Documentación Ejercicios De Programación

Carlos Santiago Sandoval Casallas
csandovalc@unal.edu.co

21 de julio de 2021

1. Problema 1

1.1. Análisis y Especificación

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

Programa: Ej1.py

1.1.1. Objetos Conocidos

- Variable: Numero de vacas (v) $\in \mathbb{N}$.
- Variable: Litros de leche que producen por metro 1 vaca (l) $\in \mathbb{R}+$.
- Variable: Alto (n) del corral $\in \mathbb{R}+$.
- Variable: Ancho (m) del corral $\in \mathbb{R}+$.

1.1.2. Objetos Desconocidos

- Valor: El area del corral (a) $\in \mathbb{R}+$.
- Valor: Litros de leche que se producen en la granja (t) $\in \mathbb{R}+$.

1.1.3. Relación entre objetos conocidos y desconocidos

El area del corral es equivalente a multiplicar el alto por el ancho.

El total de litros producidos en la granja sera igual al area divido los litros de leche producidos por la cantidad de vacas en la granja.

1.2. Diseño y Prueba Conceptual

Entradas:

- v $\in \mathbb{N}$
- n $\wedge$ m $\wedge$ l $\in \mathbb{R}+$

Salidas:

- Valor: totalLeche $\in \mathbb{R}$.

1.3. Modelo matemático

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

$$(n, m) \mapsto \Big\{ n * m = a \Big\}$$

litrosLeche: $\mathbb{N} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

$$(v, a, l) \mapsto \begin{cases} 0, \ si \ v \ = \ 0 \\ \frac{a}{k} \ en \ otro \ caso \end{cases}$$

2. Problema 2

2.1. Análisis y Especificación

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 días, ¿en un mes cuantos huevos producen? (1 mes = 30 días).

Programa: Ej2.py

2.1.1. Objetos Conocidos

- Variable: a $\in \mathbb{N}$.
- Valor: $\frac{1}{3}$ de las aves totales son gallinas.
- Valor: Mes = 30 días.
- Frecuencia: Grupo A = 3 días.
- Frecuencia: Grupo B = 5 días.

2.1.2. Objetos Desconocidos

- Valor: Gallinas en la granja $\in \mathbb{N}$.
- Valor: Huevos a final de mes $\in \mathbb{N}$.

2.1.3. Relación entre objetos conocidos y desconocidos

Gracias a que conocemos la duración del mes, y la frecuencia con la cual los grupos de gallinas colocan 1 huevo, podemos calcular cuanto sera el total por gallina en cada grupo. Siendo respectivamente $\frac{30}{3} \wedge \frac{30}{5}$.

Con el numero de aves podemos calcular cuantas son gallinas, estas dividirlas en dos grupos iguales y en base a los datos resolver la problemática expuesta.

2.2. Diseño y Prueba Conceptual

Entradas:

- a $\in \mathbb{N}$.

Salidas:

- huevos $\in \mathbb{R}$.

2.3. Modelo matemático

huevos: $\mathbb{N} \to \mathbb{R}$.

$$(a) \mapsto \left\{ \begin{array}{c} 0 \ si \leq 2 \\ \frac{a}{3} = g \\ (\frac{g}{2} * \frac{30}{5}) + (\frac{g}{2} * \frac{30}{3}) \ En \ otro \ caso \end{array} \right\}$$

3. Problema 3

3.1. Análisis y Especificación

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?.

Programa: Ej3.py

3.1.1. Objetos Conocidos

- Variable: Numero de escorpiones (e).
- Valor: Clasificación por tamaño.
- Valor: Peso por tamaño.
- Valor: Limite de venta $\frac{2}{3}$.

3.1.2. Objetos Desconocidos

- Valor: Total de kilos que se pueden vender.
- Valor: Tamaño de los escorpiones a vender.

3.1.3. Relación entre objetos conocidos y desconocidos

Podemos inferir que se desea sacar la máxima ganancia posible, únicamente manteniendo la regla de no vender mas de $\frac{2}{3}$ de la población total, por ende se venderán los escorpiones

de tamaño mas grande.

Al tener a la población total podemos calcular cuantos escorpiones significarían $\frac{2}{3}$, esto mediante la división de los escorpiones totales en 3 partes iguales. Estos serán equivalentes a los grupos y tamaños de escorpiones existentes.

3.2. Diseño y Prueba Conceptual

Entradas:

- escorp $\in \mathbb{N}$

Salidas:

- total $\in \mathbb{R}$

3.3. Modelo matemático

kilosEscorpiones: $\mathbb{N} \to \mathbb{R}$

$$(escorp) \mapsto \left\{ \frac{(p*20)+(m*30)+(g*50))}{3000} * 2 \right\}$$

4. Problema 4

4.1. Análisis y Especificación

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?.

Programa: Ej4.py

4.1.1. Objetos Conocidos

- Variable: Alto (n) del corral $\in \mathbb{R}+$
- Variable: Ancho (m) del corral $\in \mathbb{R}+$
- Variable: Costo de la madera $\in \mathbb{R}$
- Variable: Costo de las varillas $\in \mathbb{R}$
- Variable: Costo del alambre $\in \mathbb{R}$
- Cantidad de material requerido: 4 hileras de tablas, 8 hileras de varilla y 5 hileras de puas

4.1.2. Objetos Desconocidos

- Valor: Costo total de usar madera $\in \mathbb{R}$
- Valor: Costo total de usar varillas $\in \mathbb{R}$
- Valor: Costo total de usar alambre $\in \mathbb{R}$

4.1.3. Relación entre objetos conocidos y desconocidos

A partir del ancho y el alto podemos hallar el perímetro del corral a cercar, en base al perímetro, el coste del material y el material requerido podemos calcular el costo de usar cada material.

4.2. Diseño y Prueba Conceptual

Entradas:

- $n \wedge m \in \mathbb{R}+$
- madera $\wedge$ varillas $\wedge$ alambre $\in \mathbb{R}$

Salidas:

- maderaTotal, varillasTotal, alambreTotal $\in \mathbb{R}$
- material $\in$ type(str)

4.3. Modelo matemático

cercado: $\mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow$ Material

$$(madera, varillas, puas, alto, ancho) \mapsto \left\{ \begin{array}{rcc} Perimetro & = & (corralM * 2) + (corralN * 2) \\ totalM & = & (madera * 4) * Perimetro \\ totalV & = & (varillas * 8) * Perimetro \\ totalAP & = & (puas * 5) * Perimetro \end{array} \right\}$$

$$verificacion \longrightarrow \left\{ \begin{array}{c} Si\ (totalM \leq totalV \wedge totalM \leq totalAP) \longrightarrow Material = Madera \\ Si\ (totalV \leq totalM \wedge totalV \leq totalAP) \longrightarrow Material = Varillas \\ Si\ (totalAP \leq totalV \wedge totalAP \leq totalM) \longrightarrow Material = Puas \end{array} \right\}$$

5. Problema 5

5.1. Análisis y Especificación

Función potencia de un entero elevado a un entero.

Programa: Ej5.py

5.1.1. Objetos Conocidos

- Variable: Numero base $\in \mathbb{Z}$
- Variable: Numero exponente $\in \mathbb{Z}$

5.1.2. Objetos Desconocidos

- Valor: Potencia $\in \mathbb{Z}$

5.1.3. Relación entre objetos conocidos y desconocidos

La potencia es equivalente a multiplicar la base por si misma la cantidad de veces indicado por el exponente.

5.2. Diseño y Prueba Conceptual

Entradas:

- a $\wedge$ b $\in \mathbb{Z}$

Salidas:

- potencia $\in \mathbb{Z}$

5.3. Modelo matemático

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

$$(a, b) \mapsto \begin{Bmatrix} 1 \ Si \ b = 0 \\ a * a \ ... \ a_b \end{Bmatrix}$$

6. Problema 6

6.1. Análisis y Especificación

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

6.1.1. Objetos Conocidos

- Variable: Primer numero (a) y segundo numero (b) $\mathbb{R}$

6.1.2. Objetos Desconocidos

- Valor: Modulo entre los 2 números $\in \mathbb{R}$
- Booleano: ¿Es divisible?

6.1.3. Relación entre objetos conocidos y desconocidos

La manera de saber si un numero es divisible por otro es mediante la operación modulo, si el valor de esta es igual a "0" este sera divisible.

6.2. Diseño y Prueba Conceptual

Entradas:

- a $\wedge$ b $\in \mathbb{R}$

Salida:

- Bool: True $\vee$ False

6.3. Modelo matemático

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

$$(a, b) \mapsto \begin{Bmatrix} True\ si: & a\,\%b = 0 \\ False\ si: & a\,\%b \neq 0 \end{Bmatrix}$$

7. Problema 7

7.1. Análisis y Especificación

Determinar si un numero es primo.

Programa: Ej7.py

7.1.1. Objetos Conocidos

- Variable: Numero (a) $\mathbb{Z}$

7.1.2. Objetos Desconocidos

- Booleano: ¿Es primo?

7.1.3. Relación entre objetos conocidos y desconocidos

Un numero primo es aquel numero que únicamente es divisible por el mismo y por uno.

7.2. Diseño y Prueba Conceptual

Entradas:

- a $\in \mathbb{Z}$

Salida:

- Bool: True $\vee$ False

7.3. Modelo matemático

esPrimo: $\mathbb{N} \times \mathbb{N} \to$ Bool

$$(a, i) \mapsto \left\{ \begin{array}{c} i = 2 \ \wedge \ a \geq 0 \\ Si: \ a\,\%i \ \neq \ 0 \to \ esPrimo(i+1) \\ False \ si: \ a\,\%i \ = \ 0 \\ True \ si: \ i \geq a \end{array} \right\}$$

8. Problema 8

8.1. Análisis y Especificación

Dados dos naturales, determinar si son primos relativos.

Programa: Ej8.py

8.1.1. Objetos Conocidos

- Variables: a $\wedge$ b $\in \mathbb{N}$

8.1.2. Objetos Desconocidos

- Booleano: ¿Son primos relativos?

8.1.3. Relación entre objetos conocidos y desconocidos

Los números que son primos relativos entre si, son aquellos que no tienen un divisor común diferente a uno.

8.2. Diseño y Prueba Conceptual

Entradas:

- a $\wedge$ b $\in \mathbb{N}$

Salidas:

- Bool: True $\vee$ False

8.3. Modelo matemático

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

primosRelativos: $\mathbb{N} \to$ Bool

$$mcd(a,b) \mapsto \left\{ \begin{array}{c} si\ a < b : mcd(b,a) \\ si\ a > b : c \to a\,\%b \\ si\ c = 0 \to return\ b \\ si\ c \neq 0 : mcd(b,c) \end{array} \right\}$$

$$primosRelativos(mcd) \mapsto \left\{ \begin{array}{c} True\ si\ mcd\ = 1 \\ False\ en\ otro\ caso \end{array} \right\}$$

9. Problema 9

9.1. Análisis y Especificación

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

Programa: Ej9.py

9.1.1. Objetos Conocidos

- Variables: a $\wedge$ b $\wedge$ c $\in \mathbb{Z}$

9.1.2. Objetos Desconocidos

- Valor: Total suma
- Bool: ¿El numero es múltiplo?

9.1.3. Relación entre objetos conocidos y desconocidos

Un múltiplo es el producto de un numero por algún entero. En este caso si el valor a evaluar es menor que la suma se descarta de forma inmediata que sea múltiplo.

