
- Se conoce la cantidad de elementos del arreglo de reales (m); donde $m \in \mathbb{N}$y$m \neq 0$
- Se conocen los elementos del arreglo(A) $\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ A^* & \rightarrow & (A_0, A_1, A_2, \ldots, A_{n-1}) \end{array}$ tal que $A_n \in \mathbb{Z}$
- Para un segundo arreglo (B), se conocen sus elementos $\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ B & \rightarrow & (B_0, B_1, B_2, \ldots, B_{m-1}) \end{array}$ tal que $B_m \in \mathbb{R}$

1.2 Obejtos desconocidos:

- Se desconoce el resultado de la suma (S) de los elementos de los arreglos
- si se hace S del arreglo B, entonces $S \in \mathbb{R}$; si se hace S del arreglo A, entonces $S \in \mathbb{Z}$

1.3 Relación entre los objetos:

- S se obtiene de la suma de todos los elementos del arreglo, tal que $S_A = A_0 + A_1 + A_2 + \cdots + A_{n-1}$, $S_B = B_0 + B_1 + B_2 + \cdots + B_{m-1}$

2. Diseño y prueba conceptual

Relación:

Elementos del arreglo enteros:

Entradas: 2 variables: (n $\in \mathbb{N}$: $n \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de enteros A

Relación:

$$\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ arregloA & (n) \rightarrow & A \text{ tal que } \forall_{i=0}^{n-1} A + [A_i] \rightarrow (A_0, A_1, A_2, \ldots, A_{n-1}) \end{array} \text{ tal que } A_i \in \mathbb{Z}$$

Elementos arreglo de reales:

Entradas: 2 variables: (m $\in \mathbb{N}$: $m \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de reales B

Relación:

$$\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ arregloB(m) & \rightarrow & B \text{ tal que } \forall_{i=0}^{n-1} B + [B_i] \rightarrow (B_0, B_1, B_2, \ldots, B_{m-1}) \end{array} \text{ tal que } B_i \in \mathbb{R}$$

Suma arreglo:

Entradas: 1 variable: un arreglo A

Salidas: 1 variable: suma de los elementos del arreglo ($S \in \mathbb{R}$)

$Suma : \mathbb{R}^* \to \mathbb{R}$

$Suma(B^*) \to (B_0 + B_1 + B_2 + \cdots + B_{m-1})$

$Suma(B^*) \to \sum_{i=0}^{|B|-1}(B_i)$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo «Ejercicio_23_Simon-Ramos.py»

Ejercicio 24

Desarrollar un algoritmo que calcule el promedio de un arreglo de enteros (reales).

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce la cantidad de elementos del arreglo de enteros (n); donde $n \in \mathbb{N} : n \neq 0$
- Se conoce la cantidad de elementos del arreglo de reales (m); donde $m \in \mathbb{N} : m \neq 0$
- Se conocen los elementos del arreglo (A) $\begin{array}{ccc} \mathbb{Z} & \to & \mathbb{Z}^* \\ A & \to & (A_0, A_1, A_2, \ldots, A_{n-1}) \end{array}$ tal que $A_n \in \mathbb{Z}$
- Para un segundo arreglo (B), se conocen sus elementos $\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^* \\ B & \to & (B_0, B_1, B_2, \ldots, B_{m-1}) \end{array}$ tal que $B_m \in \mathbb{R}$

1.2 Objetos desconocidos:

- Se desconoce el promedio del arreglo de enteros ($P_A \in \mathbb{Z}$)
- Se desconoce el promedio del arreglo de reales ($P_B \in \mathbb{R}$)

1.3 Relación entre los objetos:

- P_A se obtiene del promedio de los elementos del arreglo A
- P_B se obtiene del promedio de los elementos del arreglo B

2. Diseño y prueba conceptual

Elementos del arreglo enteros:

Entradas: 2 variables: (n $\in \mathbb{N} : n \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de enteros A

Relación:

$$\begin{array}{ccc} \mathbb{Z} & \to & \mathbb{Z}^* \\ arregloA & (n) \to & A \text{ tal que } \forall_{i=0}^{n-1} A + [A_i] \to (A_0, A_1, A_2, \ldots, A_{n-1}) \end{array} \text{ tal que } A_i \in \mathbb{Z}$$

Promedio del arreglo de enteros:

Entradas: 2 variables: Un arreglo de enteros A , cantidad de elementos del arreglo de enteros A $(n \in \mathbb{N} : n \neq 0)$

