Instituto Politécnico Nacional

Programa de Licenciatura en Matemática Algorítmica

Trabajo Colaborativo

Segunda Practica de Conjuntos

Autores:
Aguilar Romero Angel Uriel
Gordiano Rojas Carlos Vidal
Garcia Junquera Luis Eduardo
Martínez Huicochea Victor Hugo

Profesor:
Miguel Angel Valencia Bucio

Teoría de Conjuntos

México, Ciudad de México
November 6, 2023

Práctica 2

Objetivo:El objetivo principal de este programa es proporcionar a los usuarios una herramienta para calcular y analizar funciones matemáticas complejas. El programa permite:
Evaluar funciones recursivas y operaciones matemáticas.
Utilizar resultados previamente calculados para funciones personalizadas.

1 Marco teorico

El programa se basa en una lista enlazada para almacenar resultados intermedios y utiliza una pila para realizar cálculos de manera recursiva. Los usuarios pueden definir expresiones matemáticas complejas, incluyendo la capacidad de realizar sumatorias y aplicar condicionales, para calcular resultados de manera eficiente.

2 Explicación del Programa

A continuación, se explican los pasos clave del programa:

- **Inicio y Configuración:**: El programa comienza por incluir las bibliotecas necesarias y define una estructura llamada "Datos" que parece estar definida en un archivo llamado "Structs.h".
  El programa comienza solicitando al usuario que proporcione una fórmula matemática, que puede incluir variables "i" y "N" junto con operadores matemáticos.

- **Entrada de datos**: El usuario debe proporcionar la cantidad de elementos iniciales (incluyendo 0) que se utilizarán en los cálculos. Esto se almacena en la variable "inicio".

  El usuario también debe especificar el valor deseado de "N" para el cálculo. Esto se almacena en la variable "N".

- **Inicialización de estructuras de datos:**: El programa inicializa una lista enlazada llamada "resultados" para almacenar resultados intermedios. Cada elemento de esta lista contiene un valor numérico y su posición correspondiente

  Se inicia una variable "busqueda" para almacenar el resultado final que se obtendrá para el valor de "N".

- **Cálculos iterativos:**: El programa realiza una serie de iteraciones desde 0 hasta 10 (por ejemplo) utilizando una variable "i" como contador. Esto permite calcular resultados para diferentes valores de "i".

  Para cada iteración, el programa realiza lo siguiente:
  a. Copia la fórmula proporcionada por el usuario en una cadena auxiliar llamada "NuevaFrase".
  b. Utiliza la función "Convertidor" para reemplazar las variables "i" y "N" en "NuevaFrase" por los valores actuales de "i" y "N", respectivamente.
  c. Llama a la función "Calcular" para evaluar la expresión matemática modificada en "NuevaFrase". Esta función realiza los cálculos y devuelve un resultado.
  d. Agrega el resultado junto con la posición "i" a la lista "resultados" utilizando la función "agregar".

- **Búsqueda del Resultado para "N"**: Después de realizar los cálculos para todos los valores de "i", el programa busca el resultado correspondiente a "N" en la lista "resultados". Esto se hace llamando a la función "buscar" con "resultados" y "N" como argumentos.

El resultado encontrado se almacena en la variable "busqueda" para su posterior visualización.

- **Presentación del resultado**: Finalmente, el programa muestra el resultado calculado para el valor de "N" proporcionado por el usuario. Esto se hace mediante una impresión en pantalla.

