



INSTITUTO POLITÉCNICO NACIONAL

PROGRAMA DE LICENCIATURA EN MATEMÁTICA ALGORÍTMICA

TRABAJO COLABORATIVO

Cuarta 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
December 11, 2023

Práctica 4

Objetivo: El objetivo principal de la práctica 4 es desarrollar un programa en lenguaje C que, tomando como entrada dos números enteros n y m (con $n, m > 1$), calcule la raíz de índice m del número n . Para la realización del programa, nos apoyaremos en la biblioteca GMP. Finalmente, obtendremos 20,000 dígitos de la raíz, los cuales serán impresos en un archivo de texto con el nombre 'eq1_pract4.txt'

1 Marco teorico

La práctica 4 tiene como objetivo principal desarrollar un programa en lenguaje C que utilice la biblioteca GMP (GNU Multiple Precision) para calcular la raíz de índice m de un número entero n . La biblioteca GMP proporciona funciones para realizar operaciones aritméticas con precisión arbitraria, lo que es esencial cuando se trabaja con números grandes o se requiere una alta precisión en los cálculos.

GMP es una biblioteca de software libre diseñada para realizar operaciones aritméticas con precisión arbitraria. El cálculo de la raíz de índice m de un número n se realiza mediante n Como m deben ser mayores que 1 para garantizar que la raíz exista y sea única.

La práctica requiere generar 20,000 dígitos de la raíz calculada y guardarlos en un archivo de texto llamado 'eq1_pract4.txt'. Para lograr esto, se utilizan funciones de GMP para convertir el resultado en una cadena de caracteres y luego escribir esa cadena en un archivo.

En conclusion la práctica combina la manipulación de grandes números y la necesidad de alta precisión en el cálculo de raíces mediante el uso de la biblioteca GMP. La capacidad de trabajar con precisión arbitraria

Para el calculo de raíces exactas nos basamos en el principio de convergencia aplicado a series recursivas. Dicho algoritmo trata de replicar la obtención de números irracionales obtenibles a travez de raíces mediante la recurción iterativa de la suseción:

$$x_{n+1} = \frac{1}{k}(x_n(k-1) + \frac{N}{x_n^{k-1}})$$

Dicha sucesión es convergente. Y como tal la convergencia de esta serie será hacia algún límite, el cual podemos encontrar de la siguiente manera:

$$x_{n+1} = \frac{1}{k}(x_n(k-1) + \frac{N}{x_n^{k-1}})$$

Por definición de límite, tanto x_{n+1} y x_n convergen hacia algún ℓ , siendo así que:

$$\ell = \frac{1}{k}(\ell(k-1) + \frac{N}{\ell^{k-1}})$$

$$k\ell = \ell(k-1) + \frac{N}{\ell^{k-1}}$$

$$k\ell - \ell(k-1) = \frac{N}{\ell^{k-1}}$$

$$\ell = \frac{N}{\ell^{k-1}}$$

$$\ell \ell^{k-1} = N$$

$$\ell^k = N$$

$$\ell = \sqrt[k]{N}$$

Así pues, tenemos que converge dicha serie a la raíz k -ésima de N , para todo $N, k \in \mathbb{N}$

2 Explicación del programa

Este programa en C se encarga de calcular la raíz k -ésima de un número n , utilizando el método antes mencionado. La alta precisión en los cálculos se asegura gracias a la utilización de la biblioteca GMP (GNU Multiple Precision). A continuación, se ofrece una descripción detallada de su funcionalidad:

2.1 Inicialización y configuración:

En las primeras líneas, se inicializan las variables y se crea un archivo denominado 'eq1_prac4.txt' para almacenar el resultado. Se configuran variables especiales de la biblioteca GMP, permitiendo manejar la precisión necesaria y estableciendo condiciones de parada específicas.

2.2 Entrada del usuario:

Se solicita al usuario ingresar el radicando n y el índice k de la raíz.

2.3 Verificación de entrada :

Se realiza una validación para asegurar que los valores ingresados para k y n sean mayores que 1. En caso contrario, el programa emite un mensaje de error y se detiene.

2.4 Establecimiento de condición de parada:

Se define una condición de parada basada en la diferencia entre dos iteraciones consecutivas. Esto garantiza que el programa finalice cuando la diferencia sea menor que 110^{20000} .

2.5 Serie Recursiva:

(La Implementación de la serie recursiva nos permite encontrar las raíces a partir de números naturales.)