9.2. Diseño y Prueba Conceptual

Entradas:

- a $\wedge$ b $\in \mathbb{R}$
- c $\in \mathbb{R}$

Salidas:

- Bool: True $\vee$ False

9.3. Modelo matemático

numero: $\mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow R$ Bool

$$(a,b,c) \longrightarrow \left\{ \begin{array}{c} Numero \ = \ a+b \\ c\%Numero = Var \\ Si \ Var = 0 \longrightarrow True \\ Si \ Var \neq 0 \longrightarrow False \end{array} \right\}$$

10. Problema 10

10.1. Análisis y Especificación

Dados los coeficientes de un polinomio de grado dos, evaluar el polinomio en un valor dado. $Ax^2 + bx + c$

Programa: Ej10.py

10.1.1. Objetos Conocidos

- Variables: a, b, c, x $\in \mathbb{R}$

10.1.2. Objetos Desconocidos

- Valor: Total polinomio

10.1.3. Relación entre objetos conocidos y desconocidos

Se tiene la formula $Ax^2 + bx + c$ para operar cada una de las variables presentadas por el usuario

10.2. Diseño y Prueba Conceptual

Entradas:

- a,b,c,d $\in \mathbb{R}$

Salidas:

- total $\in \mathbb{R}$

10.3. Modelo matemático

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

$$(a, b, c, x) \mapsto \Big\{a * x^2 + b * x + c\Big\}$$

11. Problema 11

11.1. Análisis y Especificación

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

Programa: Ej11.py

11.1.1. Objetos Conocidos

- Variables: Coeficientes del polinomio

11.1.2. Objetos Desconocidos

- Valor: Coeficiente lineal de la derivada

11.1.3. Relación entre objetos conocidos y desconocidos

Conocemos que en la formula $ax^2 + bx + c$ el coeficiente lineal es b. Pero al solicitar el de la derivada. Esta es equivalente a $f'(x) = 2ax + b$ de esta manera comprendemos que el resolver b igualamos a 0 y despejamos dando por resultado $b = -2ax$.

11.2. Diseño y Prueba Conceptual

Entradas:

- a,b,c,x $\in \mathbb{R}$

Salidas:

- $b = -2ax \ \in \mathbb{R}$

11.3. Modelo matemático

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

$$(a, b, c, x) \mapsto \left\{ \begin{array}{c} f(x) = ax^2 + bx + c \ \to \ f'(x) = 2ax + b \\ Igualamos\ a\ cero: \ 0 = 2ax + b \Longrightarrow b = -2ax \end{array} \right\}$$

12. Problema 12

12.1. Análisis y Especificación

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

Programa: Ej12.py

12.1.1. Objetos Conocidos

- Variables: Coeficientes del polinomio
- Variable: Punto a evaluar

12.1.2. Objetos Desconocidos

- Valor: Derivada del polinomio en el punto

12.1.3. Relación entre objetos conocidos y desconocidos

Ya poseemos el valor de la derivada a partir del ejercicio 11, de esta manera reemplazamos el valor de x por el punto indicado.

12.2. Diseño y Prueba Conceptual

Entrada:

- a,b,c,x $\in \mathbb{R}$

Salidas:

- $f'(x) = 2ax + b \ \in \mathbb{R}$

12.3. Modelo matemático

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

$$(a, b, c, x) \mapsto \Big\{ f(x) = ax^2 + bx + c \ \rightarrow \ f'(x) = 2ax + b \Big\}$$

13. Problema 13

13.1. Análisis y Especificación

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

Programa: Ej13.py

13.1.1. Objetos Conocidos

- Variable: Numero registrado por el usuario $\in \mathbb{N}$

13.1.2. Objetos Desconocidos

- Bool: ¿Hace parte de la succión de Fibonacci?

13.1.3. Relación entre objetos conocidos y desconocidos

La forma estándar de comprobar si un numero hace parte de la succión de Fibonacci es comprobando si $5 * num^2 + 4$ o $5 * num^2 - 4$ es un cuadrado perfecto, esto quiere decir que el numero producto tiene raíz entera.

La manera de permitir esta verificacion es redondeando el valor total de la raíz, si este valor por el mismo es igual al inicial el numero sera cuadrado.

13.2. Diseño y Prueba Conceptual

Entradas:

- Num $\in \mathbb{N}$

Salidas:

- True $\vee$ False

13.3. Modelo matemático

cuadradoPerfecto: $\mathbb{R} \to$ Bool

$$(x) \mapsto \left\{ \begin{array}{c} \sqrt{x} \Longrightarrow y \\ True\ si\ y * y\ = x \mid y \in \mathbb{Z} \\ False\ si\ y * y\ \neq x \mid y \in \mathbb{Z} \end{array} \right\}$$

siNumero: $\mathbb{N} \to$ Bool

$$(num) \mapsto \left\{ \begin{array}{c} 5 * num^2 + 4\ \vee\ 5 * num^2 - 4 \longrightarrow k \\ True\ si : cuadradoPerfecto(k) = True \\ False\ si : cuadradoPerfecto(k) = False \end{array} \right\}$$

14. Problema 14

14.1. Análisis y Especificación

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

Programa: Ej14.py

14.1.1. Objetos Conocidos

- Pendiente de las rectas
- Punto de corte

14.1.2. Objetos Desconocidos

- Característica rectas

14.1.3. Relación entre objetos conocidos y desconocidos

En caso de que la pendiente de las rectas sea igual y no exista punto de corte entre ellas, estas serán paralelas, de forma contraria si el valor de la recta inverso es igual a la segunda

pendiente y existe un punto de corte entre ellas, esta sera perpendicular, si no se cumple alguna de estas condiciones no se reconoce la recta como ninguno de estos.

14.2. Diseño y Prueba Conceptual

Entradas:

- m1 $\in \mathbb{R}$
- m2 $\in \mathbb{R}$
- punto corte $\forall_x \in X, x \in \mathbb{R}$

14.3. Modelo matemático

$$(m1, m2, puntoCorte) \mapsto \left\{ \begin{array}{c} Paralela\ si : m1 = m2 \wedge puntoCorte = \emptyset \\ Perpendicular\ si : (m1 * -1) = m2 \wedge \neg puntoCorte\emptyset \\ Si\ no : Null \end{array} \right\}$$

15. Problema 15

15.1. Análisis y Especificación

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

Programa: Ej15.py

15.1.1. Objetos Conocidos

- Pendiente de las rectas
- Punto de corte de las rectas

15.1.2. Objetos Desconocidos

- Puntos de intersección al origen

15.1.3. Relación entre objetos conocidos y desconocidos

Al poseer las pendientes y los puntos de corte de las rectas es posible identificar que puntos de intersección al origen poseen las rectas.

15.2. Diseño y Prueba Conceptual

Entradas:

- m1,b1,m2,b2 $\in \mathbb{R}$

Salidas:

- punto $\in \mathbb{R}$

15.3. Modelo matemático

$$(m1, b1, m2, b2) \mapsto \Big\{ punto : (b2 - b1)/(m1 - m2) \Big\}$$

16. Problema 16

16.1. Análisis y Especificación

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

Programa: Ej16.py

16.1.1. Objetos Conocidos

- Variable: Radio del circulo $\in \mathbb{R}$
- Variable: Número de lados del polígono $\in \mathbb{Z}$

16.1.2. Objetos Desconocidos

- Valor: Área del triangulo $\in \mathbb{R}$

16.1.3. Relación entre objetos conocidos y desconocidos

Cuando el circulo se encuentra dentro del polígono el radio sera equivalente a la apotema. Posteriormente dividimos el polígono en partes iguales, estos a su vez estarán divididos en 2 (división provocada por la apotema) en base a esto podemos calcular el ángulo.

Ahora poseemos cateto opuesto y ángulo, como estamos buscando el cateto adyacente usamos la función tangente, remplazamos valores y hallamos para cateto adyacente. Ya con este cateto podemos hallar el lado, con el lado el perímetro y con el perímetro finalmente el area.

16.2. Diseño y Prueba Conceptual

Entradas:

- radio $\in \mathbb{R}$
- n $\in \mathbb{Z}$

Salidas:

- Área $\in \mathbb{R}$

16.3. Modelo matemático

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

$$(radio, n) \mapsto \left\{ \begin{array}{c} \theta = \frac{360}{2*n} \\ x = radio * \tan(\theta) \\ lado = 2 * x \\ perimetro = lado * n \\ area = \frac{perimetro * apotema}{2} \end{array} \right\}$$

17. Problema 17

17.1. Análisis y Especificación

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

Programa: Ej17.py

17.1.1. Objetos Conocidos

- Variable: Radio del circulo $\in \mathbb{R}$
- Variable: Número de lados del polígono $\in \mathbb{Z}$
- Posición: Inscrito o circunscrito

17.1.2. Objetos Desconocidos

- Área del polígono
- Perimetro del polígono

17.1.3. Relación entre objetos conocidos y desconocidos

En el anterior ejercicio tenemos la manera de cubrir todos los círculos inscritos en algún polígono.

La metodología con los polígonos inscritos únicamente varia en que el calculo no se realiza mediante la tangente, si no mediante el seno. De igual manera al no poseer la apotema debemos dividir el ángulo por los grados sobre dos.

17.2. Diseño y Prueba Conceptual

Entradas:

- radio $\in \mathbb{R}$
- n $\in \mathbb{Z}$

Salidas:

- Área $\in \mathbb{R}$
- Perimetro $\in \mathbb{R}$

17.3. Modelo matemático

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

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

$$(radio, n) \mapsto \left\{ \begin{array}{c} \theta = \frac{360}{2*n} \\ x = radio * \tan(\theta) \\ lado = 2 * x \\ perimetro = lado * n \\ area = \frac{perimetro * apotema}{2} \end{array} \right\}$$

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

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

$$(radio, n) \mapsto \left\{ \begin{array}{c} \theta = \frac{360*n}{2} \\ x = radio * \sin(\theta) \\ lado = 2 * x \\ perimetro = lado * n \\ area = \frac{perimetro * apotema}{2} \end{array} \right\}$$

18. Problema 18

18.1. Análisis y Especificación

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 de πr(2) ¿qué cantidad de telaraña requiere la araña?.

Programa: Ej18.py

18.1.1. Objetos Conocidos

- Variable: Radio del hexágono $\in \mathbb{R}$
- Valor: Tamaño de la telaraña a construir $\in \mathbb{R}$
- Valor: Área del hexágono que aumenta 1 centímetro $\in \mathbb{R}$

18.1.2. Objetos Desconocidos

- Valor: Cantidad de tela que necesita para construir la telaraña

18.1.3. Relación entre objetos conocidos y desconocidos

Poseemos los valores necesarios para llevar a cabo la operación. Poseemos el area del hexágono y su aumento.

18.2. Diseño y Prueba Conceptual

Entradas:

- radio $\in \mathbb{R}$

Salidas:

- telaTotal $\in \mathbb{R}$

18.3. Modelo matemático

telaTotal: $\mathbb{R} \to \mathbb{R}$

$$(radio) \mapsto \left\{ \begin{array}{c} x = \ \pi * radio^2 + 1 \\ perimetro \ = 6 * radio \\ telaTotal = (x/perimetro) \end{array} \right\}$$

19. Problema 19

19.1. Análisis y Especificación

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?.

