Inclusiones

Declaro las variables necesarias

Mientras Comando distinto de salir

Imprimo “Ingrese comando:”

Guardo lo ingresado en un string

Parseo Cacomando(string, string) //Aquí se valida que exista el comando

Según Comando Hacer

Caso crear: crear(string), Break.

Caso sumar: sumar(string), Break.

Caso multiplicar: multiplicar(string), Break.

Caso evaluar: evaluar(string, int), Break.

Caso esraiz: esraiz(), Break.

Caso mostrar: mostrar(), Break.

Caso guardar: guardar(), Break.

Caso recuperar: recuperar(), Break.

FinSegun

Fin Mientras

sumar(string){

Validarsumar()

Creo un nuevo polinomio vacío con el nombre especificado

Realizo la suma de los otros polinomios

Muestro el resultado

}

evaluar(string, int){

Validarmultiplicar()

Creo un nuevo polinomio vacío con el nombre especificado

Realizo la multiplicacion de los otros polinomios

Muestro el resultado

}

|  |  |
| --- | --- |
| IngresarComando | |
| Entrada: | Caracteres ingresados por teclado |
| Salida: | - |
| Método: | Creo un string nulo (strcrear) Cargo lo ingresado en un string dinámico (scan) Parsear el string ingresado |

|  |  |
| --- | --- |
| Parsear | |
| Entrada: | String de comandos |
| Salida: | Lista de Strings |
| Método: | Creo nuevo string  Recorro String de comandos  Mientras caracter sea distinto a fin de línea  Si carácter es distinto a espacio  Cargo String con carácter  Sino  Agrego caracter fin de línea  Guardo nuevo String en lista de String  Creo nuevo string  FinMientras |

|  |  |
| --- | --- |
| ValidarComando | |
| Entrada: | String |
| Salida: | TRUE, FALSE |
| Método: | Guardo valor FALSE  Tomo el primer enumerado de comandos  Si es igual al String de entrada  Guardo valor TRUE  Sino  Tomo el siguiente enumerado de comandos  Mientras haya enumerados de comandos Y valor guardado sea FALSE  Si el String de entrada es igual al enumerado de comandos  Guardo valor TRUE  Sino  Tomo el proximo enumerado  FinMientras  SI el valor guardado es TRUE  Devuelvo TRUE  Sino  Devuelvo FALSE |

|  |  |
| --- | --- |
| EsValidoNombre | |
| Entrada: | String cargado |
| Salida: | TRUE, FALSE |
| Método: | Recorro String  Mientras carácter sea distinto a carácter fin de línea y variable sea TRUE  Si carácter >= ‘a’ o caracter <= ‘z’  o carácter >= ‘A’ o caracter <= ‘Z’  o carácter >= ‘0’ o caracter <= ‘9’ entonces  Seteo variable en TRUE  Sino  Seteo variable en FALSE  FinMientras |

|  |  |
| --- | --- |
| EsValidoNumero | |
| Entrada: | String cargado |
| Salida: | TRUE, FALSE |
| Método: | Recorro String  Mientras carácter sea distinto a carácter fin de línea y variable sea TRUE  Si carácter es ‘-’ entonces tomo el siguiente caracter  Si carácter >= ‘0’ y carácter <= ‘9’ entonces  Seteo variable en TRUE  Sino  Seteo variable en FALSE  Sino  Si carácter >= ‘0’ y carácter <= ‘9’ entonces  Seteo variable en TRUE  Sino  Seteo variable en FALSE  FinMientras |

|  |  |
| --- | --- |
| ExistePolinomio | |
| Entrada: | Árbol de búsqueda binario cargado  Nombre de polinomio |
| Salida: | TRUE, FALSE |
| Método: | //Falta hacer |

|  |  |
| --- | --- |
| CrearPolinomio | |
| Entrada: | Lista de String cargada parámetros  ABB de polinomios por referencia |
| Salida: | - |
| Método: | Tomo primer valor de la lista y lo elimino de ella  Mientras no se llegue al final de la lista  ConvertirCharANumero  Creo termino  Agrego termino a lista  Sino  Muestro error    Creo polinomio con nombre y lista de términos  Agrego polinomio a ABB  Sino  Muestro error  FinSi |

|  |  |
| --- | --- |
| Multiplicar | |
| Entrada: | 2 Polinomios String |
| Salida: | Un nuevo Polinomio |
| Método: | Creo nuevo Polinomio y le asigno el nombre del string  Obtengo Lista de terminos del primer polinomio  Mientras que la Lista de terminos del primer polinomio no sea nula  Tomo Termino de la Lista de terminos del primer polinomio  Obtengo Lista de terminos del segundo polinomio  Mientras que la segunda lista no sea nula  Tomo Termino de la Lista de terminos del segundo polinomio   Multiplico termino lista 1 con termino lista 2  Agrego termino al Nuevo Polinomio  Sigo recorriendo lista 2  FinMientras  sigo recorriendo lista 1 FinMientras Muestro Polinomio final |

