Tarea 1

Conversión numérica, cálculos en complemento a 2
y representación en punto flotante

Reglas generales

La tarea consiste en la realización de un programa en `Python` que convierta números entre bases numéricas, desarrolle sumas y restas usando complemento a 2, y permita obtener la representación de un número decimal como un número flotante de precisión simple. Se recomienda la utilización de `Python 3.7` o superior, ya que los estándares de esas versiones se utilizarán para la revisión de las tareas.

Enunciado

Conversión entre bases numéricas

Debe convertir un número de entrada con su base especificada a otra base distinta, imprimiendo el resultado a través de la consola (stdout). Tanto la base de entrada como la base de salida estarán entre 1 y 64, incluyendo los límites. La base 1 (unario) utiliza el número '`1`' repetido como símbolo, mientras que todas las bases siguientes (desde binario hasta tetrasexagesimal[1], 2 a 64) usan una partición de izquierda a derecha de los siguientes símbolos

`0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ+?`
(Los dígitos del 0 al 9, luego todas las letras minúsculas, luego todas las letras mayúsculas, y los símbolos '+' y '?')

Por ejemplo, la base 17 (heptadecimal) usaría los símbolos '`0123456789abcdefg`', mientras que la base 32 (duotrigesimal) usaría los símbolos '`0123456789abcdefghijklmnopqrstuv`'.

Suma y resta usando complemento a 2

Debe realizar la suma y resta de dos números binarios de 8 bits utilizando complemento a 2. La salida se debe hacer a través de la consola (stdout) y debe mostrar dos líneas con la suma de ambos números seguido del resultado, y lo mismo para la resta. Recordar que, $A - B = A + (-B)$, y que el complemento a 2 permite obtener un número con su signo opuesto en binario. Por otra parte, si llegase a ocurrir un *overflow*[2], la salida de la operación correspondiente debe ser '`OVERFLOW`'.

Si tenemos los números `00011011` (27) y `00000101` (5), su suma y resta se representan de la siguiente forma, teniendo en cuenta que -5 se representa como `11111011` según el estándar de complemento a 2

Suma	$27 + 5 = 32$	`00011011 + 00000101 = 00100000`
Resta	$27 + (-5) = 22$	`00011011 + 11111011 = 00010110`

Representación en punto flotante de precisión simple

Debe calcular la representación en punto flotante de precisión simple (IEEE754) de un número decimal. El número decimal puede ser positivo, negativo, o causar un overflow en cualquiera de las dos direcciones. Además debe poder recibir un valor no numérico que debe tratar como `NaN` (Not a Number). Haga uso de los códigos especiales establecidos por el estándar IEEE754 para representar estos casos excepcionales.

Exponente	Mantisa	Caso Excepcional
255	0	Infinito (overflow en cualquier dirección)
255	Distinto de 0	NaN (Not a Number)

[1] http://www.numberbases.com/terms/basename.html
[2] En este caso, un overflow ocurriría cuando el resultado de la operación no pueda ser representado en 8 bits.

Joaquín Montes joaquin.montes@sansano.usm.cl
Sebastián Alvarado sebastian.alvaradoa@sansano.usm.cl

Formato de entrada y salida

La **entrada** será por consola (stdin), y siempre será un número entero `M` seguido de una de tres líneas posibles según corresponda. Debe continuar recibiendo una entrada hasta que se lea un 0 como se indica más abajo.

- Si el primer número `M` es 1, entonces le seguirá una línea de la forma '`N B T`', donde `N` es un número entero, `B` la base en que se encuentra dicho número, y `T` la base a la que se debe convertir.
- Si el primer número `M` es 2, entonces le seguirá una línea de la forma '`X Y`', donde ambos números `X` e `Y` son números binarios de 8 bits cada uno, incluyendo ceros adelante para completar los 8 bits en caso de ser necesario.
- Si el primer número `M` es 3, entonces le seguirá una línea de la forma '`D`', donde `D` puede ser un número decimal o cualquier otro string.
- Si el primer número `M` es 0, entonces debe finalizar la ejecución del programa.

La **salida** será por consola (stdout) a través de un mensaje impreso con uno de dos formatos según el valor del número `M`.

- Si el número `M` es 1, entonces la salida será una sola línea con el formato '`Base T: C`', donde `T` es la base a la que se convirtió, y `C` el número convertido.
- Si el número `M` es 2, entonces la salida serán dos líneas con el formato '`Suma: P + Q = S`' para la primera, y '`Resta: P + (-Q) = S`' para la segunda. En ambos casos `P` corresponde al primer número de la entrada, `Q` al segundo, y `S` al resultado de la operación. En el caso de la resta, `-Q` será el valor del segundo número de la entrada convertido usando complemento a 2, ya que $A - B = A + (-B)$.
  - En caso de que exista un overflow en alguna de las dos operaciones, la salida debe ser '`Suma: P + Q = OVERFLOW`' o '`Resta: P + (-Q) = OVERFLOW`' según corresponda.
- Si el número `M` es 3, entonces la salida será una línea con un número binario en punto flotante de precisión simple (IEEE754). Debe tratar los casos excepcionales según la tabla de códigos especiales del estándar.

Joaquín Montes joaquin.montes@sansano.usm.cl
Sebastián Alvarado sebastian.alvaradoa@sansano.usm.cl

Datos de ejemplo

Entrada	Salida
1 f0 16 2	Base 2: 11110000
1 23 4 18	Base 18: 23
1 Dx? 64 60	Base 60: IWD
2 00011011 00000101	Suma: 00011011 + 00000101 = 00100000 Resta: 00011011 + 11111011 = 00010110
2 00110011 11001100	Suma: 00110011 + 11001100 = 11111111 Resta: 00110011 + 00110100 = 01100111
2 10101010 11110000	Suma: 10101010 + 11110000 = 10011010 Resta: 10101010 + 00010000 = 10111010
2 01011101 00110111	Suma: 01011101 + 00110111 = OVERFLOW Resta: 01011101 + 11001001 = 00100110
3 7	01000000111000000000000000000000
3 -118.625	11000010111011010100000000000000
3 test	01111111111111111111111111111111
3 400000...000 (38 ceros)[3]	01111111100000000000000000000000
0	

[3] En notación científica, $4 \cdot 10^{38}$, lo que produce un overflow positivo

Consideraciones

- **La fecha de entrega para la tarea es el viernes 8 de octubre a las 23:55 hrs.**
- **Se descontarán 25 puntos de la nota máxima por cada día o fracción de atraso en la entrega, hasta un máximo de 2 días. Cualquier atraso por sobre esto se evaluará con nota 0.**
- La tarea debe realizarse individualmente. Ante cualquier sospecha de copia o trabajo colaborativo se informará a las autoridades correspondientes y se evaluará con nota 0.
- La tarea debe realizarse usando el lenguaje `Python`, de preferencia con la versión 3.7 o superior.
- La tarea se debe entregar via Aula en un solo archivo comprimido en formato `.zip` de nombre `T1_APELLIDO.zip` que incluya los siguientes archivos:
  - Un solo archivo `README.txt` con el nombre y ROL USM del estudiante, además de cualquier aclaración que sea necesaria.
  - Un solo archivo `.py` que contenga todo el código en Python para la ejecución de la tarea.
  - Un solo archivo `.pdf` con el informe completo del desarrollo de la tarea. Se recomienda utilizar LaTeX (en Overleaf por ejemplo) u otra variante de TeX para redactar la tarea.
- El informe debe contener las siguientes secciones, cada una ordenada y con toda la información necesaria:
  - Portada, incluyendo el nombre y ROL USM del estudiante, además de un título descriptivo.
  - Resumen, donde describa brevemente el desarrollo y resultados de la tarea.
  - Introducción, dejando claro el objetivo de la tarea y cualquier algoritmo que utilice.
  - Desarrollo, explicando detalladamente la resolución de la tarea.
  - Resultados, con todos los valores que haya obtenido durante el desarrollo de la tarea. Incluya extractos de cualquier prueba que haga con su tarea.
  - Análisis, donde discuta los resultados de la sección anterior y cualquier complicación con la que se haya encontrado.
  - Conclusión, comentando el nivel de finalización de la tarea.
- La sección de código de la tarea pondera por 60 % de la nota, mientras que el informe pondera por 40 %. En caso de no entregarse una de las dos partes, se evaluará la tarea completa con nota 0.
- Para que el informe se considere válido (o entregado), al menos 3 partes de este deben ser desarrolladas correctamente.
- Todas las preguntas respecto a la tarea deben hacerse a través del foro de consultas en Aula. **No se responderán dudas durante las 48 horas previas a la entrega**.

Joaquín Montes joaquin.montes@sansano.usm.cl
Sebastián Alvarado sebastian.alvaradoa@sansano.usm.cl