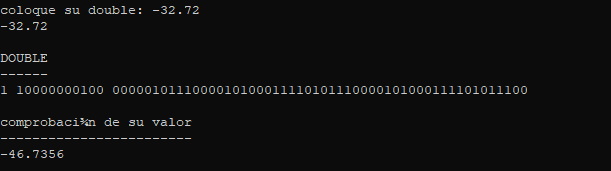
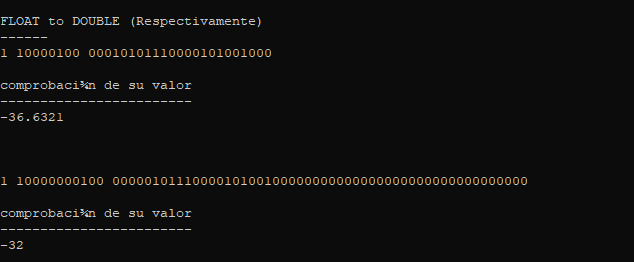


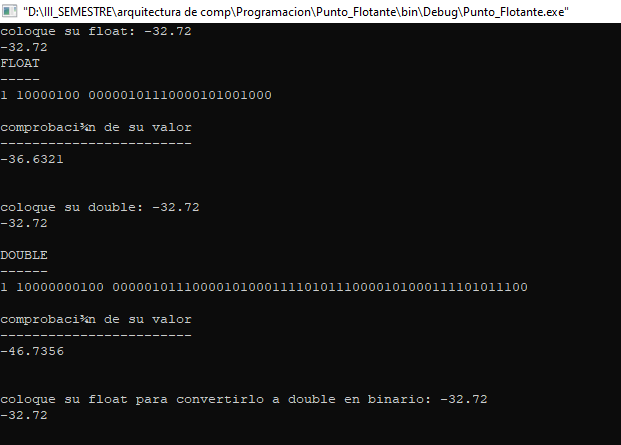
En la conversión del Float, se ve que al momento de comprobar el valor que se ha generado, ha “disminuido”, esto pasa porque en el algoritmo conversión a punto flotante, solo existe una restricción para que el significando deje de llenarse y es que todos lo 23 bits ya hayan sido tocados por nuestra función para llenarse. Mi solución a esto es que también se ponga otra restricción que evalúe en todo momento el valor que genera el Significando junto al Exponente y al signo y que la restricción sea menor al valor original.



Lo mismo de arriba.



Aquí no pude lograr que el valor de float al ser convertido en double afecte al bit, ya que el double.



Pero al ver el ejemplo anterior, como tienen el mismo valor (-32.72) hay algunas diferencias:

-Exponentes:

N=valor del exponente normalizado, n=exp.Biased (form. 32bits) = 8, m=exp.Biased (form 64bits).

La diferencia: N + pow (2, m-1) -1 – (N+ pow (2, n-1)-1)

Y para convertir el exponente de 8 bits a 11 bits se le debe sumar la diferencia

N+ pow (2, n-1)-1 + diferencia, ese será el nuevo valor del exponente biaset, luego lo convertiremos a bits y saldrá con más ceros por el medio.

-Significando:

El significando solo varía porque la parte decimal seguirá siendo multiplicada por dos y tomando de la multiplicación la parte entera, y seguirá tomando la nueva parte entera hasta que termine de llenarse el Significando.