Programa: Ej19.py

19.1.1. Objetos Conocidos

- Variables: p,k,request $\in \mathbb{R}$

19.1.2. Objetos Desconocidos

- Valor: ¿Cuantos arboles deben ser podados para dar la cantidad de hojas solicitadas?

19.1.3. Relación entre objetos conocidos y desconocidos

Entendemos que debemos encontrar cierta cantidad de arboles a podar para cumplir con lo que se solicita, por ende inferimos que la cantidad de arboles totales es insignificante, tenemos en cuenta cuantas ramas se les quitan a los arboles y cuantas hojas poseen estas ramas. En base a esto podemos definir que $\frac{request}{p*k}$.

19.2. Diseño y Prueba Conceptual

Entradas:

- p $\in \mathbb{R}$
- k $\in \mathbb{R}$
- request $\in \mathbb{R}$

Salidas:

- totalArboles $\in \mathbb{R}$

19.3. Modelo matemático

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

$$(p, k, request) \mapsto \left\{ totalArboles = \frac{request}{p*k} \right\}$$

20. Problema 20

20.1. Análisis y Especificación

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?.

Programa: Ej20.py

20.1.1. Objetos Conocidos

- Variables: k,i $\in \mathbb{R}$
- Valor: Una semana = 7 días
- Caso: ¿Es interés simple?, ¿Es interés compuesto?

20.1.2. Objetos Desconocidos

- Valor: Total a pagar en una semana $\in \mathbb{R}$

20.1.3. Relación entre objetos conocidos y desconocidos

Conocemos el valor del préstamo, la duración, el interés aplicado y la frecuencia con la que se aplica. De igual manera poseemos el tipo de interés, en base a esto podemos hallar el valor total a pagar al finalizar la semana.

20.2. Diseño y Prueba Conceptual

Entradas:

- k $\in \mathbb{R}$
- i $\in \mathbb{R}$
- tipo $\in True \vee False$

Salidas:

- totalCobro $\in \mathbb{R}$

20.3. Modelo matemático

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

$$(k, i, 7) \mapsto \left\{7 * (\tfrac{i}{100} * k)\right\}$$

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

$$(k, i, 7) \mapsto \left\{k * (1 + \tfrac{i}{100})^7\right\}$$

21. Problema 21

21.1. Análisis y Especificación

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 $1 \times$ 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?.

Programa: Ej21.py

21.1.1. Objetos Conocidos

- Variable: n $\in \mathbb{N}$
- Caso: ¿Se usan 2 o 3 fichas(con amarilla)?

21.1.2. Objetos Desconocidos

- Formas distintas de ubicar las fichas $\in \mathbb{N}$

21.1.3. Relación entre objetos conocidos y desconocidos

La manera mas efectiva de hallar este tipo de dato es mediante la succión de Fibonacci, la cual en base al caso actuara de una u otra forma.

21.2. Diseño y Prueba Conceptual

Entradas:

- n $\in \mathbb{R}$

21.3. Modelo matemático

casoDosFichas: $\mathbb{N} \to \mathbb{R}$

$$(n) \mapsto \begin{Bmatrix} f_n = f_{n-1} + f_{n-2} \\ f_0 = 0 \\ f_1 = 1 \end{Bmatrix}$$

casoTresFichas: $\mathbb{N} \to \mathbb{R}$

$$(n) \mapsto \begin{Bmatrix} f_n = f_{n-1} + f_{n-2} + f_{n-3} \\ f_0 = 0 \\ f_1 = 1 \end{Bmatrix}$$

22. Problema 22

22.1. Análisis y Especificación

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.

Programa: Ej22.py

22.1.1. Objetos Conocidos

- Variable: n $\in \mathbb{N}$

22.1.2. Objetos Desconocidos

- Lista: Números primos hasta n $\in \mathbb{N}$

22.1.3. Relación entre objetos conocidos y desconocidos

Los números primos son aquellos números que no son divisibles por otro numero natural diferente a 1 y ellos mismos. Mediante este enunciado podemos realizar las operaciones básicas para desarrollar la criba hasta el numero indicado por el usuario. De igual manera no es necesario calcular cada numero, esto sera calculado hasta que la potencia cuadrada del numero sea mayor o igual a n.

22.2. Diseño y Prueba Conceptual

Entradas:

- n $\in \mathbb{R}$

Salidas:

- listaCriba$\{numeros \in \mathbb{N}\}$

22.3. Modelo matemático

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

$$(n) \mapsto \left\{ \begin{array}{c} numeros = [2, n] \wedge i = 0 \\ Si\ (numeros[i])^2 \leq n \Longrightarrow evaluaNumCriba(numeros[i]) \end{array} \right\}$$

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

$$(n) \mapsto \left\{ \begin{array}{c} Si\ n \leq numeros[i] \Longrightarrow\ Continua \\ Si\ n\,\%numeros[i] = 0 \Longrightarrow Elimina\ numeros \\ i = i + 1 \end{array} \right\}$$

23. Problema 23

23.1. Análisis y Especificación

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

Programa: Ej23.py

23.1.1. Objetos Conocidos

- Variables: a,b,c...n $\in \mathbb{Z}$

23.1.2. Objetos Desconocidos

- Valor: Suma arreglo $\in \mathbb{Z}$

23.1.3. Relación entre objetos conocidos y desconocidos

Los valores que ingrese el usuario serán evaluados.

23.2. Diseño y Prueba Conceptual

Entrada:

- num1,num2 ... num(n) $\in \mathbb{Z}$

Salida:

- suma $\in \mathbb{Z}$

23.3. Modelo matemático

sumaArreglo: $\mathbb{Z}_1 \times \mathbb{Z}_2 \ ... \ \mathbb{Z}_n \to \mathbb{Z}$

$$(num_1, num_2 ... num_n) \mapsto \Big\{ total = \ num_1 + num_2 \ ... \ + num_n \Big\}$$

24. Problema 24

24.1. Análisis y Especificación

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

Programa: Ej24.py

24.1.1. Objetos Conocidos

- Variables: sumaValoresCadena $\in \mathbb{Z}$
- Valor: tamañoCadena $\in \mathbb{Z}$

24.1.2. Objetos Desconocidos

- Valor: promedioCadena $\in \mathbb{R}$

24.1.3. Relación entre objetos conocidos y desconocidos

El promedio es la suma de todos los valores dividido en la cantidad de valores sumados, por esta razón usaremos la función del ejercicio 23 y esto se dividirá en el tamaño de la cadena.

24.2. Diseño y Prueba Conceptual

Entrada:

- num1,num2 ... num(n) $\in \mathbb{Z}$
- tamañoCadena $\in \mathbb{Z}$

Salida:

- promedioCadena $\in \mathbb{R}$

24.3. Modelo matemático

promedioCadena: $\mathbb{Z} \times \mathbb{Z} \to \mathbb{R}$

$$(sumaArreglo, tamanoCadena) \mapsto \left\{total = \frac{sumaArreglo}{tamanoCadena}\right\}$$

25. Problema 25

25.1. Análisis y Especificación

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 + ... + v n * w n).

Programa: Ej25.py

25.1.1. Objetos Conocidos

- Arreglo1: $a_1, b_1, c_1 ... n_1 \in \mathbb{Z}$
- Arreglo2: $a_2, b_2, c_2 ... n_2 \in \mathbb{Z}$
- Valor: Tamaño cadenas

25.1.2. Objetos Desconocidos

- Valor: productoPunto $\in \mathbb{R}$

25.1.3. Relación entre objetos conocidos y desconocidos

Poseemos 2 cadenas, el resultado solicitado por el enunciado es:

- (v 1 * w 1 + v 2 * w 2 + ... + v n * w n)

25.2. Diseño y Prueba Conceptual

Entradas:

- arreglo1 = $a_1, b_1, c_1...n_1 \in \mathbb{Z}$
- arreglo2 = $a_2, b_2, c_2...n_2 \in \mathbb{Z}$
- n $\in \mathbb{Z}$

Salida:

- $productoPunto \in \mathbb{R}$

25.3. Modelo matemático

productoPunto: $\mathbb{Z}_1 \times \mathbb{Z}_2 \to \mathbb{R}$

$$(cadena1, cadena2, n) \mapsto \left\{ v^1 * w^1 + v^2 * w^2 + ... + v^n * w^n \right\}$$

26. Problema 26

26.1. Análisis y Especificación

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

Programa: Ej26.py

26.1.1. Objetos Conocidos

- Variable: Valores del arreglo

26.1.2. Objetos Desconocidos

- Valor: Valor mínimo del arreglo $\in \mathbb{Z}$

26.1.3. Relación entre objetos conocidos y desconocidos

El numero de menor valor en el arreglo sera mostrado.

26.2. Diseño y Prueba Conceptual

Entradas:

- arreglo: $a_1, b_1, c_1 ... n_1 \in \mathbb{R}$
- n = tamaño arreglo $\in \mathbb{N}$

Salida:

- mínimo $\in \mathbb{Z}$

26.3. Modelo matemático

minimoArreglo: $\mathbb{Z}_1 \times \mathbb{Z}_2 \ ... \ \mathbb{Z}_n \to \mathbb{Z}$

$$(arreglo) \mapsto \left\{ \begin{array}{c} x = arreglo[0] \\ Si\ x \leq arreglo[i] \Longrightarrow x \\ Si\ x \geq arreglo[i] \Longrightarrow x = arreglo[i] \\ i = i + 1 \end{array} \right\}$$

27. Problema 27

27.1. Análisis y Especificación

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

Programa: Ej27.py

27.1.1. Objetos Conocidos

- Variable: Valores del arreglo

27.1.2. Objetos Desconocidos

- Valor: Valor máximo del arreglo $\in \mathbb{Z}$

27.1.3. Relación entre objetos conocidos y desconocidos

El numero de mayor valor en el arreglo sera mostrado.

27.2. Diseño y Prueba Conceptual

Entradas:

- arreglo: $a_1, b_1, c_1 ... n_1 \in \mathbb{R}$
- n = tamaño arreglo $\in \mathbb{N}$

Salida:

- máximo $\in \mathbb{Z}$

27.3. Modelo matemático

maximoArreglo: $\mathbb{Z}_1 \times \mathbb{Z}_2 \ ... \ \mathbb{Z}_n \to \mathbb{Z}$

$$(arreglo) \mapsto \left\{ \begin{array}{c} x = arreglo[0] \\ Si\ x \geq arreglo[i] \Longrightarrow x \\ Si\ x \leq arreglo[i] \Longrightarrow x = arreglo[i] \\ i = i + 1 \end{array} \right\}$$

28. Problema 28

28.1. Análisis y Especificación

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).

Programa: Ej28.py

28.1.1. Objetos Conocidos

- Valor: Tamaño de las cadenas $\in \mathbb{N}$
- Valores: Valores internos de los arreglos $\in \mathbb{Z}$

28.1.2. Objetos Desconocidos

- Producto directo de los arreglos $\in \mathbb{Z}$

28.1.3. Relación entre objetos conocidos y desconocidos

Sabemos por el enunciado que el producto directo consiste en la multiplicación de los valores de aquel arreglo. Para esto necesitaremos que los arreglos sean del mismo tamaño, en caso tal de no ser así marcara un error.

