Tarea 1

1 Representación de los numeros

Ejercicio 1.1 *Recordamos que un número hexadecimal (en base 16) se convierte fácilmente en binario : cada dígito de un numero hexadecimal corresponde a 4 dígitos binarios. Un octeto se representa por un numero de dos dígitos hexadecimales. Por ejemplo,*

$$A2_H = 1010\,0010_b = 162_{10}.$$

Hacer las operaciones siguientes sobre ***enteros con signos****, sobre un octeto (indicar cuando hay un problema, como overflow), indicando los valores en decimal*

- $28_H + C1_H$
- $A4_H + 43_H$
- $27_H + EA_H$
- $E4_H - 34_H$
- $91_H - 45_H$

Ejercicio 1.2 *Hacer un programa en que se crearán las representaciones en flotantes simples de los números siguientes,* ***partiendo de su representación binaria****. Por ejemplo, el código siguiente permite imprimir el "-0".*

```
unsigned int a = 0x80000000;
float p = *(float)&a;
printf("p=%f \n",p);
```

Hacer lo mismo para:

1. *un número denormalizado,*
2. *el positivo más pequeño que se pueda representar,*
3. *el positivo más grande que se pueda representar,*
4. *mas infinito,*
5. *menos infinito,*

6. *algún Not A Number (NaN).*

Ejercicio 1.3 *¿ Cuál es el double mas grande ? ¿Y el que está justo antes ? ¿Cuál es su diferencia ? Cuales son los números mas pequeños representables en maquina (normalizados y no-normalizados) ?*

Ejercicio 1.4 *Sea la suma*

$$S_n = \sum_{i=1}^{n} \frac{1}{n}.$$

1. *¿Qué vale S_n, teóricamente?*
2. *Escribir dos funciones que calculen la suma arriba, la primera con flotantes, la segunda con doubles.*
3. *Calcular S_n para $n = 1, .., 10$. En casa caso, verificar que sí o no el resultado es igual al valor teórico, y, si no, imprimir la diferencia con el valor teórico.*
4. *Explicar los resultados obtenidos.*