

Programación científica IBIO2240

Laboratorio 2

Ejercicio 1: Número entero positivo a binario (unsigned int o entero sin-signo) (1.0)

Escribir un programa en Python (.py) donde el usuario ingrese un número entero positivo (sin signo, se incluye el cero) y éste sea convertido a binario. Si el número no es entero o positivo, se debe mostrar un mensaje de error.

Importante: No se puede usar la función *reverse*.

Ejercicio 2: Representación binaria en complemento a 2 de números enteros (1.0)

Escribir un programa en Python (.py) donde el usuario ingrese un número entero positivo o negativo, el número de bits, y éste sea convertido en su correspondiente representación binaria de complemento a dos. Si el número no es entero, o si el número de bits no es suficiente para representarlo, el programa debe mostrar un mensaje de error.

Importante: No se puede usar la función *reverse*.

Ejercicio 3: Representación binaria en offset binario de números enteros (1.0)

Escribir un programa en Python (.py) donde el usuario ingrese un número entero positivo o negativo, el número de bits, y éste sea convertido en su correspondiente representación binaria en offset binario. Si el número no es entero, o si el número de bits no es

suficiente para representarlo, el programa debe mostrar un mensaje de error.

Importante: No se puede usar la función *reverse*.

Ejercicio 4: Representación binaria de punto flotante 32-bits IEEE754 (1.0)

Escribir un programa en Python (.py) donde el usuario ingrese un número real y éste sea convertido en su correspondiente representación binaria de punto flotante de 32-bits.

Importante: No se puede usar la función *reverse*.

Ejercicio 5: Representación binaria de punto flotante 64-bits IEEE754 (1.0)

Escribir un programa en Python (.py) donde el usuario ingrese un número real y éste sea convertido en su correspondiente representación binaria de punto flotante de 64-bits.

Importante: No se puede usar la función *reverse*.