Salidas: 1 variable: promedio de los elementos del arreglo A $P_A \in \mathbb{Z}$

Relación:

$Promedio_A : \mathbb{Z}^* \times \mathbb{N} \to \mathbb{Z}$

$Promedio_A \to [\![\frac{A_0+A_1+A_2+\cdots+A_{n-1}}{n}]\!]$

$Promedio_A(A^*, n) \to \left[\!\!\left[\frac{\sum_{i=0}^{n-1}(A_i)}{n}\right]\!\!\right]$

Elementos arreglo de reales:

Entradas: 2 variables: (m $\in \mathbb{N}$: $m \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de reales B

Relación:

$$\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^* \\ arregloB(m) & \to & B \text{ tal que } \forall_{i=0}^{n-1} B + [B_i] \to (B_0, B_1, B_2, \ldots, B_{m-1}) \end{array} \text{ tal que } B_i \in \mathbb{R}$$

Promedio del arreglo de reales:

Entradas: 2 variables: Un arreglo de reales B, cantidad de elementos del arreglo de reales B $(m \in \mathbb{N} : m \neq 0)$

Salidas: 1 variable: promedio de los elementos del arreglo B, $P_B \in \mathbb{R}$

Relación:

$Promedio_B : \mathbb{R}^* \times \mathbb{N} \to \mathbb{R}$

$Promedio_B \to \frac{B_0+B_1+B_2+\cdots+B_{m-1}}{m}$

$Promedio_B(B^*, m) \to \frac{\sum_{i=0}^{m-1}(B_i)}{m}$

3. Codificación:

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_24_SimonRamos.py»

Ejercicio 25

Desarrollar un algoritmo que calcule el producto punto de dos arreglos de números enteros (reales) de igual tamaño. Sean v = (v_1, v_2, . . . , v_n) y w = (w_1, w_2, . . . , w_n) dos arreglos, el producto de v y w (notado v * w) es el número: $v_1 * w_1 + v_2 * w_2 + \cdots + v_n * w_n$.

1. Análisis y especificación del problema

1.1 Objetos conocidos:

- Se conoce el tamaño de los dos arreglos de enteros$(n \in \mathbb{N} : n \neq 0)$

- Se conoce el tamaño de los dos arreglos de reales ($m \in \mathbb{N} : m \neq 0$)
- Se conocen los elementos de los dos arreglos de enteros: $(A1) \begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ A1 & \rightarrow & (A1_0, A1_1, A1_2, \cdots, A1_{n-1}) \end{array}$; $(A2) \begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ A2 & \rightarrow & (A2_0, A2_1, A2_2, \cdots, A2_{n-1}) \end{array}$
- Se conocen los elementos de los dos arreglos de reales: $(B1) \begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ B1 & \rightarrow & (B1_0, B1_1, B1_2, \cdots, B1_{m-1}) \end{array}$; $(B2) \begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ B2 & \rightarrow & (B2_0, B2_1, B2_2, \cdots, B2_{m-1}) \end{array}$

1,2 Objetos desconocidos:

- Se desconoce el producto punto del arreglo de enteros ($PP_A \in \mathbb{Z}$)
- Se desconoce el producto punto del arreglo de reales ($PP_B \in \mathbb{R}$)

1.3 Relación entre los objetos:

- El producto punto del arreglo de enteros PP_A se obtiene de multiplicar y sumar los elementos del arreglo $A1$ y $A2$de forma ordenada
- El producto punto del arreglo de reales PP_B se obtiene de multiplicar y sumar los elementos del arreglo $B1$ y $B2$de forma ordenada

2. Diseño y prueba conceptual

Elementos de los arreglos de enteros:

Entradas: 2 variables: (n $\in \mathbb{N} : n \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 2 arreglos: dos arreglos de enteros A1 y A2

Relación:

Primer arreglo:

$$\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ arregloA1 & (n) \rightarrow & A1 \text{ tal que } \forall_{i=0}^{n-1} A1 + [A1_i] \rightarrow (A1_0, A1_1, A1_2, \ldots, A1_{n-1}) \\ A1_i \in \mathbb{Z} & & \end{array}$$ tal que

Segundo arreglo:

$$\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ arregloA & (n) \rightarrow & A2 \text{ tal que } \forall_{i=0}^{n-1} A2 + [A2_i] \rightarrow (A2_0, A2_1, A2_2, \ldots, A2_{n-1}) \\ \mathbb{Z} & & \end{array}$$ tal que $A2_i \in$