28.2. Diseño y Prueba Conceptual

Entradas:

- Arreglo1 $(elemento) \ \in \mathbb{Z}$
- Arreglo2 $(elemento) \ \in \mathbb{Z}$

Salida:

- productoDirecto $elemento \ \in \mathbb{Z}$

28.3. Modelo matemático

productoDirecto: $\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$

$$(arreglo1, arreglo2) \mapsto \Big\{ productoDirecto = (v^1 * w^1, v^2 * w^2, ..., v^n * w^n) \Big\}$$

29. Problema 29

29.1. Análisis y Especificación

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.

Programa: Ej29.py

29.1.1. Objetos Conocidos

- Valor: Tamaño arreglos $\in \mathbb{N}$
- Variables: Valores de los elementos del arreglo $\in \mathbb{Z}$

29.1.2. Objetos Desconocidos

- Orden del arreglo
- Mediana del arreglo

29.1.3. Relación entre objetos conocidos y desconocidos

La mediana es el valor que se encuentra en medio del arreglo cuando este es organizado en base a sus valores.

29.2. Diseño y Prueba Conceptual

Entradas:

- Arreglo $(elementos) \in \mathbb{Z}$

Salida:

- Mediana $\in \mathbb{R}$

29.3. Modelo matemático

mediana: $\mathbb{Z} \to \mathbb{R}$

$$(arreglo, long) \mapsto \left\{ \begin{array}{c} Si\ long\,\%2 \neq 0 : n \approx x | x \in \mathbb{Z} \\ Si\ no : \frac{long}{2} = x \\ total = \frac{arreglo[x-1]+arreglo[x]}{2} \end{array} \right\}$$

30. Problema 30

30.1. Análisis y Especificación

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

- vector original: (1, 6, 0, 7, -3, 8, 0, -2, 11)
- vector salida: (1, 6, 7, -3, 8, -2, 11, 0, 0)

Ejemplo:

- vector original: (0, 11, 36, 10, 0, 17, -23, 81, 0, 0, 12, 11, 0)
- vector salida: (11, 36, 10, 17, -23, 81, 12, 11, 0, 0, 0, 0, 0)

Programa: Ej30.py

30.1.1. Objetos Conocidos

- arreglo $\in \mathbb{R}$

30.1.2. Objetos Desconocidos

- arregloOrdenado $\in \mathbb{R}$

30.1.3. Relación entre objetos conocidos y desconocidos

Todo numero 0 va a cambiar su posición al final del arreglo.

30.2. Diseño y Prueba Conceptual

Entradas:

- arreglo $\in \mathbb{R}$

Salidas:

- arregloOrdenado $\in \mathbb{R}$

30.3. Modelo matemático

arregloOrdenado: $\mathbb{R} \to \mathbb{R}$

$$(arreglo) \mapsto \left\{ \begin{array}{c} i = 0 \\ Si\ arreglo[i] = 0 \to arreglo.remove(i) \\ arreglo.append(0) \wedge i = i + 1 \end{array} \right\}$$

31. Problema n

31.1. Análisis y Especificación

Suponga que un arreglo de enteros esta lleno de unos y ceros y que el arreglo representa un numero binario al revés. Hacer un algoritmo que calcule los números en decimal que representa dicho arreglo de unos y ceros.
Ejemplo:

- Entrada: (0,1,0,1,0,1,1) (representa el numero 1101010).
- Salida: 106

Ejemplo:

- Entrada: (1,0,0,1,0,1,1,1,1) (representa el numero 111101001).
- Salida: 389

Programa: Ej31.py

31.1.1. Objetos Conocidos

- Variable: Arreglo binario inverso $\forall X : x \in \{0, 1\}$

31.1.2. Objetos Desconocidos

- Valor: Numero decimal $\in \mathbb{N}$

31.1.3. Relación entre objetos conocidos y desconocidos

El arreglo de entrada posee un numero binario al revés, podemos invertir este arreglo y aplicar un algoritmo de conversión en base al cuadrado de la posición que nos permita finalmente sumar el total de valores que representa cada numero y hallar el resultado decimal.

31.2. Diseño y Prueba Conceptual

Entrada:

- Arreglo: $\forall X : x \in \{0, 1\}$

Salida:

- Total $\in \mathbb{N}$

31.3. Modelo matemático

binario: $\mathbb{Z}_{\not\Vdash} \times \mathbb{Z}_{\not\vDash} \ldots \mathbb{Z}_{\not\ltimes} \to \mathbb{N}$

$$(binario) \mapsto \begin{Bmatrix} t = 0 \wedge p = 0 \\ \forall_n^A : \ t + (n * 2^p) \end{Bmatrix}$$

32. Problema 32

32.1. Análisis y Especificación

Hacer un algoritmo que dado un numero entero no negativo, cree un arreglo de unos y ceros que representa el numero en binario al revés.

Ejemplo:

- Numero: 106
- Arreglo: (0,1,0,1,0,1,1) (representa el numero 1101010)

Ejemplo:

- Numero: 389
- Arreglo: (1,0,0,1,0,1,1,1,1) (representa el numero 111101001)

Programa: Ej32.py

32.1.1. Objetos Conocidos

- Variable: Num $\in \mathbb{N}$

32.1.2. Objetos Desconocidos

- Valor: Arreglo binario inverso $\forall X : x \in \{0, 1\}$

32.1.3. Relación entre objetos conocidos y desconocidos

Poseemos el numero decimal que queremos hallar, en este caso vamos a usar un algoritmo que mediante división entera y modulo nos permite hallar el valor binario del numero.

32.2. Diseño y Prueba Conceptual

Entradas:

- Variable: Numero $\in \mathbb{N}$

Salidas:

- Arreglo binario inverso $\forall X : x \in \{0, 1\}$

32.3. Modelo matemático

decimal: $\mathbb{N} \to \mathbb{N}_{\not\vdash} \ ... \ \mathbb{N}_{\ltimes}$

$$(num) \mapsto \left\{ \begin{array}{c} Si\ (\frac{num}{2} + r) \leq 1 \to Binario + (num\,\%2) \\ Binario \to Binario + (num\,\%2) \\ num \to \frac{num}{2} \end{array} \right\}$$

33. Problema 33

33.1. Análisis y Especificación

Hacer un algoritmo que calcule el Máximo Común Divisor (MCD) para un arreglo de enteros positivos.
Ejemplo:

- Arreglo: (12,20,14,124,72,2458)
- MCD del arreglo: 2

Programa: Ej33.py

33.1.1. Objetos Conocidos

- Variable: Arreglo de valores $\in \mathbb{N}$

33.1.2. Objetos Desconocidos

- Valor: Máximo común divisor $\in \mathbb{N}$

33.1.3. Relación entre objetos conocidos y desconocidos

El máximo común divisor es el mayor número que divide exactamente a dos o más números a la vez. Por ende la relación que comparten estos es que han de tener un múltiplo en común.

33.2. Diseño y Prueba Conceptual

Entradas:

- Arreglo: $[a_1, a_2, ..., a_n]$ $\forall\ a \in arreglo : a \in \mathbb{N}$

Salidas:

- n $\in \mathbb{N}$

33.3. Modelo matemático

mcd: $\mathbb{N}_1 \times \mathbb{N}_2 ... \mathbb{N}_3 \rightarrow \mathbb{N}$

$$(arreglo) \mapsto \left\{ \begin{array}{c} a, b\ si:\ a > b \vee b, a\ si:\ a < b \\ b\ si:\ a\,\%b = 0 \\ a, b = (b, (a\,\%b)) \end{array} \right\}$$

34. Problema 34

34.1. Análisis y Especificación

Hacer un algoritmo que calcule el Mınimo Común Múltiplo (MCM) para un arreglo de enteros positivos. Ejemplo:

- Arreglo: (12,20,30,15)
- MCD del arreglo: 60

Programa: Ej34.py

34.1.1. Objetos Conocidos

- Variable: Arreglo de valores $\in \mathbb{N}$

34.1.2. Objetos Desconocidos

- Valor: Mínimo común múltiplo $\in \mathbb{N}$

34.1.3. Relación entre objetos conocidos y desconocidos

El mínimo común múltiplo es el menor múltiplo común de todos ellos (o el ínfimo del conjunto de los múltiplos comunes).

34.2. Diseño y Prueba Conceptual

Entradas:

- Arreglo: $[a_1, a_2, ..., a_n]\ \forall\ a \in arreglo : a \in \mathbb{N}$
- Índice: $[x_0, x_1, ..., x_n - 1]\ \forall\ a \in arreglo : a \in \mathbb{N}$

Salidas:

- n $\in \mathbb{N}$

34.3. Modelo matemático

mcd: $\mathbb{N}_1 \times \mathbb{N}_2 ... \mathbb{N}_3 \rightarrow \mathbb{N}$

$$(arreglo) \mapsto \left\{ \begin{array}{c} mcm = a_1 \\ \frac{mcm\ *\ arreglo[\forall x\ \in\ indice]}{mcd(mcm,\ arreglo[\forall x\ \in\ indice])} \end{array} \right\}$$

35. Problema 35

35.1. Análisis y Especificación

Unión: Calcula en un arreglo la unión de los conjuntos y la imprime.

Programa: Ej35a42.py

35.1.1. Objetos Conocidos

El usuario digitada un arreglo, este representara un conjunto finito de elementos del tipo "T", de igual manera se conoce la operación de unión como el total de valores presente en ambos conjuntos. Cabe destacar que en caso de que existan repeticiones estas serán ignoradas a la hora de representar el conjunto final.

35.1.2. Objetos Desconocidos

El conjunto (representado como arreglo) producto de la unión de los dos inputs.

35.1.3. Relación entre objetos conocidos y desconocidos

Se debe realizar el conjunto total de los elementos que pertenecen a cada elemento.

35.2. Diseño y Prueba Conceptual

Entradas:

- Arreglos base donde: $\forall x \in (x_1, x_2, ...x_n)$

Salidas:

- Arreglo operación unión donde: $\forall z \in Z : Z = (X \cup Y)$

35.3. Modelo matemático

union: $\mathbb{R} = (\mathbb{R} \cup \mathbb{R})$

$$(arreglo1, arreglo2) \mapsto \left\{ Z_n \ Si \ \forall x \in X \wedge \forall y \in Y \right\}$$

36. Problema 36

36.1. Análisis y Especificación

Intersección: Calcula en un arreglo la intersección de los conjuntos y la imprime.

Programa: Ej35a42.py

36.1.1. Objetos Conocidos

Los conjuntos en manera de arreglos que ingresa el usuario y la operación intersección que equivale a todo elemento que pertenezca a ambos conjuntos.

36.1.2. Objetos Desconocidos

El conjunto (representado como arreglo) producto de la intersección de los dos inputs.

36.1.3. Relación entre objetos conocidos y desconocidos

Todo aquel elemento que pertenezca a conjunto A y B sera también un elemento del conjunto resultante.

36.2. Diseño y Prueba Conceptual

Entradas:

- Arreglos base donde: $\forall x \in (x_1, x_2, ...x_n)$

Salidas:

- Arreglo operación interseccion donde: $\forall z \in Z : Z = (X \cap Y)$

36.3. Modelo matemático

intersección: $\mathbb{R} = (\mathbb{R} \cap \mathbb{R})$

$$(arreglo1, arreglo2) \mapsto \left\{ Z_n \ Si \ \forall z \in X \wedge z \in Y \right\}$$

37. Problema 37

37.1. Análisis y Especificación

Diferencia: Calcula en un arreglo la diferencia del primero con el segundo y la imprime.
Programa: Ej35a42.py

37.1.1. Objetos Conocidos

Los conjuntos en manera de arreglos que ingresa el usuario y la operación diferencia que equivale a todo elemento que pertenezca a X pero no pertenezca a Y.