3 Pseudocodigo

```
Estructura de Datos:
- Datos: estructura para almacenar valores y sus posiciones en una lista enlazada.
- Nodo: estructura para implementar una pila de valores.

Función agregar(Datos **p, valor, posición):
    Crear un nuevo elemento Datos con valor y posición.
    Si la lista está vacía:
        Apuntar *p al nuevo elemento.
    De lo contrario:
        Recorrer la lista hasta el final.
        Agregar el nuevo elemento al final de la lista.

Función buscar(Datos *p, posición):
    Iniciar un puntero auxiliar en p.
    Mientras el puntero auxiliar no sea nulo y la posición no coincida:
        Avanzar al siguiente elemento en la lista.
    Devolver el elemento encontrado.

Función poner(Nodo **p, valor):
    Crear un nuevo elemento Nodo con el valor y apuntar al siguiente elemento.
    Actualizar *p para que apunte al nuevo elemento.

Función sacar(Nodo **p):
    Iniciar un puntero auxiliar en *p.
    Actualizar *p para que apunte al siguiente elemento.
    Liberar la memoria del elemento auxiliar.

Función vacio(Nodo *p):
    Devolver verdadero si p es nulo, de lo contrario, falso.

Función Convertidor(Frase, i, N, resultados):
    Crear una cadena Nueva vacía.
    Crear un número de cadena número.
    Inicializar una variable op en 0.
    Inicializar una variable c en 0.

    Definir delimitador como un espacio en blanco.

    Dividir Frase en elementos utilizando el delimitador.

    Para cada elemento en los elementos:
        Si el elemento contiene 'i':
            Establecer op en 1.
        Si el elemento contiene 'N':
            Establecer op en 2.
```

```
        Según el valor de op:
            Caso 1: Reemplazar 'i' con el valor actual de i en el elemento.
            Caso 2: Reemplazar 'N' con el valor actual de N en el elemento.
            Por defecto: Agregar el elemento sin cambios a Nueva.

        Restablecer op a 0.
        Agregar un espacio en blanco al final de Nueva.

    Calcular el resultado c llamando a la función Calcular con Nueva y resultados.
    Devolver c.

Función Calcular(Frase, resultados):
    Crear una pila Numeros vacía.
    Crear un conjunto de operadores permitidos (por ejemplo, +, -, *, %, ^, f).

    Dividir Frase en elementos utilizando un delimitador.

    Para cada elemento en los elementos:
        Si el elemento es un número o una variable:
            Convertir el elemento en un valor numérico y ponerlo en la pila Numeros.
        Si el elemento es un operador:
            Realizar las operaciones adecuadas según el operador y los valores en la pila Numeros.
            Colocar el resultado en la pila Numeros.

    Devolver el valor final de la pila Numeros.

Función principal:
    Inicializar una lista resultados como nulo.
    Inicializar un puntero busqueda como nulo.
    Inicializar N, inicio e i en 0.
    Inicializar una cadena ParaFrase con la fórmula deseada.

    Imprimir un mensaje de bienvenida y solicitar la fórmula al usuario.

    Solicitar al usuario la cantidad de elementos iniciales (incluyendo 0).
    Solicitar al usuario el valor deseado de N.

    Para i en el rango de inicio a 10:
        Crear una copia de ParaFrase en una cadena NuevaFrase.
        Calcular c llamando a Convertidor con NuevaFrase, i, N y resultados.
        Agregar c a la lista resultados utilizando la función agregar.

    Buscar el resultado para N en la lista resultados y almacenarlo en busqueda.

    Imprimir el resultado para N.
```

4 Pruebas

A continuación mostramos algunas pruebas que realizamos al programa para poder verificar que su ejecución procede sin problemas, esto meadiante algunas relaciones únicas cuyas soluciónes puedan representar un problema para nuestro programa. Algunos de estos casos son:

- **Sucesión de fibonacci**
  La sucesión de fibonacci consiste en sumar los ultimos dos elementos de la sucesión comenzando con f(0)=1

y f(1)=1
Datos de entrada:
La función de entrada es f(i)=f(i-1)+f(i-2)
Elementos iniciales: 2
N=30
base:
0=1
1=1

```
Bienvenido a la funcion recursiva, por favor digite su funcion:
f(i)=f(i-1)+f(i-2)
Por favor, digite cuantos elementos iniciales tendra (el 0 cuenta)
2
Por favor, digite hasta que elemento desea llegar
 N=30
Por favor, digite la base
 0=1
Por favor, digite la base
 1=1
En N=30, el valor obtenido fue: 1346269

------------
Process exit
Presione una
```

- **Sucesión de unos y ceros**
  Esta sucesión usa la operación modulo 2 para hacer que los terminos sean 0 si i es par y 1 si es impar
  Datos de entrada:
  La función de entrada es f(i)=i%2
  Elementos iniciales: 1
  N=99
  base:
  0=0

```
Bienvenido a la funcion recursiva, por favor digite su funcion:
f(i)=i%2
Por favor, digite cuantos elementos iniciales tendra (el 0 cuenta)
1
Por favor, digite hasta que elemento desea llegar
 N=99
Por favor, digite la base
 0=0
En N=99, el valor obtenido fue: 1

--------------------------------
Process exited after 17.08 seconds with return value 0
Presione una tecla para continuar . . .
```

