

A = 11111(10) -> configuración en base 10 de un número almacenado en formato binario de punto fijo con signo de 2 bytes
B = 250415(8) -> configuración en base 8 de otro número almacenado en formato empaquetado de 3 bytes

a) Indicar cuales son los números almacenados en base 10.

A = 11111 (10)

1) Pasar de (10) -> (2) Divisiones sucesivas

Mejorado 1bis) pasar el nro (10) a (16)

11111 (10) = 2B67 (16)

2 bis) pasar de (16) a (2) usando la regla de potencia $2^4 = 16$

2B67 (16) = 0010 1011 0110 0111 (2)

2) vemos el formato cuantos bytes tiene y si lo cumple -> 0010 1011 0110 0111 (2)
como el 1er bit es 0 implica que positivo -> NO SE COMPLEMENTA

3) pasarlo a base 10 -> por lo tanto nos queda 11111 (10)

(2) -> (10) -> multiplicaciones sucesivas $1 \cdot 2^{13} + 1 \cdot 2^{11} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$

(2) -> (16) -> usando potencia

2B67

y luego a base (10) -> $2 \cdot 16^3 + 11 \cdot 16^2 + 6 \cdot 16^1 + 7 \cdot 16^0$

```
0010 1011 0110 0111 -> tiene 16 bits
00 | 0010 1011 0110 0111
1 | 0010 1011 0110 0111
```

que pasaria si cuando paso a base 2 me queda asi --> 0010 1011 0110 0111
1000 1010 -> es un nro en BPF c/2 de 8 bits -> lo quiero llevar a 16 bits - con que completo?

B = 250415(8) -> configuración en base 8 de otro número almacenado en formato empaquetado de 3 bytes
empaquetado se escribe en (16)

1) pasarlo a base (2) -> 00010101000100001101 (2)

2) pasarlo a base (16) -> 01510D (16)

3) obtenemos el nro -> - 1510 porque BD es negativo!!

b) Realizar la división entera de su número de padrón por 5 y a eso lo llamamos P. Restar el valor almacenado en A con P es decir hacer A - P en formato BPF c/signo de 16 bits indicando la validez del resultado. Ej padrón 100123 => P es 20024

A - P -> A + Comp (P)

1) hallar A y P

A ?? -> 11111 (10)

P ?? -> 106957/ 5 = 21391 es mi nro P (10)

2) pasar A y P a BPF c/s 16 bits

A = 0010 1011 0110 0111 (2)

P -> pasamos a (16) y luego (2) -> 0101 0011 1000 1111 (2)

A - P -> A + Comp (P)

P -> 0101 0011 1000 1111 (2) como hallo Comp(P) -> Not (P) + 1
1010 1100 0111 0000

```
+
1010 1100 0111 0000
+
1
-----
```

1010 1100 0111 0001 -> Comp (P)

A = 0010 1011 0110 0111 (2)

```
+
Comp (P) = 1010 1100 0111 0001 (2)
-----
```

1101 0111 1101 1000 (2) -> es valido por que? porque los 2 ultimos bits de acarreo son iguales a 0

c) Tomar el valor del resultado del punto anterior si es válido o el valor de B si no lo fuera y almacenarlo en formato BPFlotante IEEE 754 de precisión simple, dando su configuración octal.

1101 0111 1101 1000 (2) es valido

Almacenar en BPFlotante IEEE 754 PS -> 32 bits

1) Obtener el nro

1101 0111 1101 1000 (2) -> aplicamos Not + 1

- 0010 1000 0010 1000 -> esto es Not +1

2) flotante -> S M x B Exponente

Mantisa -> normalizada => tomo el nro y lo normalizo -1, 0100000101000 x 2¹³

3) almacenamos en BPFlotante

S = 1

EExc ->

EExc = Exp + 127

= 13 + 127

= 140 (10) -> 1000 1100 (2)

EExc = 1000 1100

M = 0100000101000 0000000000 -> agrego los 0 hasta completar los 23 digitos de la mantisa

1 1000 1100 0100000101000 0000000000 (2) -> el nro almacenado en BPFlotante IEEE 754 PS

Pasar de (2) a (8) -> agrupamos de 3

Ejercicio Parcial 1er Cuat 2021

Dado A que representa la configuración en base 8 de un número almacenado en formato BPF c/signo de 16 bits y B que representa la configuración en base 10 de otro número almacenado en formato empaquetado de 3 bytes se pide:

A = 177546(8)

$$B = 78987(10)$$

1. Indicar cuáles son los números almacenados en base 10.
2. Tomar el valor P como los últimos 4 dígitos del número de padrón (ej.: Padrón 9876510 => P = 6510) y restarle al valor del A obtenido en el punto anterior en formato BPF c/signo de 16 bits (A – P) indicando la validez del resultado. Nota: Escriba CLARAMENTE su número de Padrón.
3. Tomar el valor resultado del punto anterior y almacenarlo en formato BPFlotante IEEE 754 de precisión simple expresando su configuración octal.