37.1.2. Objetos Desconocidos

El conjunto (representado como arreglo) producto de la diferencia de los dos inputs.

37.1.3. Relación entre objetos conocidos y desconocidos

Todo elemento que pertenezca a X pero no pertenezca a Y sera parte del conjunto resultante

37.2. Diseño y Prueba Conceptual

Entradas:

- Arreglos base donde: $\forall x \in (x_1, x_2, ...x_n)$

Salidas:

- Arreglo operación diferencia donde: $\forall z \in Z : Z = (X \triangle Y)$

37.3. Modelo matemático

diferencia: $\mathbb{R} = (\mathbb{R} \triangle \mathbb{R})$

$$(arreglo1, arreglo2) \mapsto \left\{ Z_n \ Si \ \forall z \in X \wedge z \notin Y \right\}$$

38. Problema 38

38.1. Análisis y Especificación

Diferencia simétrica: Calcula en un arreglo la diferencia simétrica de los conjuntos y la imprime.

Programa: Ej35a42.py

38.1.1. Objetos Conocidos

Los conjuntos en manera de arreglos que ingresa el usuario y la operación diferencia simétrica que equivale a todo elemento que pertenezca a X pero no pertenezca a Y.

38.1.2. Objetos Desconocidos

El conjunto (representado como arreglo) producto de la diferencia simétrica de los dos inputs.

38.1.3. Relación entre objetos conocidos y desconocidos

Todo elemento que pertenezca a X pero no pertenezca a Y sera parte del conjunto resultante

38.2. Diseño y Prueba Conceptual

Entradas:

- Arreglos base donde: $\forall x \in (x_1, x_2, ...x_n)$

Salidas:

- Arreglo operación diferencia simétrica donde: $\forall z \in Z : Z = (X \triangle Y)$

38.3. Modelo matemático

diferencia simétrica: $\mathbb{R} = (\mathbb{R} \triangle \mathbb{R})$

$$(arreglo1, arreglo2) \mapsto \left\{ \begin{array}{c} Z_n \ Si \ (\forall z \in X \wedge z \notin Y) \vee (\forall z \notin X \wedge z \in Y) \\ Falso \ si : z \in X \wedge z \in Y \end{array} \right\}$$

39. Problema 39

39.1. Análisis y Especificación

Pertenece: Lee un entero y determina si el elemento pertenece o no a cada uno de los conjuntos y lo imprime.

Programa: Ej35a42.py

39.1.1. Objetos Conocidos

El usuario deberá ingresar 2 arreglos como representación de los conjuntos, posteriormente deberá ingresar un valor entero a evaluar si este elemento pertenece a alguno de los conjuntos.

39.1.2. Objetos Desconocidos

Valores de verdad sobre si el elemento $j \in X \vee Y$

39.1.3. Relación entre objetos conocidos y desconocidos

Se deberá evaluar dentro de los 2 conjuntos si el elemento a evaluar se encuentra en los conjuntos, en caso de que sea así se retornara un valor Verdadero en lo contrario un valor Falso, se retorna 1 valor por cada conjunto para un total de 2 valores.

39.2. Diseño y Prueba Conceptual

Entradas:

- Arreglos base donde: $\forall x \in (x_1, x_2, ...x_n)$
- Valor a evaluar donde: $z \in \mathbb{Z}$

Salidas:

- Valores de verdad sobre si: $z \in X \wedge z \in Y$

39.3. Modelo matemático

pertenece: $\mathbb{Z} = (\mathbb{R} \in \mathbb{R})$

$$(arreglo1, arreglo2, entero) \mapsto \left\{ \begin{array}{c} Verdadero\ si: \ int \in X \vee int \in Y \\ Falso\ si: \ int \notin X \vee int \notin Y \end{array} \right\}$$

40. Problema 40

40.1. Análisis y Especificación

Contenido: Determina si el primer conjunto esta contenido en el segundo y lo imprime.

Programa: Ej35a42.py

40.1.1. Objetos Conocidos

- Variable: Arreglo1 $\forall x | x \in \mathbb{R}$
- Variable: Arreglo2 $\forall x | x \in \mathbb{R}$

40.1.2. Objetos Desconocidos

- Booleano: Si A $\in$ B

40.1.3. Relación entre objetos conocidos y desconocidos

En caso de que cada elemento dentro del arreglo 1 se encuentren en el arreglo 2 la función debería retornar un valor "True"

40.2. Diseño y Prueba Conceptual

Entradas:

- arreglo1 $\forall X : x \in \mathbb{R}$
- arreglo2 $\forall Y : y \in \mathbb{R}$

Salidas:

- bool: True $\vee$ False

40.3. Modelo matemático

contenido: $X \times Y \rightarrow True \ \vee \ False$

$$(arreglo1, arreglo2) \mapsto \left\{ \begin{array}{c} \forall \ n \in X, counts[n] = 1 \\ \forall \ n \in Y, counts[n] + 1 \\ Si \ X[n] \ \leq 1, \ False \\ Si \ no, \ True \end{array} \right\}$$

41. Problema 41

41.1. Análisis y Especificación

Salir: Permite al usuario salir de la aplicación.

Programa: Ej35a42.py

41.1.1. Objetos Conocidos

- Variable: elección $\in \mathbb{Z}$

41.1.2. Objetos Desconocidos

- Booleano: decisión $True \vee False$

41.1.3. Relación entre objetos conocidos y desconocidos

Para las operaciones del ejercicio se solicita al usuario un numero dentro del menú que permita conocer la operación que desea realizar, en caso de que el usuario seleccione salir se retornara la cadena "Finaliza" la cual indicara al programa terminar todas las tareas en ejecución.

41.2. Diseño y Prueba Conceptual

Entradas:

- Eleccion $\in \mathbb{Z}+$

Salidas:

- cadena

41.3. Modelo matemático

salir: $\mathbb{R} \rightarrow Cadena$

$$(eleccion) \mapsto \left\{ \begin{array}{c} Si\ eleccion = 7,\ "Finaliza" \\ Si\ no,\ operacion \end{array} \right\}$$

42. Problema 42

42.1. Análisis y Especificación

Desarrollar el programa anterior usando la representación modificada con las operaciones entre conjuntos optimizadas

Programa: Ej35a42.py

43. Problema 43

43.1. Análisis y Especificación

Evaluar: Lee un real e imprime la evaluación de los dos polinomios en dicho dato.

Programa: Ej43a49.py

43.1.1. Objetos Conocidos

- Polinomio1
- numEvaluar

43.1.2. Objetos Desconocidos

- Valor polinomio evaluado

43.1.3. Relación entre objetos conocidos y desconocidos

Se le otorgara un valor a la x, de esta manera se resuelve el polinomio y se retorna el valor total del mismo.

43.2. Diseño y Prueba Conceptual

Entradas:

- polinomio $\forall n \in polinomio, n \in \mathbb{R}$
- real $\in \mathbb{R}$

43.3. Modelo matemático

$$(polinomio, real) \mapsto \begin{Bmatrix} \forall_n, n = real \\ resolver \end{Bmatrix}$$

44. Problema 44

44.1. Análisis y Especificación

Sumar: Calcula el polinomio suma y lo imprime.

Programa: Ej43a49.py

44.1.1. Objetos Conocidos

- polinomio1
- polinomio2

44.1.2. Objetos Desconocidos

- polinomio resultante de la suma

44.1.3. Relación entre objetos conocidos y desconocidos

La manera de operar estos polinomios es agrupándolos por polinomios del mismo grado y sumando los mismos, después se organizaran en base a la exponente de su variable.

44.2. Diseño y Prueba Conceptual

Entradas:

- pol1 $\forall n \in X, n \in \mathbb{R}$
- pol2 $\forall n \in Y, n \in \mathbb{R}$

Salidas:

- polSuma $\forall n \in Z, n \in \mathbb{R}$

44.3. Modelo matemático

suma: $X, Y | \forall n \in \mathbb{R}$

$$(pol1, pol2) \mapsto \left\{ \begin{array}{c} pos = 0 \\ X[pos] + Y[pos] \\ \forall n, pos = pos + 1 \end{array} \right\}$$

45. Problema 45

45.1. Análisis y Especificación

Resta: Calcula el polinomio resta y lo imprime.

Programa: Ej43a49.py

45.1.1. Objetos Conocidos

- polinomio1
- polinomio2

45.1.2. Objetos Desconocidos

- polinomio resultante de la resta

45.1.3. Relación entre objetos conocidos y desconocidos

La manera de operar estos polinomios es agrupándolos por polinomios del mismo grado y restando los mismos, después se organizaran en base a la exponente de su variable.

45.2. Diseño y Prueba Conceptual

Entradas:

- pol1 $\forall n \in X, n \in \mathbb{R}$
- pol2 $\forall n \in Y, n \in \mathbb{R}$

Salidas:

- polResta $\forall n \in Z, n \in \mathbb{R}$

45.3. Modelo matemático

resta: $X, Y | \forall n \in \mathbb{R}$

$$(pol1, pol2) \mapsto \left\{ \begin{array}{c} pos = 0 \\ X[pos] - Y[pos] \\ \forall n, pos = pos + 1 \end{array} \right\}$$

46. Problema 46

46.1. Análisis y Especificación

Multiplicar: Calcula el polinomio multiplicación y lo imprime.

Programa: Ej43a49.py

46.1.1. Objetos Conocidos

- polinomio1
- polinomio2

46.1.2. Objetos Desconocidos

- polinomio resultante del producto

46.1.3. Relación entre objetos conocidos y desconocidos

La manera de operar estos polinomios multiplicando todos los elementos del primer polinomio por los elementos del segundo polinomio.

46.2. Diseño y Prueba Conceptual

Entradas:

- pol1 $\forall n \in X, n \in \mathbb{R}$
- pol2 $\forall n \in Y, n \in \mathbb{R}$

Salidas:

- polProduc $\forall n \in Z, n \in \mathbb{R}$

46.3. Modelo matemático

multiplicacion: $X, Y | \forall n \in \mathbb{R}$

$$(pol1, pol2) \mapsto \left\{ \begin{array}{c} pos = 0 \\ X[pos] * \forall n \in \mathbb{Y} \\ \forall n, pos = pos + 1 \end{array} \right\}$$

47. Problema 47

47.1. Análisis y Especificación

Dividir: Calcula el polinomio división del primer polinomio por el segundo y lo imprime.

Programa: Ej43a49.py

47.1.1. Objetos Conocidos

- polinomio1
- polinomio2

47.1.2. Objetos Desconocidos

- polinomio resultante de la división

47.1.3. Relación entre objetos conocidos y desconocidos

La manera de operar estos polinomios es agrupándolos por polinomios del mismo grado y dividiendo los mismos, después se organizaran en base a la exponente de su variable.

47.2. Diseño y Prueba Conceptual

Entradas:

- pol1 $\forall n \in X, n \in \mathbb{R}$
- pol2 $\forall n \in Y, n \in \mathbb{R}$

Salidas:

- polDivi $\forall n \in Z, n \in \mathbb{R}$

47.3. Modelo matemático

$$(pol1, pol2) \mapsto \left\{ \begin{array}{c} pos = 0 \\ X[pos]/Y[pos] \\ \forall n, pos = pos + 1 \end{array} \right\}$$

48. Problema 48

48.1. Análisis y Especificación

Residuo: Calcula el polinomio residuo de la división del primero por el segundo y lo imprime.

