TEC | Tecnológico de Costa Rica

Escuela de programación
Ingenieria en Computación
IC1803 - Taller de programación

Numeradores y Denominadores para un cociente dado

-Proyecto de Taller-

Oscar Daniel Rojas Rodriguez
o.rojas.1@estudiantec.cr
2024124324

Alajuela, Costa Rica
Abril 2024

Índice general

Capítulo 1

Resumen Ejecutivo

El proyecto consta de un problema el cual se basa en encontrar fracciones con numerador y denominador que resuelven a un cociente C dado como entrada. Los numeradores y denominadores deben de ser de 5 digitos, no mas, no menos, y hay una entrada R que define cuantas veces se pueden repetir los digitos en ambos numeros por igual. Ademas se agruega un T que sirve como entrada reguladora de los C y R, es decir, regula la cantidad de C y R que vallan a entrar, por ejemplo, si T es 2, entonces entran dos C y dos R.

La solucion dada para este proyecto se basa en una serie de funciones que de manera iterativa buscan un a y b como denominador y numerador, utilizando la regla de divisibilidad se llega a la conclusion de que b como numerador es igual al denominador multiplicado por el cociente que es la entrada C dada.

Para resolver el problema impuesto por la R sobre la repeticion de digitos en la division total, se crearon dos funciones, una que separa los digitos de un numero en una lista, utilizando modulo 10 y dividiendo el numero original entre 10 para poder obtener el siguiente digito, y otra que se va a utilizar para contar cuantas veces se repite un mismo numero en una lista y devuelve el maximo de repeticiones de la lista anterior, despues se vuelve a la funcion iterativa y compara este maximo de repeticiones con el número R para definir si hay o no más repeticiones de parte de algun digito que lo permitido. La entrada T se utiliza de manera que mientras T exista el ciclo de C y R se va a repetir una y otra vez hasta que T sea igual a cero, teniendo en cuenta que cada vez que se repite el ciclo, se le resta uno a T para que llegue a cero en algun momento.

Los casos de prueba utilizados fueron diez en total y todos funcionaron de manera correcta, la mayoria de estas pruebas salieron a la excelencia incluso usando T como el numero cien con numeros aleatorios dentro de el rango permitido.

Capítulo 2

Proyecto

2.1. Introduccion

Este documento se basa en la descripcion de un proyecto por partes, asi como lo son el marco teórico de el proyecto, que sobre todo se trata de extender un poco de conocimiento sobre el lenguaje utilizado, en este caso python, la solucion de este mismo, es decir la solucion dada hacia el problema dado en el proyecto, que en este caso son una serie de funciones, mayoitariamente iterativas, los resultados de pruebas dadas y las conclusiones que se obtienen del proyecto, ademas de los aprendizajes de este.

El proyecto consistio en buscar valores a y b como denominador y numerador hacia un cociente C dado, ambos teniendo cinco digitos, de lo anterior se concluye que b = a * c, y con esto hay paso para crear funciones iterativas que busquen los resultados de esta posible fraccion, retornando "b/a = c". Para R se utilizan otras funciones de manera que se cuenta el maximo de repeticiones de cualquier numero.

2.2. Metodologia

En la resolucion de este proyecto se utilizo el lenguaje de programacion Python 3, lenguaje de programacion que la gran mayoria de personas creen como el más simple de todos, realmente es un lenguaje bastante eficiente y simple de aprender, es uno de los lenguajes de programacion mas utilizados de la actualidad, por su eficiencia y facilidad de aprender. La historia de Python empieza en 1991 con su creador Guido Van Rossum lanzando la versión 0.9.0, que despues en el año 1994 se actualizaria a la versión 1.0 y despues a la versión 2.0 en el año 2000, para asi terminar en la version 3.12.4 que es la mas actual. La versión que se utiliza en el proyecto es la de Python 3.

