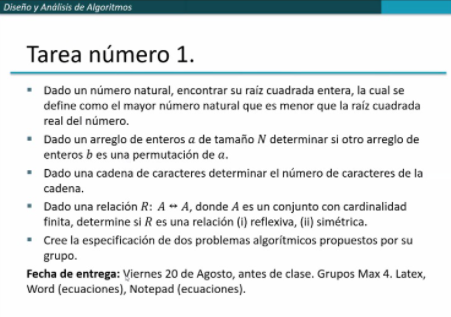
Juan Sebastián Hoyos

Andrés Arévalo Fajardo

Alejandro Salgado

María Camila Gómez



EJERCICIO 1:

|  |  |  |
| --- | --- | --- |
| E/S | NOMBRE | TIPO |
| ENTRADA | n | N (nat) |
| SALIDA | k | N (nat) |

|  |  |  |
| --- | --- | --- |
|  | DESCRIPCIÓN | EXPLICACIÓN |
| PRECONDICIÓN | Ninguna | No se pide ninguna condición particular sobre la entrada. Cualquier constante arbitraria A se puede asignar a n. |
| POSTCONDICIÓN | k será un número que cumpla | Se necesita especificar  con un predicado que la salida será k, si esta cumple la postcondición. |

Formalización

|  |  |
| --- | --- |
| PRECONDICIÓN |  |
| POSTCONDICIÓN |  |

EJERCICIO 2:

|  |  |  |  |
| --- | --- | --- | --- |
| E/S | NOMBRE | TIPO | DESCRIPCIÓN |
| ENTRADA | a[0, N)  b[0, N) | Array de números enteros  Array de números enteros | Arreglo de números enteros  Arreglo de números enteros |
| SALIDA | k | bool | Variable boolean que verifica la permutación entre los conjuntos |

|  |  |  |
| --- | --- | --- |
|  | DESCRIPCIÓN | EXPLICACIÓN |
| PRECONDICIÓN | Dos arreglos a[] y b[] del mismo tamaño | No se puede obtener la permutación de un conjunto con respecto a otro, si los dos conjuntos ni poseen la misma cardinalidad |
| POSTCONDICIÓN | todos los elementos en a[] deben estar contenidos en b[], cualesquiera sea la posición de los elementos de a[] en b[]. | Una permutación es la variación en el orden de un conjunto de partida, considerando que el orden inicial es una permutación resultante. |

Formalización

|  |  |
| --- | --- |
| PRECONDICIÓN | Logotipo  Descripción generada automáticamente |
| POSTCONDICIÓN |  |

EJERCICIO 3:

|  |  |  |  |
| --- | --- | --- | --- |
| E/S | NOMBRE | TIPO | DESCRIPCIÓN |
| ENTRADA | S[] | Arreglo de caracteres(String) | Cadena de caracteres |
| SALIDA | p | N(Entero natural) | Número que indica el número de caracteres presentados en la cadena |

|  |  |  |
| --- | --- | --- |
|  | DESCRIPCIÓN | EXPLICACIÓN |
| PRECONDICIÓN | Ninguna | El número de caracteres se puede calcular para cualquier cadena de caracteres |
| POSTCONDICIÓN | Para encontrar el número de caracteres de la cadena, se recorre desde el inicio y por cada carácter que se presente se suma 1 al valor a devolver | Se necesita especificar como se realiza la sumatoria con los elementos de la cadena |

Formalización

|  |  |
| --- | --- |
| PRECONDICIÓN | {Q:**true**} |
| POSTCONDICIÓN |  |

EJERCICIO 4:

|  |  |  |  |
| --- | --- | --- | --- |
| E/S | NOMBRE | TIPO | DESCRIPCIÓN |
| ENTRADA | LE(), n | Lista encadenada (Conjunto), Nodo(Relación) | Si el nodo adentro de la lista enlazada posee un atributo (this) es decir, una referencia así mismo y además la lista es doblemente enlazad, el nodo (n) se relaciona con el nodo (n+1, siguiente) y viceversa(de n+1 a n, anterior), entonces tenemos que las referencias de n con respecto a otros nodos de la lista enlazada es una relación. |
|  |  |  |  |
| SALIDA | b | Boolean | Retorna un booleano que indica si la relación es reflexiva y simétrica |

|  |  |  |
| --- | --- | --- |
|  | DESCRIPCIÓN | EXPLICACIÓN |
| PRECONDICIÓN | El conjunto A tiene cardinalidad finita | El numero de elementos del conjunto A es determinado y por lo tanto finito |
| POSTCONDICIÓN | Para determinar si se cumple la condición debemos revisar que la relación sea tanto reflexiva como simétrica.   1. Una relación es reflexiva cuando todo elemento del conjunto se relaciona consigo mismo. 2. Una relación es simétrica siempre y cuando para todo par de elementos del conjunto existe una relación bidireccional | Se necesita especificar con un predicado que todos los elementos del conjunto cumplan las condiciones de relación reflexiva y simetrica |

Formalización

|  |  |
| --- | --- |
| PRECONDICIÓN |  |
| POSTCONDICIÓN |  |

EJERCICIO 5:

Dado dos arreglos de igual longitud determina si los dos arreglos son completamente distintos.

|  |  |  |  |
| --- | --- | --- | --- |
| E/S | NOMBRE | TIPO | DESCRIPCIÓN |
| ENTRADA | a[0, N), b[0, N) | Array of int, array of int | Arreglos de enteros |
| SALIDA | c | Boolean | Retorna un booleano que indica si los arreglos son completamente distintos |

|  |  |
| --- | --- |
|  | DESCRIPCIÓN |
| PRECONDICIÓN | Ambos arreglos deben ser arreglos de enteros y tener la misma longitud. |
| POSTCONDICIÓN | Para ser cierto, a y b no deben compartir ningún numero en común. De lo contrario será falso. |

Formalización

|  |  |
| --- | --- |
| PRECONDICIÓN |  |
| POSTCONDICIÓN |  |

Dado un numero entero verificar si este sigue una tendencia de crecimiento, de parámetro máximo x, con respecto a los valores de una lista de enteros previamente analizados.

|  |  |  |  |
| --- | --- | --- | --- |
| E/S | NOMBRE | TIPO | DESCRIPCIÓN |
| ENTRADA | -L[0,N)  -p  -x | - Array de enteros  - Numero entero en los positivos  -Numero entero positivo | - Arreglo de enteros en donde se guardan elementos analizados por el algoritmo  - Numero entero a ser evaluado  -Parámetro entero que limita el máximo crecimiento de la secuencia de números |
| SALIDA | b | Boolean | Boolean que permite saber si el numero p sigue el crecimiento, dado el parámetro máximo x |

|  |  |
| --- | --- |
|  | DESCRIPCIÓN |
| PRECONDICIÓN | -Si L[] no esta vacía, entonces para aplicar el algoritmo la diferencia entre *p* y L[N-1] debe ser menor a 0, o ser mayor a *x*.  - Si L[] es vacía entonces no hay precondición |
| POSTCONDICIÓN | caso 1: Para ser correcto el algoritmo, la diferencia de p y L[N-1] debe estar entre 0 y el parámetro *x*  Caso 2: en caso de L[] ser vacía, el algoritmo es correcto si el primer elemento de L[] corresponde a p |

Formalización

|  |  |
| --- | --- |
| PRECONDICIÓN |  |
| POSTCONDICIÓN |  |