48.1.1. Objetos Conocidos

- polinomio1
- polinomio2

48.1.2. Objetos Desconocidos

- polinomio resultante del residuo

48.1.3. Relación entre objetos conocidos y desconocidos

La manera de operar estos polinomios es agrupándolos por polinomios del mismo grado y operando con residuos los mismos, después se organizaran en base a la exponente de su variable.

48.2. Diseño y Prueba Conceptual

Entradas:

- pol1 $\forall n \in X, n \in \mathbb{R}$
- pol2 $\forall n \in Y, n \in \mathbb{R}$

Salidas:

- polRes $\forall n \in Z, n \in \mathbb{R}$

48.3. Modelo matemático

$$(pol1, pol2) \mapsto \left\{ \begin{array}{c} pos = 0 \\ X[pos] \,\%Y[pos] \\ \forall n, pos = pos + 1 \end{array} \right\}$$

49. Problema 49

49.1. Análisis y Especificación

Salir: Permite salir de la aplicación al usuario.

Programa: Ej43a49.py

49.1.1. Objetos Conocidos

- Variable: elección $\in \mathbb{Z}$

49.1.2. Objetos Desconocidos

- Booleano: decisión $True \vee False$

49.1.3. Relación entre objetos conocidos y desconocidos

Para las operaciones del ejercicio se solicita al usuario un numero dentro del menú que permita conocer la operación que desea realizar, en caso de que el usuario seleccione salir se retornara la cadena "Finaliza" la cual indicara al programa terminar todas las tareas en ejecución.

49.2. Diseño y Prueba Conceptual

Entradas:

- Eleccion $\in \mathbb{Z}+$

Salidas:

- cadena

49.3. Modelo matemático

salir: $\mathbb{R} \rightarrow Cadena$

$$(eleccion) \mapsto \left\{ \begin{array}{c} Si\ eleccion = 7,\ "Finaliza" \\ Si\ no,\ operacion \end{array} \right\}$$

50. Problema 50

50.1. Análisis y Especificación

Desarrollar un algoritmo que permita sumar dos matrices de n´umeros reales (enteros).

Programa: Ej50.py

50.1.1. Objetos Conocidos

- Matrices a sumar

50.1.2. Objetos Desconocidos

- Matriz producto de la suma

50.1.3. Relación entre objetos conocidos y desconocidos

La matriz suma sera equivalente a la suma de valores individuales en la misma posición, esto quiere decir que matrices de distinto orden no pueden ser sumadas.

50.2. Diseño y Prueba Conceptual

Entradas:

- mat1 $\forall_n^{mat1}, n \in \mathbb{R}$
- mat2 $\forall_n^{mat2}, n \in \mathbb{R}$

Salidas:

- Matriz suma: $\forall_n^{mat1}, n = mat1[n] + mat2[n]$

50.3. Modelo matemático

sumaMatriz: $Z, X \in \mathbb{R}$

$$(Z, X) \mapsto \left\{ \begin{array}{c} Z_{1,1} + X + 1, 1 + ... + Z_{1,m} + X + 1, m \\ Z_{n,1} + X + n, 1 + ... + Z_{n,m} + X + n, m \end{array} \right\}$$

51. Problema 51

51.1. Análisis y Especificación

Desarrollar un algoritmo que permita multiplicar dos matrices de números reales (enteros).

Programa: Ej51.py

51.1.1. Objetos Conocidos

- matriz a, matriz b
- Dimensiones de las matrices

51.1.2. Objetos Desconocidos

- matriz producto

51.1.3. Relación entre objetos conocidos y desconocidos

Para que una matriz sea multiplicable debe presentar una relación donde el numero de columnas de la primera matriz es igual al numero de filas de la segunda matriz. Dando como resultado una matriz de las dimensiones no evaluadas.

51.2. Diseño y Prueba Conceptual

Entradas:

- matrices A,B $\forall_n \in X : n \in \mathbb{R}$
- m2 (tamaño columna) $\in \mathbb{Z}$

Salidas:

- Matriz producto $\forall_n \in X : n \in \mathbb{R}$

51.3. Modelo matemático

$$(mat1, mat2, m2) \mapsto \left\{ \begin{array}{c} \forall_{linea1} \in mat1 : x_n = linea1_n * linea2_n \\ \sum_{i=1}^{n} x_i = x_1 + x_2 + ... + x_n \end{array} \right\}$$

52. Problema 52

52.1. Análisis y Especificación

Desarrollar un programa que sume los elementos de una columna dada de una matriz.

Programa: Ej52.py

52.1.1. Objetos Conocidos

- Matriz

52.1.2. Objetos Desconocidos

- Columna a sumar
- Valor suma columna

52.1.3. Relación entre objetos conocidos y desconocidos

La columna a sumar sera indicada por el usuario, mientras que el valor total dependerá de los valores inscritos en la matriz.

52.2. Diseño y Prueba Conceptual

Entradas:

- matriz $\forall x_n \in X, n \in \mathbb{R}$
- columna $\in \mathbb{Z}$

Salidas:

- total $\in \mathbb{R}$

52.3. Modelo matemático

$$(matriz, n) \mapsto \left\{ \sum_{i=0}^{eleccion1} = matrix[i \times n] \right\}$$

53. Problema 53

53.1. Análisis y Especificación

Desarrollar un programa que sume los elementos de una fila dada de una matriz

Programa: Ej53.py

53.1.1. Objetos Conocidos

- Matriz

53.1.2. Objetos Desconocidos

- Fila a sumar
- Valor suma fila

53.1.3. Relación entre objetos conocidos y desconocidos

La fila a sumar sera indicada por el usuario, mientras que el valor total dependerá de los valores inscritos en la matriz.

53.2. Diseño y Prueba Conceptual

Entradas:

- matriz $\forall x_n \in X, n \in \mathbb{R}$
- fila $\in \mathbb{Z}$

Salidas:

- total $\in \mathbb{R}$

53.3. Modelo matemático

$$(matriz, n) \mapsto \left\{ \sum_{i=0}^{n} = matriz[n \times i] \right\}$$

54. Problema 54

54.1. Análisis y Especificación

Desarrollar un algoritmo que determine si una matriz es mágica. Se dice que una matriz cuadrada es mágica si la suma de cada una de sus filas, de cada una de sus columnas y de cada diagonal es igual.

Programa: Ej54.py

54.1.1. Objetos Conocidos

- matriz
- dimensiones de la matriz

54.1.2. Objetos Desconocidos

- La matriz es mágica?

54.1.3. Relación entre objetos conocidos y desconocidos

Para que la matriz se considere como una matriz mágica la suma de cada una de sus filas, cada una de sus columnas y las diagonales tienen el mismo resultado.

54.2. Diseño y Prueba Conceptual

Entradas:

- matriz $\forall x_n \in X, n \in \mathbb{R}$
- n1 $\in \mathbb{Z}$
- m1 $\in \mathbb{Z}$

Salida:

- esMagica $True \vee False$

54.3. Modelo matemático

$$suma\_magica\_filas(matriz, n1) \mapsto \left\{ \begin{array}{c} \forall n \in matriz \\ A = \sum_{i=0}^{n1} suma\_filas(matriz, i+1) \end{array} \right\}$$

$$suma\_magica\_columnas(matriz, m1) \mapsto \left\{ \begin{array}{c} \forall m \in matriz \\ B = \sum_{i=0}^{m1} suma\_columnas(matriz, i+1) \end{array} \right\}$$

$$suma\_magica\_diagonal(matriz) \mapsto \left\{ \begin{array}{c} \sum_{n=0}^{matriz[0]} n = n_1 + n_2 + ... + n_n \\ \forall_{i=0}^{n} : A\{matriz[i][i]\} \end{array} \right\}$$

$$suma\_magica\_diagonal\_contraria(matriz) \mapsto \left\{ \begin{array}{c} matriz = inverso_arreglo(matriz) \\ \sum_{n=0}^{matriz[0]} n = n_1 + n_2 + ... + n_n \\ \forall_{i=0}^{n} : A\{matriz[i][i]\} \end{array} \right\}$$

$$es\_magica \mapsto \left\{ \begin{array}{c} False\ si : \forall n \in operaciones, n \neq n+1 \vee n1 \\ Si\ no : True \end{array} \right\}$$

55. Problema 55

55.1. Análisis y Especificación

Desarrollar un algoritmo que dado un entero, reemplace en una matriz todos los números mayores a un numero dado por un uno y todos los menores o iguales por un cero.

Programa: Ej55.py

55.1.1. Objetos Conocidos

- matriz
- entero_remplazo

55.1.2. Objetos Desconocidos

- matriz_reemplazo

55.1.3. Relación entre objetos conocidos y desconocidos

Todo valor de la matriz que posea un valor mayor al registrado por el usuario cambie su valor por "1" y el que sea menor o igual "0"

55.2. Diseño y Prueba Conceptual

Entradas:

- matriz $\forall x_n \in X, n \in \mathbb{R}$
- entero $\in \mathbb{Z}$

Salidas:

- matriz $\forall x_n \in X, n \in \{0, 1\}$

55.3. Modelo matemático

$$(matriz, entero) \mapsto \left\{ \begin{array}{c} \forall_m^{len(matriz)}, \forall_n^{matriz[m]} \\ A[m][n] = 1 \ Si: \ n > entero \\ A[m][n] = 0 \ Si \ no \end{array} \right\}$$

56. Problema 56

56.1. Análisis y Especificación

Desarrollar un programa que calcule el determinante de una matriz cuadrada.

Programa: Ej56.py

56.1.1. Objetos Conocidos

- Matriz A

56.1.2. Objetos Desconocidos

- determinante matriz

56.1.3. Relación entre objetos conocidos y desconocidos

Es un numero que representa a la matriz, esta misma sera modificada hasta llegar a una determinante simple de una matriz 2×2

56.2. Diseño y Prueba Conceptual

Entradas:

- Matriz A $\forall A_n \in A, n \in \mathbb{R}$

Salida:

- determinante $\in \mathbb{R}$

56.3. Modelo matemático

$$(m) \mapsto \left\{ \begin{array}{c} Caso\ 1: \\ len(m) = 2 \rightarrow total = m[0][0] * m[1][1] - m[1][0] * m[0][1] \end{array} \right\}$$

$$(m) \mapsto \left\{ \begin{array}{c} Caso\ 2: \\ indice = \forall i \in I = len(m) : indice[i_1, i_2, ..., i_m] \\ \forall i \in indice : mc = m \wedge a = len(mc) \\ \forall k \in a : mc[k] = mc[k][0 : indices[i]] + mc[k][indices[i] + 1 :] \\ temp = (-1)^{(indices[i]\,\%2)} \\ sd = determinante_m atriz(mc) total = (total + temp * m[0][indices[i]] * sd) \end{array} \right\}$$

57. Problema 57

57.1. Análisis y Especificación

Desarrollar un programa que dadas una matriz cuadrada A y un arreglo de números reales del mismo tamaño B, calcule una solución x para el sistema de ecuaciones lineales Ax=B

Programa: Ej57.py

57.1.1. Objetos Conocidos

- Matriz A
- Arreglo B

57.1.2. Objetos Desconocidos

- Arreglo X

57.1.3. Relación entre objetos conocidos y desconocidos

Se despeja la X, manera tal que la ecuación A/B sera la forma de solucionar el ejercicio, se realiza esta operación con cada elemento del arreglo y la matriz correspondiente a la misma posición

57.2. Diseño y Prueba Conceptual

Entradas:

- Matriz A $\forall A_n \in A, n \in \mathbb{R}$
- Arreglo B $\forall B_n \in X, n \in \mathbb{R}$

Salida:

- Arreglo X $\forall x_n \in X, x_n = \frac{A_n}{B_n}$

57.3. Modelo matemático

$$(matriz, arreglo) \mapsto \begin{Bmatrix} 0, \ Si\ len(matriz) \neq len(arreglo) \\ \sum_{i=0}^{len(matriz)} X_i = \{\frac{B_i}{A_i}\} \end{Bmatrix}$$

58. Problema 58

58.1. Análisis y Especificación

Desarrollar un programa que calcule la inversa de una matriz cuadrada.

Programa: Ej58.py

58.1.1. Objetos Conocidos

- Matriz A

58.1.2. Objetos Desconocidos

- determinante matriz

58.1.3. Relación entre objetos conocidos y desconocidos

Esta puede ser calculada mediante la inversa y matriz transpuesta

58.2. Diseño y Prueba Conceptual

Entradas:

- Matriz A $\forall A_n \in A, n \in \mathbb{R}$

Salida:

- matriz inversa $\forall A_n \in A, n \in \mathbb{R}$

58.3. Modelo matemático

$$(matriz) \mapsto \left\{ inversa = \frac{1}{determinante(matriz)} * transpuesta(adjunta(m)) \right\}$$

59. Problema 59

59.1. Análisis y Especificación

Desarrollar un programa que tome un arreglo de tamaño n2 y llene en espiral hacia adentro una matriz cuadrada de tamaño n

Programa: Ej59.py

59.1.1. Objetos Conocidos

- Arreglo

59.1.2. Objetos Desconocidos

- Matriz espiral

59.1.3. Relación entre objetos conocidos y desconocidos

El tamaño del arreglo es equivalente a n^2, por este motivo para armar la matriz espiral se le sacara raiz cuadrada a este valor y se conformara la matriz de tamaño n con los valores del arreglo

59.2. Diseño y Prueba Conceptual

Entradas:

- arreglo $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- matriz_espiral $\forall y_n \in Y, n \in \mathbb{R}$

59.3. Modelo matemático

$$(arreglo) \mapsto \left\{ \begin{array}{c} n = \sqrt{len(arreglo)} \\ \sum_0^{len(arreglo)} A = \forall_{i=0}^{n} A[i][arreglo[pos]] \end{array} \right\}$$

60. Problema 60

60.1. Análisis y Especificación

Una matriz se puede usar para representar una relación entre dos conjuntos A y B. Esta representación es como sigue: Si $A = x0, x1, x2, ..., x_{n1}$ y $B = y0, y1, y2, ..., y_{m1}$ una relación R de A en B se representa mediante una matriz de unos y ceros de tamaño n×m, donde $A_{ij} = 1$ si el elemento x_i se relaciona con el elemento y_j, en caso contrario $A_{ij} = 0$.

Unión: Calcula e imprime la relación unión.

Programa: Ej60a70.py

60.1.1. Objetos Conocidos

60.1.2. Objetos Desconocidos

60.1.3. Relación entre objetos conocidos y desconocidos

60.2. Diseño y Prueba Conceptual

60.3. Modelo matemático

61. Problema 61

61.1. Análisis y Especificación

Programa: Ej60a70.py

61.1.1. Objetos Conocidos

- Matriz binaria (A,B)

61.1.2. Objetos Desconocidos

- Matriz Unión (X)

61.1.3. Relación entre objetos conocidos y desconocidos

La unión de matrices binarias tomara su valor en base a si hay la presencia de un "1" en la pareja ordenada.

61.2. Diseño y Prueba Conceptual

Entradas:

- matriz A $\forall x_n \in X, n \in \mathbb{R}$
- matriz B $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- matriz X $\forall x_n \in X, n \in \mathbb{R}$

61.3. Modelo matemático

$$(A,B) \mapsto \begin{Bmatrix} \forall_{n,m} : A[n][m], B[n][m] \\ 1\ Si: \ \exists 1 \in A, B[n][m] \\ 0\ Si\ no \end{Bmatrix}$$

62. Problema 62

62.1. Análisis y Especificación

Programa: Ej60a70.py

62.1.1. Objetos Conocidos

- Matriz binaria (A,B)

62.1.2. Objetos Desconocidos

- Matriz Intersección (X)

62.1.3. Relación entre objetos conocidos y desconocidos

La unión de matrices binarias tomara su valor en base a si ambos valores son un 1, en caso contrario tomara el valor de 0

62.2. Diseño y Prueba Conceptual

Entradas:

- matriz A $\forall x_n \in X, n \in \mathbb{R}$
- matriz B $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- matriz X $\forall x_n \in X, n \in \mathbb{R}$

62.3. Modelo matemático

$$(A,B) \mapsto \left\{ \begin{array}{c} \forall_{n,m} : A[n][m], B[n][m] \\ 1\ Si:\ A[n][m] = 1 \wedge B[n][m] = 1 \\ 0\ Si\ no \end{array} \right\}$$

63. Problema 63

63.1. Análisis y Especificación

Programa: Ej60a70.py

63.1.1. Objetos Conocidos

- Relación binaria (X)
- Conjunto (A)

63.1.2. Objetos Desconocidos

- Es simétrica?

63.1.3. Relación entre objetos conocidos y desconocidos

Es simétrica si un par ordenado (a,b) pertenece a la relación entonces el par (b,a) también pertenece a esa relación:

63.2. Diseño y Prueba Conceptual

Entradas:

- Relación X $\forall x_n \in X, n \in \mathbb{R}$
- Conjunto A $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- bool $True \vee False$

63.3. Modelo matemático

$$(A, X) \mapsto \left\{ \begin{array}{c} True\ si : \forall a, b \in \mathbf{A} : (a, b) \in \mathbf{X} \rightarrow (b, a) \in \mathbf{X} \\ False\ si\ no \end{array} \right\}$$

64. Problema 64

64.1. Análisis y Especificación

Programa: Ej60a70.py

64.1.1. Objetos Conocidos

- Relación binaria (X)
- Conjunto (A)

64.1.2. Objetos Desconocidos

- Es transitiva?

64.1.3. Relación entre objetos conocidos y desconocidos

Es transitiva si dado los elementos a, b, c del conjunto, si a está relacionado con b y b está relacionado con c, entonces a está relacionado con c

64.2. Diseño y Prueba Conceptual

Entradas:

- Relación X $\forall x_n \in X, n \in \mathbb{R}$
- Conjunto A $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- bool $True \vee False$

64.3. Modelo matemático

$$(A, X) \mapsto \left\{ \begin{array}{c} True\ si : \forall a, b, c \in A : ((a, b) \in \mathbf{X} \wedge (b, c) \in) \rightarrow (a, c) \in \mathbf{X} \\ False\ si\ no \end{array} \right\}$$

65. Problema 65

65.1. Análisis y Especificación

Programa: Ej60a70.py

65.1.1. Objetos Conocidos

- Relación binaria (X)
- Conjunto (A)

65.1.2. Objetos Desconocidos

- Es ordenada?

65.1.3. Relación entre objetos conocidos y desconocidos

Es ordenada si es reflexiva, antisimétrica y transitiva

65.2. Diseño y Prueba Conceptual

Entradas:

- Relación X $\forall x_n \in X, n \in \mathbb{R}$
- Conjunto A $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- bool $True \vee False$

65.3. Modelo matemático

$$(A, X) \mapsto \left\{ \begin{array}{c} True\ si : \neg(simetrica(A, X)) \wedge reflexiva(A, X) \wedge transitiva(A, X) \\ False\ si\ no \end{array} \right\}$$

66. Problema 66

66.1. Análisis y Especificación

Programa: Ej60a70.py

66.1.1. Objetos Conocidos

- Relación binaria (X)
- Conjunto (A)

66.1.2. Objetos Desconocidos

- Es equivalente?

66.1.3. Relación entre objetos conocidos y desconocidos

Es equivalente si es reflexiva, simétrica y transitiva.

66.2. Diseño y Prueba Conceptual

Entradas:

- Relación X $\forall x_n \in X, n \in \mathbb{R}$
- Conjunto A $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- bool $True \vee False$

66.3. Modelo matemático

$$(A, X) \mapsto \left\{ \begin{array}{c} True\ si : (simetrica(A, X) \wedge reflexiva(A, X) \wedge transitiva(A, X)) \\ False\ si\ no \end{array} \right\}$$

67. Problema 67

67.1. Análisis y Especificación

Programa: Ej60a70.py

67.1.1. Objetos Conocidos

- Relación binaria (X)

67.1.2. Objetos Desconocidos

- Es una función?

67.1.3. Relación entre objetos conocidos y desconocidos

Una función es una relación en donde a cada elemento de un conjunto (A) le corresponde uno y sólo un elemento de otro conjunto (B).

67.2. Diseño y Prueba Conceptual

Entradas:

- Relación X $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- bool $True \vee False$

67.3. Modelo matemático

$$(X) \mapsto \left\{ \begin{array}{c} True\ si : \forall x, y \in \mathbf{X} : \nexists x \in D \to D \cup \{x\} \\ False\ si\ no \end{array} \right\}$$

68. Problema 68

68.1. Análisis y Especificación

Programa: Ej60a70.py

68.1.1. Objetos Conocidos

- Relación binaria (X)

68.1.2. Objetos Desconocidos

- Es una función inyectiva?

68.1.3. Relación entre objetos conocidos y desconocidos

Una función inyectiva si las imágenes de elementos distintos son distintas o, de modo equivalente, si solo asigna imágenes idénticas a elementos idénticos.

68.2. Diseño y Prueba Conceptual

Entradas:

- Relación X $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- bool $True \vee False$

68.3. Modelo matemático

$$(X) \mapsto \left\{ \begin{array}{c} True\ si: \\ (a, a" \in \mathbf{A} \wedge a \neq a" \rightarrow f(a) \neq f(a")) \vee (a, a" \in \mathbf{A} \wedge f(a) = f(a") \rightarrow a = a") \\ False\ si\ no \end{array} \right\}$$

69. Problema 69

69.1. Análisis y Especificación

Programa: Ej60a70.py

69.1.1. Objetos Conocidos

- Relación binaria (X)

69.1.2. Objetos Desconocidos

- Es una función sobreyectiva?

69.1.3. Relación entre objetos conocidos y desconocidos

Una función f : A → B se dice que es sobreyectiva si su imagen es igual a su codominio, de modo equivalente, si todo elemento del codominio es la imagen de algún elemento del dominio

69.2. Diseño y Prueba Conceptual

Entradas:

- Relación X $\forall x_n \in X, n \in \mathbb{R}$

Salidas:

- bool $True \vee False$

69.3. Modelo matemático

$$(f) \mapsto \left\{ \begin{array}{c} True\ si : [Im(f) = B] \vee [\forall b \in \mathbf{B} \exists a \in \mathbf{A} : f(a) = b] \\ False\ si\ no \end{array} \right\}$$

70. Problema 70

70.1. Análisis y Especificación

Salir: Permite al usuario salir de la aplicación.