2.3. Descripción de la solución

La solucion constaba de varias funciones, que se explican junto con el codigo, el codigo es el siguiente:

```
1  t = int(input())
2  while t:
3      t -= 1
4      c, r = input().split()
5      c = int(c)
6      r = int(r)
7      #Contador de digitos
8      List = [0] * 10
9      def digitos(numero):
10         digits = [int(d) for d in str(numero)]
11         return digits
12
13     def maximod(d):
14         cuenta = [0] * 10
15         for i in d:
16             cuenta[i] += 1
17         maxi = max(cuenta)
18         return maxi
19
20     #Funciones de "buscar a y b"
21     def encontrarayb(c,r):
22       i = 1000
23       while i < 100000:
24             a = i
25             b = a * c
26             m = a + (100000 * b)
27             dm = [x for x in digitos(m)]
28             if b >= 100000:
29                 break
30             if (maximod(dm) <= r) and (b >= 10000):
31                 print(str(str(b) + str("/") + str(a) + str(" = ") +
                       str(c)))
32             i += 1
33
34
35
36     encontrarayb(c,r)
```

De inicio se tiene la entrada de T, C y R, creando un while para que hayan T cantidad de C y R, despues de esto se empieza con el contador de digitos o la resolucion al problema de R, la primera funcion trabaja de manera que obtiene un numero cualquiera y lo interpreta como string, para despues sacar sus partes enteras(en este caso uno a uno sus digitos puesto que al interpretar un numero como string el codigo puede obtener cada numero de un digito por separado como entero) y insertarlas en una lista recien creada. La siguiente función, maximod, se basa en obtener una lista y contar cuantas veces se repiten los mismos numeros para crear otra llena de ceros y poner la repeticion de los numeros en esta, es decir, si hay una lista [1,2,3,1,5], terminaria creando otra lista que va cada ves que este numero se repite, en este caso quedaría [2,1,1,1], despues esta misma función va a retornar el numero más grande de esta lista(lo

que significa que esta retornando el maximo de repeticiones de digitos que hay en el numero). Para encontrarayb, que es la funcion en la que se encuentra la fraccion que pide el proyecto, se crea una variable i que empieza en 1000 y termina antes de 10000, se utiliza un ciclo iterativo para pobar con todos los numeros que van del 1000 al 100000 y encontrar a y b, como ya se sabe, por regla de divisibilidad, b/a = c es lo mismo que b = a * c, entonces se crean las variables a y b, y se crea una variable extra, m, la variable m se crea para poder utilizar la funcion que separa el numero en digitos, puesto que esta funcion solo obtiene un numero, entonces se utiliza la formula m = a + (100000 * b) para conservar todos los digitos, esto se puede probar con lo siguiente: a = 12345 y b = 67890, entonces a + (100000 * b) = 6789012345; se conservan todos los digitos, como dicho anteriormente, asi la funcion de separacion de digitos hace su trabajo en ambos numeros, despues, se especifica que si b supera los cinco digitos, el ciclo va a parar. Por ultimo, solo se revisa si se cumple la condicion de R y si los dos son de 5 digitos

2.4. Resolucion de pruebas

Las pruebas aplicadas fueron 10 y los tiempor mas importantes fueron estos: Menor Duración: real - 0m0.356s user - 0m0.341s sys - 0m0.046s Mayor Duración: real - 0m5.709s user - 0m4.400s sys - 0m1.147s Todas las 10 pruebas que fueron aplicadas fueron exitosas y estass son las duraciones mas importantes, la menor y la mayor, la mayor coresponde al caso de prueba 0 y la menor duracion corresponde al caso de prueba 1, del caso de prueba 2 al caaso de prueba 9 duran

2.5. Conclusiones

Los resultados fueron totalmente funcionales y nunca dieron ningun tipo de error y la devolucion que da, consiste en una serie de strings que se verian de la manera b/a = c, de esta manera se devuelven todos los resultados posibles en los que todas las se cumplan. Los resultados se dan por cada c y r que hay de ingreso y devuelven todas las posibles para cada c y cada r.

2.6. Aprendizajes

Al realizar este proyecto se aprendio sobre la utilizacion de listas y iteración, además como los comandos max, break y demás se utilizan.

Referencias

- [1] Lisa Tagliaferri. Cómo usar las instrucciones break, continue y pass cuando se trabaja con bucles en Python 3. 2021. - [2] Apuntes y notas de las clases de Introduccion a la programación y Taller de programación(IC1803) del TEC. 2024