

# Tarea 1

Fecha de entrega: antes del lunes 22 de Agosto

**i)** Verificar que las siguientes conversiones de base son correctas

$$(17)_{10} = (10001)_2$$

$$(17.25)_{10} = (10001.01)_2$$

**ii)** Verificar que multiplicar por 2 un número en base 2 es correrlo a la izquierda y rellenar por ceros por la derecha. Análogamente ¿qué operación es dividir por 2?

**ii)** Hacer las siguientes conversiones en papel

$$(5637.265625) = (?)_2$$

$$(?)_{10} = (10001011101.000110101011)_2$$

$$(67345934076.33203125)_{10} = (?)_2$$

**iii)** Escribir los 3 números del punto anterior en formato normalizado en base 2 de la IEEE en float y en double.

**iv)** Hacer el cálculo paso a paso de la distancia absoluta entre 1 y su número anterior en representación maquina que se muestra al final de la lámina 5 de la clase01b.pdf.

v) Hacer una función en C/C++ que *redondea por truncamiento* a  $n$  dígitos de precisión el valor que se le manda por parámetro. El prototipo puede ser

```
double chopNDigits(double val,int nDigits){...}
```

De tal forma que si ejecuto:

```
double a = 3.141592; a = chopNDigits(a,3);  
ahora la variable a debe de tener el valor 3.14.
```

```
double a = 31415.92; a = chopNDigits(a,4);  
ahora la variable a debe de tener el valor 31410.
```

tip: algunos pasos para llevar a precisión a 5 dígitos el # 3.14159216 -> 0.314159216 -> 31415.0 -> 3.1415

vi) Del libro "Análisis Numérico" de Richard Burden y J. Douglas Faires, 7a edición, y usando la función del punto anterior reproducir el ejemplo 5 del capítulo 1, página 24, sobre el cálculo de las raíces de una ecuación cuadrática . Se anexa pdf de una versión digital.

vii) Buscar en internet el programa en C/C++ para calcular el epsilon machine, ejecutarlo para float y double y verificar que se cumplen las formulas directas para calcularlo vistas en clase.