Programa: Ej60a70.py

70.1.1. Objetos Conocidos

- Variable: elección $\in \mathbb{Z}$

70.1.2. Objetos Desconocidos

- Booleano: decisión $True \vee False$

70.1.3. Relación entre objetos conocidos y desconocidos

Para las operaciones del ejercicio se solicita al usuario un numero dentro del menú que permita conocer la operación que desea realizar, en caso de que el usuario seleccione salir se retornara la cadena "Finaliza" la cual indicara al programa terminar todas las tareas en ejecución.

70.2. Diseño y Prueba Conceptual

Entradas:

- Eleccion $\in \mathbb{Z}+$

Salidas:

- cadena

70.3. Modelo matemático

salir: $\mathbb{R} \to Cadena$

$$(eleccion) \mapsto \left\{ \begin{array}{c} Si\ eleccion = 7,\ "Finaliza" \\ Si\ no,\ operacion \end{array} \right\}$$

71. Problema 71

71.1. Análisis y Especificación

Desarrollar un algoritmo que reciba como entrada un carácter y de como salida el numero de ocurrencias de dicho carácter en una cadena de caracteres.

Programa: Ej71.py

71.1.1. Objetos Conocidos

- carácter a evaluar
- cadena de caracteres

71.1.2. Objetos Desconocidos

- Numero de ocurrencias del carácter $\in \mathbb{Z}$

71.1.3. Relación entre objetos conocidos y desconocidos

El carácter a evaluar debe estar presente en el texto, posteriormente solo se enumerara el numero de ocurrencias del mismo.

71.2. Diseño y Prueba Conceptual

Entradas:

- caracter $\in$ **char**
- cadena $\in$ **string**

Salidas:

- total $\in \mathbb{Z}$

71.3. Modelo matemático

$$(c, S) \mapsto \left\{ \begin{array}{c} total = 0 \\ \forall s \in \mathbf{S} : s = c \rightarrow total + 1 \end{array} \right\}$$

72. Problema 72

72.1. Análisis y Especificación

Desarrollar un algoritmo que reciba como entrada dos cadenas y determine si la primera es subcadena de la segunda. (No se deben usar operaciones de subcadenas propias del lenguaje de programación).

Programa: Ej72.py

72.1.1. Objetos Conocidos

- subcadena a evaluar
- cadena de caracteres

72.1.2. Objetos Desconocidos

- Presencia de la subcadena en la cadena $True \vee False$

72.1.3. Relación entre objetos conocidos y desconocidos

La subcadena debe existir sin modificaciones en la cadena principal.

72.2. Diseño y Prueba Conceptual

Entradas:

- subcadena $\in$ **string**
- cadena $\in$ **string**

Salidas:

- bool $True \vee False$

72.3. Modelo matemático

$$(C, sc) \mapsto \begin{Bmatrix} True\ si : \exists sc \in \mathbf{C} \\ False\ si\ no \end{Bmatrix}$$

73. Problema 73

73.1. Análisis y Especificación

Desarrollar un algoritmo que reciba dos cadenas de caracteres y determine si la primera esta incluida en la segunda. Se dice que una cadena esta incluida en otra, si todos los caracteres (con repeticiones) de la cadena esta en la segunda cadena sin tener en cuenta el orden de los caracteres.

Programa: Ej73.py

73.1.1. Objetos Conocidos

- subcadena a evaluar
- cadena de caracteres

73.1.2. Objetos Desconocidos

- Presencia de la subcadena en caracteres en la cadena $True \vee False$

73.1.3. Relación entre objetos conocidos y desconocidos

La subcadena debe existir respecto a numero de repeticiones en la cadena principal.

73.2. Diseño y Prueba Conceptual

Entradas:

- subcadena ∈ **string**
- cadena ∈ **string**

Salidas:

- bool $True \vee False$

73.3. Modelo matemático

$$(C, sc) \mapsto \left\{ \begin{array}{c} \sum A[n] = A[n] + 1 : A[n] = \mathbf{C}_n \\ \sum B[n] = B[n] + 1 : A[n] = \mathbf{sc}_n \\ \forall n \in B \wedge \forall n \in A : X_n = A_n B_n = 0 \\ True\ si : \forall n \in \mathbf{X} : X_n = 0 \\ False\ si\ no \end{array} \right\}$$

74. Problema 74

74.1. Análisis y Especificación

Desarrollar un algoritmo que invierta una cadena de caracteres.

Programa: Ej74.py

74.1.1. Objetos Conocidos

- cadena

74.1.2. Objetos Desconocidos

- la cadena invertida

74.1.3. Relación entre objetos conocidos y desconocidos

La cadena debe ser inversa, esto corresponde a que las posiciones o método de lectura es contrario al habitual.

74.2. Diseño y Prueba Conceptual

Entradas:

-
- cadena ∈ **string**

Salidas:

- la cadena invertida ∈ **string**

74.3. Modelo matemático

$$(c) \mapsto \begin{Bmatrix} \forall cin\mathbf{C} : c = len(C) \\ \forall_{i=c}^{-1} : i = i - 1 : c[i] \end{Bmatrix}$$

75. Problema 75

75.1. Análisis y Especificación

Desarrollar un algoritmo que determine si una cadena de caracteres es palındrome. Una cadena se dice palındrome si al invertirla es igual a ella misma.

Programa: Ej75.py

75.1.1. Objetos Conocidos

- cadena

75.1.2. Objetos Desconocidos

- Si es palindrome

75.1.3. Relación entre objetos conocidos y desconocidos

Una cadena palíndroma es aquella que tanto de forma normal o inversa es igual

75.2. Diseño y Prueba Conceptual

Entradas:

-
- cadena $\in$ **string**

Salidas:

- bool $True \vee False$

75.3. Modelo matemático

$$(c) \mapsto \begin{Bmatrix} True\ Si : c = inverso_cadena(c) \\ False\ si\ no \end{Bmatrix}$$

76. Problema 76

76.1. Análisis y Especificación

Desarrollar un algoritmo que determina si una cadena de caracteres es frase palındrome. Una cadena se dice frase palındrome si la cadena al eliminarle los espacios es palındrome.

Programa: E76.py

76.1.1. Objetos Conocidos

- cadena

76.1.2. Objetos Desconocidos

- Si es palindrome (excluyendo espacios)

76.1.3. Relación entre objetos conocidos y desconocidos

Una cadena palíndroma es aquella que tanto de forma normal o inversa es igual (esto sin contar los espacios)

76.2. Diseño y Prueba Conceptual

Entradas:

-
- cadena $\in$ **string**

Salidas:

- bool $True \vee False$

76.3. Modelo matemático

$$(C) \mapsto \left\{ \begin{array}{c} \forall c \in C : c \neq \text{" "} \rightarrow nc + c \\ True\ Si : nc = inverso_cadena(nc) \\ False\ si\ no \end{array} \right\}$$

77. Problema 77

77.1. Análisis y Especificación

Desarrollar un algoritmo que realice el corrimiento circular a izquierda de una cadena de caracteres. El corrimiento circular a izquierda es pasar el primer carácter de una cadena como ultimo carácter de la misma.

Programa: Ej77.py

77.1.1. Objetos Conocidos

- cadena

77.1.2. Objetos Desconocidos

- la cadena con corrimiento circular

77.1.3. Relación entre objetos conocidos y desconocidos

El primer dígito sera el ultimo de la nueva cadena

77.2. Diseño y Prueba Conceptual

Entradas:

-

- cadena ∈ **string**

Salidas:

- la cadena modificada ∈ **string**

77.3. Modelo matemático

$$(C) \mapsto \left\{ \begin{array}{c} N = \text{""} \\ \forall c \in C : c = 0 \rightarrow N \\ (c \neq (len(C) - 1)) \rightarrow N + C_c \\ (c = (len(C) - 1)) \rightarrow N + (C_c + C_0) \end{array} \right\}$$

78. Problema 78

78.1. Análisis y Especificación

Desarrollar un algoritmo que realice el corrimiento circular a derecha de una cadena de caracteres. El corrimiento circular a derecha de una cadena es poner el ultimo carácter de la cadena como primer carácter de la misma.

Programa: Ej78.py

78.1.1. Objetos Conocidos

- cadena

78.1.2. Objetos Desconocidos

- la cadena con corrimiento circular

78.1.3. Relación entre objetos conocidos y desconocidos

El ultimo dígito sera el primero de la nueva cadena

78.2. Diseño y Prueba Conceptual

Entradas:

- cadena ∈ **string**

Salidas:

- la cadena modificada ∈ **string**

78.3. Modelo matemático

$$(C) \mapsto \left\{ \begin{array}{c} A = corrimiento_circular_izq(inverso_cadena(C)) \\ inverso_cadena(A) \end{array} \right\}$$

79. Problema 79

79.1. Análisis y Especificación

Desarrollar un algoritmo que codifique una cadena de caracteres mediante una cadena de correspondencias de caracteres dada. La cadena de correspondencias tiene como el primer carácter el carácter equivalente para el carácter "a", el segundo carácter para la "b"y ası sucesivamente hasta la "z". No se tiene traducción para las mayúsculas ni para la "ñ".

Programa: Ej79.py

79.1.1. Objetos Conocidos

- Alfabeto
- Cadena
- Cadena de codificación

79.1.2. Objetos Desconocidos

- Cadena codificada

79.1.3. Relación entre objetos conocidos y desconocidos

Cada uno de los valores presentados en la cadena de codificación tomaran un nuevo valor respecto al alfabeto y el mensaje original (cadena) sera reemplazado por estas nuevas reglas.

79.2. Diseño y Prueba Conceptual

Entradas:

- cadena ∈ **string**
- cadena de codificacion ∈ **string**

Salidas:

- Cadena codificada ∈ **string**

79.3. Modelo matemático

$$(cadena, codifica) \mapsto \left\{ \begin{array}{c} temp = "" \\ diccionario = \forall i \in alfabeto \\ diccionario[alfabeto[i]] = codificarlo[i] \\ \forall x, y \in \textbf{diccionario} : cadena[k] = x \rightarrow temp + y \end{array} \right\}$$

80. Problema 80

80.1. Análisis y Especificación

Desarrollar un algoritmo que decodifique una cadena de caracteres mediante una cadena de correspondencias de caracteres dada. La cadena de correspondencias tiene como el primer carácter el carácter equivalente para el carácter "a", el segundo carácter para la "b"y ası sucesivamente hasta la "z". No se tiene traducción para las mayúsculas ni para la "ñ".

Programa: Ej80.py

80.1.1. Objetos Conocidos

- Alfabeto
- Cadena codificada
- Cadena de decodificación

80.1.2. Objetos Desconocidos

- Cadena original

80.1.3. Relación entre objetos conocidos y desconocidos

Cada uno de los valores presentados en la cadena de codificación tomaran un nuevo valor respecto al alfabeto y el mensaje original (cadena) sera reemplazado por estas nuevas reglas.

80.2. Diseño y Prueba Conceptual

Entradas:

- cadena codificada $\in$ **string**
- cadena de decodificación $\in$ **string**

Salidas:

- Cadena original $\in$ **string**

80.3. Modelo matemático

$$(cadena, codifica) \mapsto \left\{ \begin{array}{c} temp = "" \\ diccionario = \forall i \in alfabeto \\ diccionario[codificarlo[i]] = alfabeto[i] \\ \forall x, y \in \textbf{diccionario} : cadena[k] = x \rightarrow temp + y \end{array} \right\}$$