

Proyecto N°1

Programación en Lenguaje C

Documentación

Massetti, Matías

Privitello, Gaetano

Vazquez, Santiago

Proyecto:

El proyecto presente es parte de la asignatura “Organización de Computadoras”, dictada en la Universidad Nacional del Sur, en la ciudad de Bahía Blanca.

El mismo consiste en implementar un programa en el lenguaje de programación C, que evalúe expresiones aritméticas expresadas en notación de pre orden.

Con este objetivo en mente, se debe implementar lo siguiente:

- Un TDA Lista, utilizando memoria dinámica mediante las funciones de librería `void free(*void ptr)` y `void malloc(*size_t size)`.
- Un TDA Pila para almacenar strings de caracteres.
- Un programa que evalúe expresiones aritméticas por entrada estándar, utilizando los TDA anteriormente implementados.

Programa principal:

El programa principal debe evaluar expresiones aritméticas ingresadas en preorden y devolver el valor entero de la resolución.

Para el ingreso de datos al sistema de manera correcta solamente admitimos números enteros positivos.

Luego de procesada la expresión y evaluada, se mostrará por pantalla el resultado.

TDA Lista

La lista implementada es simplemente enlazada cuyos elementos contenidos dentro de cada celda son arreglos de enteros y su implementación es realizada con punteros.

Métodos TDA Lista:

- **lista_t lista_crear()**
Crea una lista vacía y la retorna.
- **int lista_insertar(lista_t lista, unsigned int pos, int elem)**
Inserta el elemento elem en la posición pos de la lista. Sobrescribe el valor existente en la posición indicada, o agrega un nuevo elemento si la posición coincide con la cantidad de elementos. Si procede exitosamente retorna verdadero. En otro caso retorna falso.
- **int lista_eliminar(lista_t lista, unsigned int pos)**
Elimina el elemento en la posición pos. Reacomoda la lista adecuadamente al eliminar en posiciones intermedias. Retorna verdadero si procede con éxito, falso en caso contrario. Si la posición no es válida, aborta con exit status LST_POS_INV.
- **int lista_cantidad(lista_t lista)**
Retorna la cantidad de elementos de la lista. Si la lista no está inicializada, el programa aborta con exit status LST_NO_INI.
- **int lista_obtener(lista_t lista, unsigned int pos)**
Retorna el elemento en la posición pos de la lista. Si la posición no es válida, aborta con exit status LST_POS_INV.
- **int lista_adjuntar(lista_t lista, int elem)**
Inserta el elemento elem en la última posición de la lista. Si la lista no está inicializada, aborta con exit status LST_NO_INI.
- **int lista_destruir(lista_t* lista)**
Libera la memoria ocupada por la lista y le asigna NULL. Retorna verdadero en caso de éxito. Si la lista no está inicializada, aborta con exit status LST_NO_INI.

Métodos auxiliares TDA Lista:

- **void reacomodarLista(lista_t lista,int indice,celda_t * celda)**
Luego de eliminar una posición de la lista se utiliza este método para acomodar las posiciones con sus respectivos elementos en la lista.
- **void destruirCelda(celda_t* celda)**
Elimina una celda y libera el espacio asignado en memoria.
- **celda_t* obtenerCelda(lista_t lista,int nro_celda)**
Retorna la celda creada en la posición número celda.

TDA Pila

La pila está implementada con una lista dinámica, simplemente enlazada sin centinela.

Métodos TDA Pila:

- **pila_t pila_crear()**
Retorna una pila nueva vacía.
- **char* tope(pila_t pila)**
Retorna el string que se encuentra en el tope de la pila. Si la pila se encuentre vacía, aborta su ejecución con exit status PIL_VACIA.
- **char* desapilar(pila_t* pila)**
Elimina el string que se encuentra en el tope de la pila y lo retorna. Si la pila se encuentra vacía, aborta su ejecución con exit status PIL_VACIA.
- **int apilar(pila_t* pila, char* str)**
Inserta el string str en el tope de la pila. Retorna verdadero si la inserción fue exitosa, falso en caso contrario. Si la pila no se encuentra inicializada, finaliza la ejecución con exit status PIL_NO_INI.
- **pila_vacia(pila_t pila)**
Retorna verdadero si la pila está vacía, falso en caso contrario. Si la pila no se encuentra inicializada, finaliza la ejecución con exit status PIL_NO_INI.