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  Lista de String vacía por referencia |
| Salida: | - |
| 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: | Lista de String cargada con el comando y parametros |
| Salida: | Int (1-9 o 0 en caso de error) |
| Método: | Tomo primer valor de la lista y lo elimino de ella.  EnCasoQueSea  crear: 1  sumar: 2  multiplicar: 3  evaluar: 4  esraiz: 5  mostrar: 6  guardar: 7  recuperar: 8  salir: 9  OTRO: 0 FinEnCasoQueSea |

|  |  |
| --- | --- |
| 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’ o carácter <= ‘9’ entonces  Seteo variable en TRUE  Sino  Seteo variable en FALSE  Sino  Si carácter >= ‘0’ o 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  Si EsValidoNombre es TRUE y ExistePolinomio es FALSE  Mientras no se llegue al final de la lista  Si EsValidoNumero es TRUE  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 Listas de Terminos Polinomio pasado por referencia |
| Salida: | - |
| Método: | Mientras que la primer lista no sea nula  Obtengo termino lista 1  Mientras que la segunda lista no sea nula  Obtengo termino lista 2  Multiplico termino lista 1 con termino lista 2  Agrego termino al Polinomio por referencia  Sigo recorriendo lista 2  FinMientras  sigo recorriendo lista 1 FinMientras Muestro Polinomio final |

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

|  |  |
| --- | --- |
| Crear | |
| Entrada: | String Lista de Terminos Polinomio pasado por referencia |
| Salida: | - |
| Método: | Agrego el string como nombre del Polinomio. Mientras que la lista no sea nula  Agrego el termino al principio de la lista de terminos del polinomio  FinMientras |

|  |  |
| --- | --- |
| 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: