

INF245: Arquitectura y Organización de Computadores

Tarea 1

Profesor Mauricio Solar
Ayudantes Joaquín Montes
 Sebastián Alvarado
 Felipe González
 Javier Rojas

1. Reglas Generales

Para esta tarea deberá utilizar el lenguaje de programación Python para escribir un programa capaz de completar los objetivos mencionados en la sección de enunciado. Se recomienda usar Python 3.7.

Además deberá confeccionar un informe detallando el desarrollo de su tarea. El informe debe entregarse en formato PDF y contener las secciones Portada, Resumen, Introducción, Desarrollo, Resultados, Análisis, y Conclusión. Se recomienda utilizar \LaTeX para construir el informe, márgenes de 25 milímetros, y mantener el orden haciendo uso de secciones y subsecciones.

2. Enunciado

La empresa *OcularTM* lo contrató para desarrollar el futuro de la entretenición digital: *Abyss*, una consola de videojuegos de inmersión completa, que captura todos los sentidos del jugador y le permite disfrutar de un mundo digital ultra realista.

Su primera tarea consiste en crear un programa que traduzca los distintos tipos de código que utilizará la consola, desde código en unario (base 1), hasta código en base 64. Se le ha otorgado una estación de trabajo con software de vanguardia, incluyendo el nuevo lenguaje de programación que revolucionó al mundo, Python 3. **Utilice los algoritmos que conoce para convertir números entre dos bases cualesquiera.**

OcularTM se preocupa mucho de la seguridad de sus sistemas, por lo que además le solicite que implemente los códigos BCD, Gray, Exceso de 3, Johnson, Paridad, PentaBit y Hamming; todos estos tomando números en binario e interpretándolos a su manera. Las bases numéricas entre 1 y 64 se expresan con su valor, mientras que los códigos siguen la siguiente tabla:

Código	Expresión
BCD	bcd
Gray	gry
Exceso de 3	ed3
Johnson	jsn
Paridad	par
PentaBit	pbt
Hamming	ham

3. Entrada y salida de datos

- La entrada de datos se hará a través de la entrada estándar (`stdin`).
- La salida de datos se hará a través de la salida estándar (`stdout`).
- Cada línea de la entrada tendrá 3 valores `n b t`, donde
 - $1 \leq n \leq 1000$ (1000 en base 10)
 - $1 \leq b \leq 64$ o uno de los siguientes códigos: `bcd`, `gry`, `ed3`, `jsn`, `par`, `pbt`, `ham`
 - $1 \leq t \leq 64$ o uno de los siguientes códigos: `bcd`, `gry`, `ed3`, `jsn`, `par`, `pbt`, `ham`
 - `n` es el número a convertir, `b` la base en que se encuentra `n`, y `t` la base a la que convertir.
 - El programa debe seguir leyendo líneas hasta que se lea un único carácter - (un guión)
 - Si `b` es un código, entonces se asume que la base de `n` siempre es 2.
- Cada línea de la salida debe seguir uno de los siguientes formatos
 - Base `t`: `x` si `t` es una base entre 1 y 64
 - Código `t`: `x` si `t` es un código (debe usar el nombre completo del código según la tabla)
 - Entrada inválida si `n` no es válido en la base `b`, o no se cumple alguno de los límites expresados anteriormente.
- La base 1 (unario) usa solo el símbolo 1.
- Las bases entre 2 y 64 usan una partición desde la izquierda de los siguientes símbolos
`0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ+?`
(dígitos del 0 al 9, letras minúsculas, letras mayúsculas, el símbolo +, y el símbolo ?)
- Todos los códigos entregan su resultado en binario.
- Si `t` es `bcd`, `ed3`, entonces debe transformar el número a decimal para luego obtener su codificación.
- Si `t` es `gry`, `jsn`, `par`, `pbt`, `ham`, entonces el valor a convertir siempre debe ser binario.
- Si `t` es `gry`, debe entregar la posición o índice del número en la tabla de valores de Gray.
- Si `t` es `jsn`, la longitud del código debe ser la menor potencia de 2 suficiente.
- Si `t` es `par`, el resultado debe ser 0 si se detecta un error, y 1 en caso contrario. Se asume que el número es correcto si la cantidad de 1s es par.
- Si `t` es `pbt`, el resultado debe ser 0 si se detecta un error, y 1 en caso contrario. Se asume que el número es correcto si la longitud del número es múltiplo de 5.
- Si `t` es `ham`, el resultado debe ser el valor corregido. Si no es posible corregir el error, la salida debe ser 0.

4. Datos de ejemplo

A continuación se presenta una serie de entradas y sus respectivas salidas tras ser procesadas por el programa.

stdin	stdout
1010 2 10	Base 10: 10
1337 8 2	Base 2: 1011011111
221 3 bcd	Código BCD: 100101
10001 bcd 10	Base 10: 11
11 gry 2	Base 2: 11

Explicación de los ejemplos

- 1010 2 10: Convertir 1010 de binario a decimal. Usando el algoritmo estándar trivialmente obtenemos 10_{10} .
- 1337 8 2: Convertir 1337 de octal a binario. La conversión es trivial, obteniendo 1011011111_2 . Notar que 1337_8 es menor a 1000_{10} , pues el límite en octal es 1750_8 .
- 221 3 bcd: Convertir 221 de ternario a código BCD. Como BCD necesita un número decimal, primero lo convertimos obteniendo el valor 25_{10} . Luego interpretamos este decimal con el código BCD, resultando en 100101_{bcd} .
- 10001 bcd 10: Convertir 10001 de código BCD a decimal. Tomamos el número 10001_{bcd} y lo convertimos a base 10, obteniendo el valor 11_{10} .
- 11 gry 2: Convertir 11 de código Gray a binario. Comprobamos que 11_2 es un índice válido. Luego convertimos el índice a la base de salida, obteniendo el mismo número: 11_2 .

5. Consideraciones

- La tarea debe realizarse individualmente. Ante cualquier sospecha de copia o trabajo colaborativo se informará a las autoridades correspondientes.
- La tarea debe realizarse utilizando un editor de texto plano o código cualquiera. El código debe estar escrito usando el lenguaje de programación Python 3. Se recomienda utilizar Python 3.7 o superior.
- La entrega se realizará a través de Aula en un solo archivo en formato `.zip` de nombre `T1_APELLIDO_ROL.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` con el código fuente 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 alumno, 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 fórmula 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 programa.
 - 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. Además explique generalmente la función de las distintas bases numéricas y códigos de corrección estudiados en clases.
- **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.**
- **La fecha límite de entrega de la tarea es el domingo 9 de mayo de 2021, a las 23:59 horas.**
- **Solo se recibirán tareas con un retraso máximo de 72 horas desde la fecha límite de entrega, las que podrán optar a una nota máxima de 75. Cualquier retraso por sobre las 72 horas mencionadas será evaluado con nota 0.**