



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Segundo Semestre del 2014

IIC1103 Introducción a la Programación Laboratorio 3

Objetivo General

A través del siguiente laboratorio tendrás que poner en práctica los conceptos vistos hasta ahora en clases: algoritmos, variables, flujo de control. Para ello deberás programar algoritmos en Python que den solución a los problemas planteados en la sección Enunciado de este documento.

Introducción

Las estructuras de flujo de control *if*, *elif*, *else* permiten que los programas no sean una secuencia lineal de instrucciones, sino que puedan ejecutar distintas instrucciones según los valores que existan en las variables en el momento de ejecución del programa; mientras que la estructura de control *while* permite repetir secuencias de instrucciones.

Enunciado

1. Raduz Setum, de la Universidad de Harvard está sumamente enfermo. Ha estado sufriendo de tuberculosis por un largo tiempo. Ahora, su pulmón izquierdo está irreparablemente dañado. No hay medicina que funciona en su cuerpo para aliviar el dolor, y es difícil para su familia mantener los gastos. Es así como nosotros debemos ayudarlo financieramente,
En este problema, debes construir un software que calcule donaciones. Inicialmente, el total de dinero es 0 y cada turno, habrá disponibles dos tipos de operaciones.

1. “donar” indica que el usuario ingresará un monto a donar.
 K ($100 \leq K \leq 10^5$), es un número entero que representa el monto que donará el usuario. Por lo tanto tendrás que agregar K a la cuenta.
2. “reportar”, reporta todo el dinero que hay en la cuenta hasta este momento.

Debes tener en consideración que en este problema hay un número variable de líneas de entradas y salidas, puedes ir escribiendo la salida a medida que te lo solicite el problema. El programa termina cuando el usuario, en lugar de ingresar “donar” o “reportar”, ingresa 0 (cero).

	Entrada	Salida
Diálogo 1	donar 1000	1000
	reportar donar 500	
	reportar 0	1500

2. A los niños se les enseña a sumar números de varios dígitos sumando las unidades, decenas, centenas, etc. de cada número. Para cada una de las sumas, si el resultado es mayor o igual a 10 entonces ocurre una operación de acarreo. El siguiente cuadro muestra que $756+329$ tiene dos operaciones de acarreo:

$$\begin{array}{r}
 \mathbf{1} \quad \mathbf{1} \quad \leftarrow \text{acarreo} \\
 7 \ 5 \ 6 \leftarrow \text{primer sumando} \\
 3 \ 2 \ 9 \leftarrow \text{segundo sumando} \\
 \hline
 1 \ 0 \ 8 \ 5 \leftarrow \text{suma}
 \end{array}$$

Muchos encuentran que esta operación es un desafío significativo, por lo que se te pide escribir un programa que cuente el número de operaciones de acarreo en una suma de dos números para que los educadores puedan evaluar la dificultad de la suma. Escribe un programa en Python que pida dos números enteros mayores que 0 y diga el número de operaciones de acarreo necesarias para hacer la suma.

	Entrada	Salida
Diálogo 1	759 329	2
Diálogo 2	1001 19	1

3. El sistema de numeración más conocido es el decimal (base 10) y cuenta con 10 dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Pero también existen otros sistemas numéricos, por ejemplo, el binario (base 2) que sólo se compone de 0 y 1, o el sistema numérico en base 3 que cuenta sólo con 0, 1 y 2 como dígitos (ver Tabla 1).

Base	Dígitos válidos
2	0,1

3	0,1,2
4	0,1,2,3
5	0,1,2,3,4
6	0,1,2,3,4,5
7	0,1,2,3,4,5,6
8	0,1,2,3,4,5,6,7
9	0,1,2,3,4,5,6,7,8
10	0,1,2,3,4,5,6,7,8,9

Tabla 1: Sistemas Numéricos

Ahora bien, es posible convertir un número de base n ($2 \leq n \leq 9$ para simplificar) a base decimal utilizando la formula del *Teorema Fundamental de la Numeración* (TFN) que establece que en cualquier sistema de numeración posicional todos los números pueden expresarse mediante la siguiente suma de productos:

$$N_b = \sum_{i=0}^{p-1} a_i * b^i$$

Que es equivalente a:

$$N_b = a_{p-1} * b^{p-1} + a_{p-2} * b^{p-2} + \dots + a_1 * b^1 + a_0 * b^0$$

Donde b es la base del número N , a_i son los elementos pertenecientes al sistema numérico de N (dígitos), y p es el número de dígitos de N .

Por ejemplo, convertir el número 452_7 (7 es la base) a base decimal (base 10) sería:

$$452_7 = 4 * 7^2 + 5 * 7^1 + 2 * 7^0 = 4 * 49 + 5 * 7 + 2 = 233_{10}$$

Crea un programa que permita al usuario ingresar 2 números: n y m , donde n es la base y m es un número en esa base (ambos positivos y enteros). Si los datos de entrada no están correctos (es decir, m no está en base n), el programa debe retornar -1. Si los datos son correctos, el programa debe retornar la conversión del número ingresado a base 10.

	Entrada	Salida
Diálogo 1	2 1201110	-1
Diálogo 2	7 452	233
Diálogo 3	10 134	134

4. Problema de Cajero

Suponga que un cajero quiere minimizar el numero de billetes y monedas que da

de vuelto cuando le hacen un pago. El cajero tiene que manejar 12 variables en la resolución de este problema que son la cantidad de billetes de disponibles en caja de \$20.000, \$10.000, \$5.000, \$2.000 y \$1.000, la cantidad de monedas de \$500, \$100, \$50 \$10 y \$1. A estas variables se suman las variables del monto a pagar y el monto entregado sobre el que debe dar vuelto.

Ejemplo: Si le pagan \$2.500 con \$10.000 el debería dar de vuelto 1 billete de \$5.000, 1 de \$2.000 y una moneda de \$500, en el entendido que tiene de estos billetes y monedas. Si la cantidad de billetes de \$5.000 es cero, pero tiene suficientes billetes de \$2.000 y \$1.000 puede dar 3 billetes de \$2.000, 1 de \$1.000 y la moneda de \$500 (si es que la tiene, si no debe dar monedas de \$100 o \$50 o de \$10).

Para no estar pensando el cajero le dice a Usted que haga un programa que realice esta operación de recibiendo como entrada estas variables.

	Entrada	Salida
Diálogo 1	2 1 2 2 3 2 2 3 4 1000 2500 10000	1 5000 1 2000 1 500
Diálogo 2	2 0 0 10 3 0 100 3 4 1000	3 2000 1 1000 5 100

[illegible]