7.a) Cual es la implementación correcta?
La implementación correcta es la 4. f4(x) = ((sin(x) + cos(x))/2)*(a+b).

b) En cada caso, en que se diferencia cada implementación de la formula correcta?
Lo que se quiere calcular más explicito en cada caso es:

f1(x) = (in(x) + ((cos(x)/2) * a) + b
f2(x) = sin(x) + ((cos(x)/2) * (a + b))
f3(x) = ((sin(x) + (cos(x)/2)) * a) + b

c) Que aprendió sobre la precedencia de los operadores en este ejemplo?
Que si no se esta seguro de que la operación que se esta realizando es la operación que se quiere, es mejor utilizar los paréntesis para dejar todo más claro.

11.para cuantas iteraciones se vuelve importante considerar la optimización mencionada?
Desde que se realicen más de una iteración, ya que estas evitando calcular el seno y el coseno una vez por iteración.

12. Cual código tarda mas?

El primer caso tarda: 0.000004
El segundo caso tarda: 0.000228

13.Puede notar alguna diferencia en el tiempo de ejecución del anterior programa si se corre en int, float, double y long double precisión?

El primer caso con int tarda: 0.000002
El segundo caso con int tarda: 0.000114

El primer caso con float tarda: 0.000001
El segundo caso con float xtarda: 0.000094

El primer caso con double tarda: 0.000001
El segundo caso con double tarda: 0.000097

El primer caso con long double tarda: 0.000002
El segundo caso con long double tarda: 0.000136

14.Cual ejecución demora mas? Por que?
El loop1 tardo: 0.050105
El loop2 tardo: 0.047089

El caso 1 dura más, ya que en el caso 2 es mas probable que x[i] permanezca en el caché mientras se usa, lo cual favorece la localidad temporal.

15. Cual método funciona mas rápido? Cambian los resultados con cada implementación?
Los tres métodos entregan una misma duración en promedio y el mismo resultado.

16.Hasta donde se observa ganancia en el tiempo de ejecución?
Para este caso (suma de los primeros n-1 números) se observa ganancia de tiempo hasta que se fracciona la suma en 7 partes, ya que cuando se fracciona en 10 partes el tiempo de computo vuelve a subir.