Producto punto de los arreglos de enteros:

Entradas: 2 arreglos de enteros:

Salidas: 1 variable: Producto punto del arreglo de enteros$PP_A \in \mathbb{Z}$

Relación:

$ProductoPuntoA : \mathbb{Z}^* x \mathbb{Z}^* \to \mathbb{Z}$

$ProductoPuntoA \to A1_0 * A2_0 + A1_1 * A2_1 + A1_2 * A2_2 + \cdots + A1_{n-1} * A2_{n-1}$

$ProductoPuntoA(A1^*, A2^*) \to \sum_{i=0}^{|A|-1}(A1_i * A2_i)$

Elementos de los arreglos de reales:

Entradas: 2 variables: (m $\in \mathbb{N} : m \neq 0$) que indica la cantidad de elementos del arreglo; elementos del arreglo

Salidas: 2 arreglos: Dos arreglos de reales B1 y B2

Relación:

Primer arreglo

$$\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^* \\ arregloB1(m) & \to & B1 \text{ tal que } \forall_{i=0}^{n-1} B1 + [B1_i] \to (B1_0, B1_1, B1_2, \ldots, B1_{m-1}) \end{array} \text{ tal que}$$

$B1_i \in \mathbb{R}$

Segundo arreglo

$$\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^* \\ arregloB2(m) & \to & B2 \text{ tal que } \forall_{i=0}^{n-1} B2 + [B2_i] \to (B2_0, B2_1, B2_2, \ldots, B2_{m-1}) \end{array} \text{ tal que}$$

$B2_i \in \mathbb{R}$

Producto punto de los arreglos de reales:

Entradas: 2 arreglos de reales:

Salidas: 1 variable: producto punto del arreglo de reales $PP_B \in \mathbb{R}$

Relación:

$ProdcutoPruntoB : \mathbb{R}^* x \mathbb{R}^* \to \mathbb{R}$

$ProductoPuntoB \to B1_0 * B2_0 + B1_1 * B2_1 + B1_2 * B2_2 + \cdots + B1_{m-1} * B2_{m-1}$

$ProductoPuntoB(B1^*, B2^*) \to \sum_{i=0}^{|B|-1}(B1_i * B2_i)$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo «Ejercicio_25_Simon-Ramos.py»

Ejercicio 26:

Desarrollar un algoritmo que calcule el mínimo de un arreglo de números enteros (reales).

1. Análisis y especificación

1.1 Objetos conocidos:

- Se conoce el tamaño del arreglo de enteros ($n \in \mathbb{N} : n \neq 0$)

- Se conoce el tamaño del arreglo de reales ($p \in \mathbb{N} : p \neq 0$)
- Se conocen los elementos del arreglo de enteros (A) $\begin{array}{ccc} \mathbb{Z} & \to & \mathbb{Z}^* \\ A & \to & (A_0, A_1, A_2, \cdots, A_{n-1}) \end{array}$
- Se conocen los elementos del arreglo de reales (B) $\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^* \\ B & \to & (B_0, B_1, B_2, \cdots, B_{p-1}) \end{array}$

1.2 Objetos desconocidos:

- Se desconoce el menor elemento x_{min}del arreglo de enteros y del arreglo de reales

1.3 Relación entre objetos:

- El mínimo elemento de A será aquel que, empezando a comparar desde el último elemento del arreglo, sea el menor de todos los elementos restantes
- El mínimo elemento de B será aquel que, empezando a comparar desde el último elemento del arreglo, sea el menor de todos los elementos restantes

2. Diseño y prueba conceptual

Elementos del arreglo enteros:

Entradas: 2 variables: (n $\in \mathbb{N} : n \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de enteros A

Relación:

$$\begin{array}{ccc} \mathbb{Z} & \to & \mathbb{Z}^* \\ arregloA & (n) \to & A \text{ tal que } \forall_{i=0}^{n-1} A + [A_i] \to (A_0, A_1, A_2, \ldots, A_{n-1}) \end{array} \text{ tal que } A_i \in \mathbb{Z}$$

minimo número del arreglo de enteros:

Entradas: 2 variables: Un arreglo de enteros A y la cantidad de elementos del arreglo A ($n \in \mathbb{N} : n \neq 0$)

Salidas: 1 variable: el elemento más pequeño del arreglo A ($A_{min} \in \mathbb{Z}$)