Se implementa dicho método para calcular iterativamente la raíz k -ésima de n .

En cada iteración, se actualiza el valor de x_1 utilizando la fórmula antes mencionada.

2.6 Impresión del archivo:

El resultado se convierte a una cadena de caracteres y se almacena en el archivo 'eq1_prac4.txt'.

2.7 Impresión del resultado :

Además, se imprime el resultado en la consola para facilitar la revisión.

Se libera la memoria utilizada por las variables de la biblioteca GMP.

El programa culmina su ejecución y emite un mensaje indicando su finalización.

En síntesis, el programa emplea el método de la secante para calcular la raíz de k de n , asegurando una precisión destacada mediante la biblioteca GMP. Los resultados son guardados en un archivo y se presentan en la consola. La condición de parada garantiza que el programa termine cuando se logra la precisión deseada.

3 Pseudocódigo

PROCEDIMIENTO EncontrarRaiz(Funcion f, REAL tolerancia, ENTERO N, ENTERO k):

Parámetros:

- f: Función iterativa para la cual se busca la raíz.

- tolerancia: Tolerancia para la convergencia.

- N: Número dentro de la raíz - k: Índice de la raíz

Variables:

REAL x_actual , x_anterior, diferencia;

Inicialización de variables:

x_anterior = N

diferencia = INFINITO

Bucle principal:

PARA cada iteracion MIENTRAS diferencia > tolerancia HACER:

Calcular la nueva iteración para el punto actual:

$x_{\text{actual}} = f(x_{\text{anterior}}, N, k)$

Calcular la diferencia entre las aproximaciones sucesivas:

$\text{diferencia} = \text{ABS}(x_{\text{anterior}} - x_{\text{actual}})$

Verificar convergencia:

SI diferencia < tolerancia ENTONCES

RETORNAR x_{actual} // Aproximación suficientemente
cercana a la raíz

FIN SI

Actualizar puntos para la siguiente iteración:

$x_{\text{anterior}} = x_{\text{actual}}$

FIN PARA

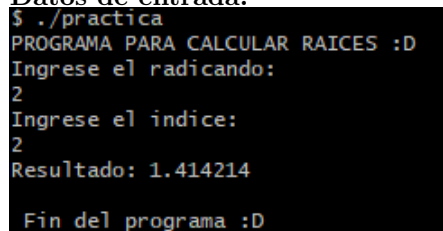
FIN DEL PROCEDIMIENTO

4 Pruebas

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

- **Raíz cuadrada de 2:**

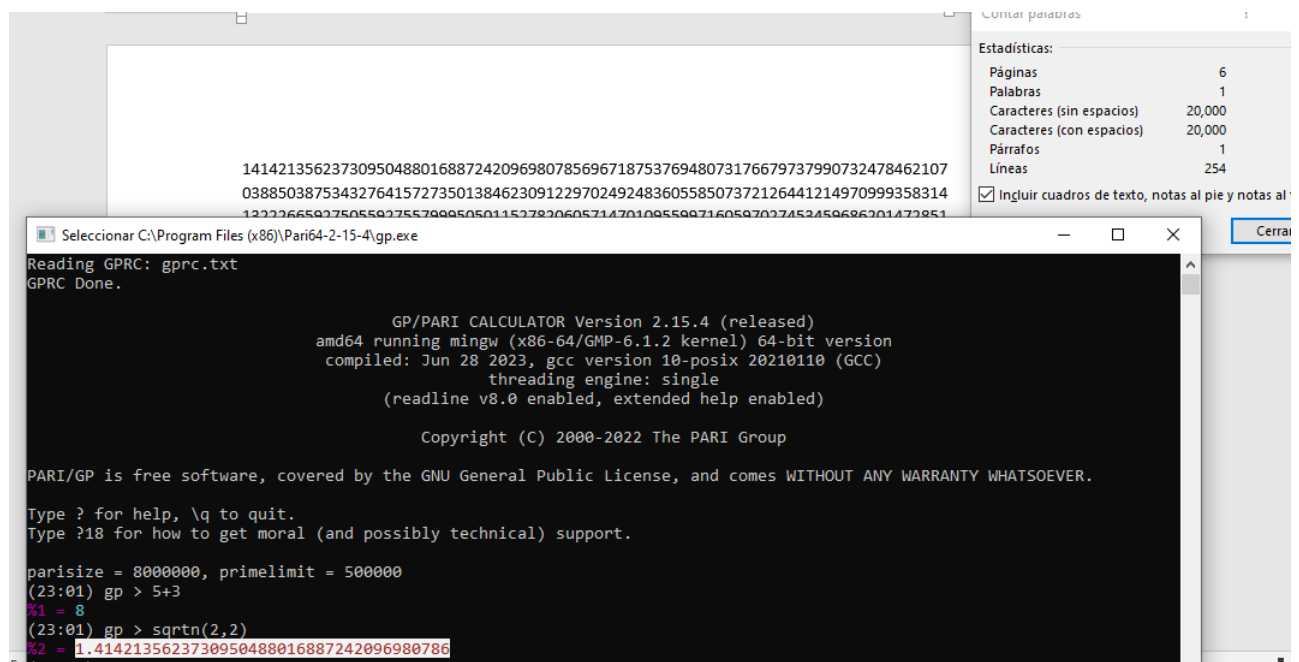
Datos de entrada:



```
$ ./practica
PROGRAMA PARA CALCULAR RAICES :D
Ingrese el radicando:
2
Ingrese el indice:
2
Resultado: 1.414214
Fin del programa :D
```