|  |  |
| --- | --- |
| Evaluar | |
| Entrada: | Polinomio Entero para evaluación |
| Salida: | Valor del polinomio dado el entero de entrada |
| Método: | Creo variable total  Mientras que la lista no sea nula  Obtengo termino  Multiplico entrada por si mismo tantas veces como el exponente  Multiplico el resultado por valor de la base del termino  Guardo valor más valor de variable total en variable total FinMientras Devuelvo valor total por pantalla |

|  |  |
| --- | --- |
| CantItemsEnLista | |
| Entrada: | Lista String cargada |
| Salida: | Int |
| Método: | Declaro variable cant  Recorro lista  Mientras haya ítems en lista  Sumo uno a variable cant  FinMientras |

|  |  |
| --- | --- |
| EsRaiz | |
| Entrada: | Lista String cargada  Entero para calcular |
| Salida: | TRUE, FALSE |
| Método: | Creo variable total  Creo variable Boolean en FALSE  Mientras que la lista no sea nula  Obtengo termino  Multiplico entrada por si mismo tantas veces como el exponente  Multiplico el resultado por valor de la base del termino  Guardo valor más valor de variable total en variable total  FinMientras  Si total es igual a 0  Guardo variable en TRUE  Retorno variable Boolean |

|  |  |
| --- | --- |
| Sumar | |
| Entrada: | Dos polinomios a ser sumados  Polinomio vacio por referencia |
| Salida: | - |
| Método: | Obtengo términos de primer polinomio  Obtengo términos de segundo polinomio  Creo variable Booleana en FALSE  Recorro primer lista  Mientras lista1 no sea vacia  Obtengo primer exponente de primer termino de primer lista  Recorro segunda lista  Mientras lista2 no sea vacía y variable Booleana sea FALSE  Obtengo primer exponente de primer termino de segunda lista  Si ambos exponentes son iguales  Seteo variable en TRUE  Creo nuevo termino con base igual a suma de ambas bases y exponente  Agrego término a lista de términos  FinSi  FinMientras  FinMientras  Agrego lista de términos a Polinomio vacio |

|  |  |
| --- | --- |
| ConvertirCharANumero | |
| Entrada: | String cargado |
| Salida: | Int |
| Método: | Recorro string  Mientras carácter sea distinto a fin de línea  Si carácter es ‘-’ tomo siguiente caracter  EnCasoQueSea  1: valor = -1  2: valor = -2  3: valor = -3  4: valor = -4  5: valor = -5  6: valor = -6  7: valor = -7  8: valor = -8  9: valor = -9  OTRO: error  FinEnCasoQueSea  Si no  EnCasoQueSea  1: valor = 1  2: valor = 2  3: valor = 3  4: valor = 4  5: valor = 5  6: valor = 6  7: valor = 7  8: valor = 8  9: valor = 9  OTRO: error  FinEnCasoQueSea |

MÓDULO MAIN

Defino texto\_entrada y guardo lo ingresado por teclado hasta pulsar Enter

Parsear texto\_entrada a ListaStrings

Defino variable comando de tipo string con el valor del primer elemento de ListaString

Si ValidarComando de variable comando Devuelve TRUE

. Según DarComando de variable comando sea:

. . crear: Tomo el segundo valor de ListaString

. . Si ValidarNombre devuelve TRUE

. . . Si ExistePolinomio devuelve FALSE

. . . . Defino variable cant con CantItemsEnLista y le resto 2

. . . . Tomo tercer valor de la lista

. . . . Guardo valor TRUE

. . . . Mientras que haya elementos en la lista Y valor sea TRUE

. . . . . Si ValidarNumero devuelve FALSE

. . . . . Guardo Valor FALSE

. . . . . FinSi

. . . . . Tomo siguiente valor de la lista

. . . . FinMientras

. . . . Si Valor es TRUE

. . . . . Tomo tercer valor de la lista

. . . . . Convierto CharAInt

. . . . . Si cant es mayor a 1 y valor diferente de 0

. . . . . CrearPolinomio

. . . . . FinSi

. . . . . Sino

. . . . . MostrarError Primer coeficiente 0

. . . . . FinSino

. . . . FinSi

. . . . Sino

. . . . MostrarError Coeficiente invalido

. . . . FinSino

. . . FinSi

. . . Sino

. . . MostrarError polinomio ya existe

. . . FinSino

. . FinSi

. . Sino

. . MostrarError Nombre polinomio Invalido

. . FinSino

. . break;

. . Modificar:

. . ETC:

. FinSegun

. . FinSi

Sino MostrarError de comando no valido

FinSino