Relación:

$$\begin{array}{ccc} \mathbb{Z}^* \times \mathbb{N} & \to & \mathbb{Z} \\ \text{minimo1(A.n)} & \to & A_m \end{array} \text{talque} A_m = A_{n-1} \text{pero si} \forall_{i=n-1, i=i-1}^{0} A_i < A_m \to A_m = A_i$$

Elementos arreglo de reales:

Entradas: 2 variables: (m $\in \mathbb{N} : m \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de reales B

Relación:

$$\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^* \\ arregloB(m) & \to & B \text{ tal que } \forall_{i=0}^{n-1} B + [B_i] \to (B_0, B_1, B_2, \ldots, B_{m-1}) \end{array} \text{ tal que } B_i \in \mathbb{R}$$

minimo número del arreglo de reales:

Entradas: 2 variables: Un arreglo de reales B y la cantidad de elementos del arreglo B ($p \in \mathbb{N} : p \neq 0$)

Salidas: 1 variable: el elemento más pequeño del arreglo B ($B_{min} \in \mathbb{R}$)

Relación:

$$\begin{array}{ccc} \mathbb{R}^* \times \mathbb{N} & \rightarrow & \mathbb{R} \\ \text{minimo2(B.p)} & \rightarrow & B_m \end{array} \text{talque} B_m = B_{p-1} \text{pero si} \forall_{i=p-1,i=i-1}^{0} B_i < B_m \rightarrow B_m = B_i$$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_26_SimonRamos.py»

Ejercicio 27:

Desarrollar un algoritmo que calcule el máximo de un arreglo de números enteros (reales).

1. Análisis y especificación

1.1 Objetos conocidos:

- Se conoce el tamaño del arreglo de enteros ($n \in \mathbb{N}$)
- Se conoce el tamaño del arreglo de reales ($p \in \mathbb{N}$)
- Se conocen los elementos del arreglo de enteros (A) $\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ A & \rightarrow & (A_0, A_1, A_2, \cdots, A_{n-1}) \end{array}$
- Se conocen los elementos del arreglo de reales (B) $\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ B & \rightarrow & (B_0, B_1, B_2, \cdots, B_{p-1}) \end{array}$

1.2 Objetos desconocidos:

- Se desconoce el máximo elemento x_{max}del arreglo de enteros y del arreglo de reales

1.3 Relación entre objetos:

- El máximo elemento de A será aquel que, empezando a comparar desde el último elemento del arreglo, sea el mayor de todos los elementos restantes
- El máximo elemento de B será aquel que, empezando a comparar desde el último elemento del arreglo, sea el mayor de todos los elementos restantes

2. Diseño y prueba conceptual

Elementos del arreglo enteros:

Entradas: 2 variables: (n $\in \mathbb{N} : n \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de enteros A

Relación:

$$\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ arregloA & (n) \rightarrow & A \text{ tal que } \forall_{i=0}^{n-1} A + [A_i] \rightarrow (A_0, A_1, A_2, \ldots, A_{n-1}) \end{array} \text{tal que } A_i \in \mathbb{Z}$$

máximo número del arreglo de enteros:

Entradas: 2 variables: Un arreglo de enteros A y la cantidad de elementos del arreglo A ($n \in \mathbb{N} : n \neq 0$)

Salidas: 1 variable: el elemento más grande del arreglo A ($A_{max} \in \mathbb{Z}$)

Relación:

$$\begin{array}{ccc} \mathbb{Z}^* \times \mathbb{N} & \rightarrow & \mathbb{Z} \\ \text{maximo1(A.n)} & \rightarrow & A_m \end{array} \text{talque} A_m = A_{n-1} \text{pero si} \forall_{i=n-1, i=i-1}^{0} A_i > A_m \rightarrow A_m = A_i$$

Elementos arreglo de reales:

Entradas: 2 variables: (m $\in \mathbb{N} : m \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de reales B

Relación:

$$\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ arregloB(m) & \rightarrow & B \text{ tal que } \forall_{i=0}^{n-1} B + [B_i] \rightarrow (B_0, B_1, B_2, \ldots, B_{m-1}) \end{array} \text{tal que } B_i \in \mathbb{R}$$

máximo número del arreglo de reales:

Entradas: 2 variables: Un arreglo de reales B y la cantidad de elementos del arreglo B ($p \in \mathbb{N} : p \neq 0$)