Datos de salida:

Para comprobar la cantidad de cifras usamos word, y para verificar algunos terminos de la raíz usamos la calculadora Pari/GP



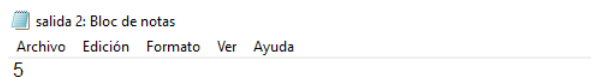
- Raíz cuarta de 625:

Datos de entrada:

```
$ ./practica
PROGRAMA PARA CALCULAR RAICES :D
Ingrese el radicando:
625
Ingrese el indice:
4
Resultado: 5.000000
Fin del programa :D
```

Datos de salida:

En este caso el objeto es mostrar que el programa puede detenerse si la raíz es exacta.



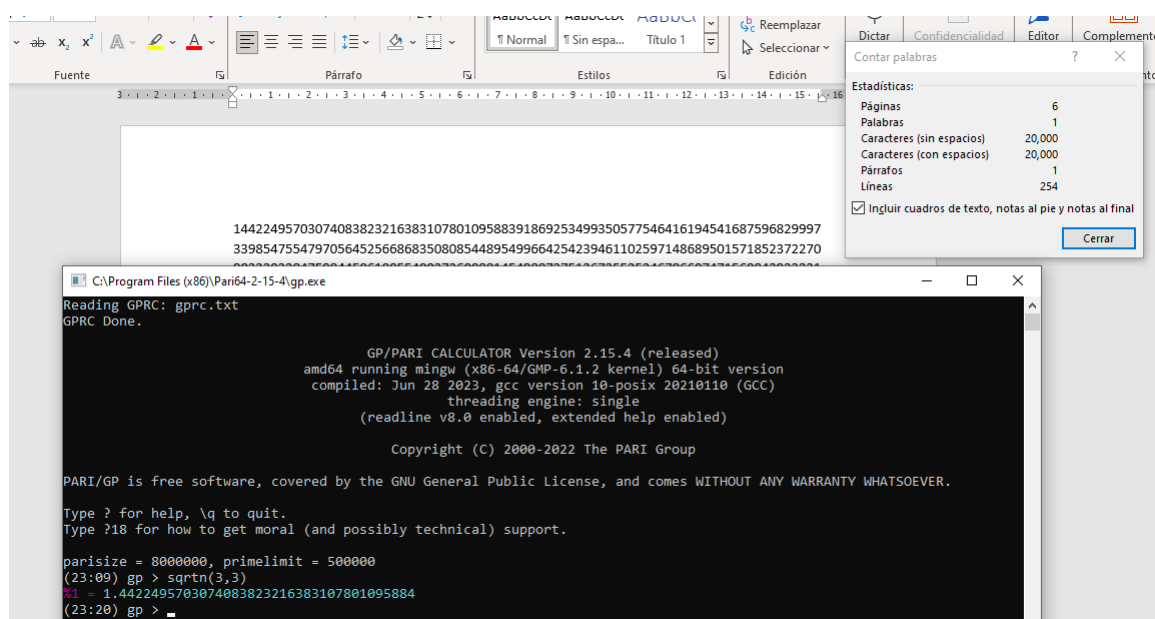
- Raíz tercera de 3:

Datos de entrada:

```
$ ./practica
PROGRAMA PARA CALCULAR RAICES :D
Ingrese el radicando:
3
Ingrese el indice:
3
Resultado: 1.442250
Fin del programa :D
```

Datos de salida:

Para esta prueba, volvimos a comprobar las cifras usando Word y la calculadora Pari/GP.