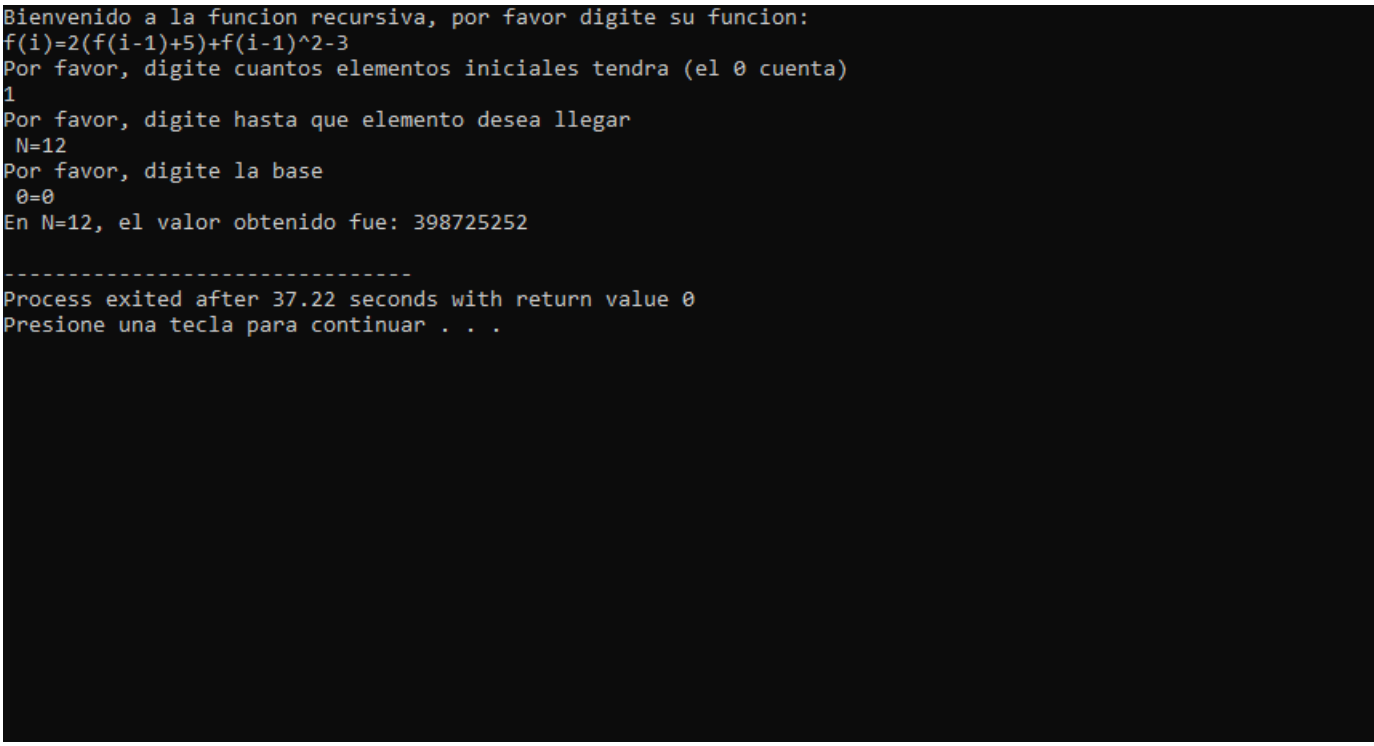
- **Sucesión de suma**
  Esta sucesión empieza en 13 y genera cada termino apartir del anterior y suma 5
  Datos de entrada:
  La función de entrada es f(i)=f(i-1)+5
  Elementos iniciales: 1
  N=45
  base:
  0=13

```
Bienvenido a la funcion recursiva, por favor digite su funcion:
f(i)=f(i-1)+5
Por favor, digite cuantos elementos iniciales tendra (el 0 cuenta)
1
Por favor, digite hasta que elemento desea llegar
 N=45
Por favor, digite la base
 0=13
En N=45, el valor obtenido fue: 238

--------------------------------
Process exited after 13.84 seconds with return value 0
Presione una tecla para continuar . . .
```

- **Sucesión con potencia**
  Esta sucesión la definimos apartir de $f(i) = i^2 + i$
  Datos de entrada:
  La función de entrada es $f(i) = i^2 + i$
  Elementos iniciales: 1
  N=26
  base:
  0=0

```
Bienvenido a la funcion recursiva, por favor digite su funcion:
f(i)=i^2+i
Por favor, digite cuantos elementos iniciales tendra (el 0 cuenta)
1
Por favor, digite hasta que elemento desea llegar
 N=26
Por favor, digite la base
 0=0
En N=26, el valor obtenido fue: 702

--------------------------------
Process exited after 24.79 seconds with return value 0
Presione una tecla para continuar . . .
```

- **Sucesión con todas las operaciones**
  Esta sucesión usamos todas las operaciones
  Datos de entrada:
  La función de entrada es $f(i) = 2(f(i-1) + 5) + f(i-1)^2 - 3$
  Elementos iniciales: 1
  N=12
  base:
  0=0


Bienvenido a la funcion recursiva, por favor digite su funcion:
f(i)=2(f(i-1)+5)+f(i-1)^2-3
Por favor, digite cuantos elementos iniciales tendra (el 0 cuenta)
1
Por favor, digite hasta que elemento desea llegar
N=12
Por favor, digite la base
0=0
En N=12, el valor obtenido fue: 398725252

Process exited after 37.22 seconds with return value 0
Presione una tecla para continuar . . .