Salidas: 1 variable: el elemento más grande del arreglo B ($B_{max} \in \mathbb{R}$)

Relación:

$$\begin{array}{ccc} \mathbb{R}^* \times \mathbb{N} & \rightarrow & \mathbb{R} \\ \text{maximo2(B.p)} & \rightarrow & B_m \end{array} \text{talque} B_m = B_{p-1} \text{pero si} \forall_{i=p-1, i=i-1}^{0} B_i > B_m \rightarrow B_m = B_i$$

3, Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_27_SimonRamos.py»

Ejercicio 28

Desarrollar un algoritmo que calcule el producto directo de dos arreglos de enteros (reales) de igual tamaño. Sean v = (v_1, v_2, . . . , v_n) y w = (w_1, w_2, . . . , w_n) dos arreglos, el producto directo de v y w (notado v ∗ w) es el vector: ($v_1 * w_1$, $v_2 * w_2$, . . . , $v_n * w_n$).

1. Análisis y especificación

1.1 Objetos conocidos:

- Se conoce el tamaño de los dos arreglos de enteros($n \in \mathbb{N}$)
- Se conoce el tamaño de los dos arreglos de reales ($m \in \mathbb{N}$)
- Se conocen los elementos de los dos arreglos de enteros ($A1$) $\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ A1 & \rightarrow & (A1_0, A1_1, A1_2, \cdots, A1_{n-1}) \end{array}$; ($A2$) $\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ A2 & \rightarrow & (A2_0, A2_1, A2_2, \cdots, A2_{n-1}) \end{array}$

- Se conocen los elementos de los dos arreglos de reales $(B1)$ $\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^* \\ B1 & \to & (A1_0, A1_1, A1_2, \cdots, A1_{m-1}) \end{array}$; $(B2)$ $\begin{array}{ccc} \mathbb{R} & \to & \mathbb{R}^* \\ B2 & \to & (B2_0, B2_1, B2_2, \cdots, B2_{m-1}) \end{array}$

1,2 Objetos desconocidos:

- Se desconoce el producto directo del arreglo de enteros $(PDA \in \mathbb{Z}^*)$
- Se desconoce el producto directo del arreglo de reales $(PDB \in \mathbb{R}^*)$

1.3 Relación entre los objetos:

- El producto directo del arreglo de enteros (PDA) se obtiene al multiplicar los elementos de $A1$y $A2$ de forma ordenada y asignando los resulatdos a un nuevo arreglo $PDA \to \mathbb{Z}^*$
- El producto directo del arreglo de reales PDB se obtiene al multiplicar los elementos del arreglo $B1$y $B2$ de forma ordenada y asignando los resultados a un nuevo arreglo $PDB \to \mathbb{R}^*$

2. Diseño y prueba conceptual

Elementos de los arreglos de enteros:

Entradas: 2 variables: (n $\in \mathbb{N} : n \neq 0$) que indica la cantidad de elementos del arreglo; elementos del arreglo

Salidas: 2 arreglos: dos arreglo de enteros A

Relación:

Primer arreglo:

$$\begin{array}{ccc} \mathbb{Z} & \to & \mathbb{Z}^* \\ A1(n) & \to & (A1_0, A1_1, A1_2, \cdots, A1_{n-1}) \end{array}$$

Segundo arreglo:

$$\begin{array}{ccc} \mathbb{Z} & \to & \mathbb{Z}^* \\ A2(n) & \to & (A2_0, A2_1, A2_2, \cdots, A2_{n-1}) \end{array}$$

Producto directo de los arreglos de enteros:

Entradas: Dos arreglos de enteros

Salida: Un arreglo de enteros resultado del producto directo

Relación:

$ProductoDirectoA : \mathbb{Z}^* x \mathbb{Z}^* \to \mathbb{Z}^*$

$ProductoDirectoA \to A1_0 * A2_0, A1_1 * A2_1, A1_2 * A2_2, \cdots, A1_{n-1} * A2_{n-1}$

$ProductoPuntoA(A1^*, A2^*) \to PDA$ tal que $\forall_{i=0}^{|A1|-1}(A1_i * A2_i) = PDA_i$ por lo que todos los elementos de PDA provienen de $A1 * A2$

Elementos de los arreglos de reales:

Entradas: 2 variables: (m $\in \mathbb{N}$: $m \neq 0$) que indica la cantidad de elementos del arreglo; elementos del arreglo