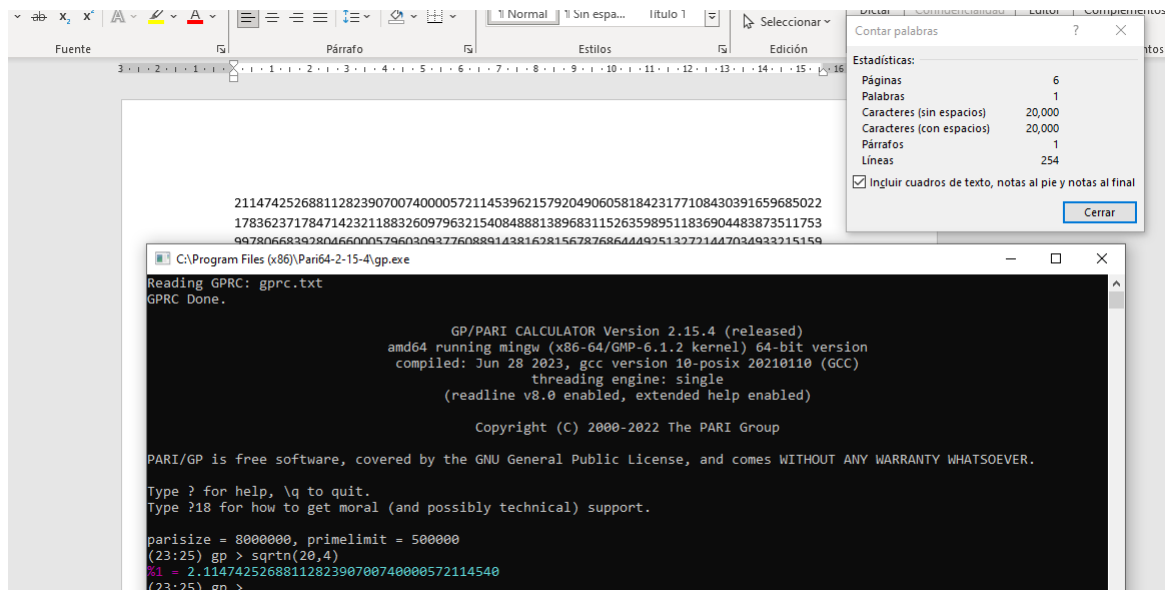
- Raíz cuarta 20:

Datos de entrada:

```
$ ./practica
PROGRAMA PARA CALCULAR RAICES :D
Ingrese el radicando:
20
Ingrese el indice:
4
Resultado: 2.114743
Fin del programa :D
```

Datos de salida:

Para esta prueba, volvimos a comprobar las cifras usando Word y la calculadora Pari/GP.



- Raíz decima de 325:

Datos de entrada:

```
$ ./practica
PROGRAMA PARA CALCULAR RAICES :D
Ingrese el radicando:
325
Ingrese el indice:
10
Resultado: 1.783152

Fin del programa :D
```

Datos de salida:

Para esta prueba, volvimos a comprobar las cifras usando Word y la calculadora Pari/GP.

The image shows a Microsoft Word document with a long decimal string: 1783151881253044733475511249025272659582932088973007253650468004041134031525575354721549191003538094509330376412723345459292221966240731294995134313628389432363398173743334378336315674895433932324964165025616681442139863944844877572324. Overlaid on the Word window is the Pari/GP calculator interface. The calculator window title is 'C:\Program Files (x86)\Pari64-2-15-4\gp.exe'. It shows the version 'GP/PARI CALCULATOR Version 2.15.4 (released)', the kernel 'amd64 running mingw (x86-64/GMP-6.1.2 kernel) 64-bit version', the compilation date 'compiled: Jun 28 2023, gcc version 10-posix 20210110 (GCC)', and the threading engine 'threading engine: single'. It also displays the copyright 'Copyright (C) 2000-2022 The PARI Group' and the license 'PARI/GP is free software, covered by the GNU General Public License, and comes WITHOUT ANY WARRANTY WHATSOEVER.' The calculator prompt shows the command '(23:26) gp > sqtrn(325,10)' and the result '%1 = 1.7831518812530447334755112490252726596'.