Salidas: 2 arreglos: Dos arreglos de reales B1 y B2

Relación:

Primer arreglo

$$\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ B1(m) & \rightarrow & (B1_0, B1_1, B1_2, \cdots, B1_{m-1}) \end{array}$$

Segundo arreglo

$$\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ B2(m) & \rightarrow & (B2_0, B2_1, B2_2, \cdots, B2_{m-1}) \end{array}$$

Producto directo de los arreglos de reales:

Entradas: Dos arreglos de reales

Salidas: Un arreglo de reales resultado del producto directo

Relación:

$ProdcutoDirectoB : \mathbb{R}^* x \mathbb{R}^* \rightarrow \mathbb{R}^*$

$ProductoDirectoB \rightarrow B1_0 * B2_0 + B1_1 * B2_1 + B1_2 * B2_2 + \cdots + B1_{m-1} * B2_{m-1}$

$ProductoDirectoB(B1^*, B2^*) \rightarrow PDB$ tal que $\forall_{i=0}^{|B1|-1}(B1_i * B2_i) = PDB_i$ por lo que todos los elementos de PDB provienen de $B1 * B2$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_28_SimonRamos.py»

Ejercicio 29

Desarrollar un algoritmo que determine la mediana de un arreglo de enteros (reales). La mediana es el número que queda en la mitad del arreglo después de ser ordenado.

1. Análisis y especificación

1.1 Objetos conocidos:

- Se conoce el tamaño del arreglo de enteros ($n \in \mathbb{N} : n \neq 0$)
- Se conoce el tamaño del arreglo de reales ($p \in \mathbb{N} : p \neq 0$)
- Se conocen los elementos del arreglo de enteros (A) $\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ A & \rightarrow & (A_0, A_1, A_2, \cdots, A_{n-1}) \end{array}$
- Se conocen los elementos del arreglo de reales (B) $\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ B & \rightarrow & (B_0, B_1, B_2, \cdots, B_{m-1}) \end{array}$

1.2 Objetos desconocidos:

- Se desconoce la mediana x_{med}del arreglo de enteros y del arreglo de reales

1.3 Relación entre objetos:

- Para ordenar el arreglo ascendentemente: primero se encontrará la posición del elemento mayor, posteriormente este elemento intercambiará posiciones con el elemento al final del arreglo y se repetira el proceso encontrando el elemento mayor que queda desde la primera hasta la penultima posicion, haciendo que ahora el elemento mayor ocupe la penultima posicion.
- Si la cantidad de elementos del arreglo es impar, la mediana será el elemento que ocupe la posición de la mitad después de haber sido ordenado el arreglo
- Si la cantidad de elementos es par, la mediana será el pormedio de los dos elementos de la mitad

2. Diseño y prueba conceptual

Elementos del arreglo enteros:

Entradas: 2 variables: (n $\in \mathbb{N}$: $n \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de enteros A

Relación:

$$\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ arregloA & (n) \rightarrow & A \text{ tal que } \forall_{i=0}^{n-1} A + [A_i] \rightarrow (A_0, A_1, A_2, \ldots, A_{n-1}) \end{array} \text{ tal que } A_i \in \mathbb{Z}$$

Elementos arreglo de reales:

Entradas: 2 variables: (m $\in \mathbb{N}$: $m \neq 0$) que indica la cantidad de elementos del arreglo; los elementos del arreglo

Salidas: 1 arreglo: Un arreglo de reales B

Relación:

$$\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ arregloB(m) & \rightarrow & B \text{ tal que } \forall_{i=0}^{n-1} B + [B_i] \rightarrow (B_0, B_1, B_2, \ldots, B_{m-1}) \end{array} \text{ tal que } B_i \in \mathbb{R}$$

ïndice del máximo elemento de un arreglo:

Entradas: 2 variables: Un arreglo (A^*)y la cantidad de elementos del arreglo ($n \in \mathbb{N}$: $n \neq 0$)

Salidas: 1 variable: Indice del elemento maximo del arreglo ($m \in \mathbb{N}$)

Relación:

$$\begin{array}{ccc} \mathbb{R}^* \times \mathbb{N} & \rightarrow & \mathbb{N} \\ \max(A^*, n) & \rightarrow & m \end{array} \text{talque} m = n-1 \wedge A_m = A_{n-1} \text{pero si} \forall_{i=n-1, i=i-1}^{0} A_i > A_m \rightarrow A_m = A_i \wedge m = i$$

orden de los elementos de un arreglo:

Entradas: 2 variables: Un arreglo (A^*)y la cantidad de elementos del arreglo ($n \in \mathbb{N} : n \neq 0$)

Salidas: 1 arreglo B^*

Relación:

$$\begin{array}{ccc} \mathbb{R}^* \times \mathbb{N} & \rightarrow & \mathbb{R}^* \\ ordena(A^*, n) & \rightarrow & B^* \end{array}$$ tal que $\forall_{i=n-1,i=i-1}^{0} B_i = A_{max(A^*,i)} \rightarrow B_0 \leq B_1 \leq B_2 \leq ... \leq B_{n-1}$

mediana del arreglo:

Entradas: 2 variables: 1 arreglo (B^*) y la cantidad de elementos de ese arreglo ($n \in \mathbb{N} : n \neq 0$)

Salidas: 1 variable: El número de la mediana: ($med \in \mathbb{R}$)

Relación:

$$\begin{array}{ccc} \mathbb{R}^*, \mathbb{N} & \rightarrow & \mathbb{R} \\ \text{mediana}(B^*, n) & \rightarrow & \begin{cases} \frac{B_{\frac{n}{2}} + B_{\frac{n}{2}-1}}{2} & si \quad n \bmod 2 = 0 \\ B_{[\![\frac{n}{2}]\!]} & \text{en otro caso} \end{cases} \end{array}$$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo llamado «Ejercicio_29_SimonRamos.py»

Ejercicio 30

Hacer un algoritmo que deje al final de un arreglo de números todos los ceros que aparezcan en dicho arreglo.

1. Análisis y especificación

1.1 Objetos conocidos:

- Se conoce el tamaño de arreglo ($n \in \mathbb{N} : n \neq 0$)
- Se conocen los elementos del arreglo (A) $\begin{array}{ccc} \mathbb{Z} & \rightarrow & \mathbb{Z}^* \\ A & \rightarrow & (A_0, A_1, A_2, \cdots, A_{n-1}) \end{array}$

1.2 Objetos desconocidos:

- Se desconoce la cantidad de ceros (0) que pueda tener el arreglo A
- Se desconoce el arreglo B que tendrá todos los ceros a la izquierda

1.3 Relación entre objetos:

- Del arreglo dado se crearan otros dos arreglos intermediarios, uno de ellos tendrá todos los ceros del arreglo dado y el otro todos los números distintos de cero. Finalmente, estos dos arreglos se concatenarán para formar un último arreglo con todos los ceros a la derecha

2. Diseño y prueba conceptual

Elementos del arreglo:

Entradas: 2 variables: cantidad de elementos del arreglo A ($p \in \mathbb{N} : p \neq 0$) ;elementos que conformaran el arreglo

Salidas: Un arreglo A

Relación:

$$\begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R}^* \\ \text{elementosA}(p) & \rightarrow & A \text{ tal que } \forall_{i=0}^{p-1} A + [A_i] \rightarrow (A_0, A_1, A_2, \cdots, A_{p-1}) \end{array}$$

función que separa y crea los dos arreglos intermediarios:

Entradas: Un arreglo de reales A

Salidas: Dos arreglos de reales B y C

Relación:

$$\begin{array}{ccc} separa : \mathbb{R}^* & \rightarrow & \mathbb{R}^* \times \mathbb{R}^* \\ separa(A) & \rightarrow & B, C \end{array} \text{ tal que } \begin{cases} B + [A_i] & si \quad \forall_{i=0}^{|A|-1} A_i \neq 0 \\ C + [A_i] & \text{En otro caso} \end{cases}$$

Función que devuelve el arreglo final:

Entradas: Dos arreglos de reales B y C

Salidas: Un arreglo de reales D

Relaciones:

$\begin{array}{ccc} juntar : \mathbb{R}^* \times \mathbb{R}^* & \rightarrow & \mathbb{R}^* \\ juntar(B, C) & \rightarrow & B + C \end{array}$ tal que $\forall_{i=0}^{|B|-1} B_i \neq 0$ y $\forall_{i=0}^{|C|-1} C_i = 0$ por lo tanto en el arreglo resultante D $\forall_{i=0}^{|B|-1} D_i \neq 0 \wedge \forall_{i=|B|}^{|B|+|C|-1} D_i = 0$

3. Codificación

El código donde se resuelve el ejercicio se encuentra en el archivo «Ejercicio_30_Simon-Ramos.py»