PROCEDURES AS ARGUMENTS

Here is the generalized procedure for computing sums of terms:

\[
\text{(define (sum term a next b)}
\text{  (if (> a b)}
\text{    0}
\text{    (+ (term a)}
\text{       (sum term (next a) next b))))}
\]

**Question 1.** What sort of thing is each of the following parameters to the `sum` procedure? Write the name of the predicate which would answer `#t` for each parameter.

Possible predicates are: `number?` `symbol?` `pair?` `boolean?` `procedure?`

Parameter | Predicate that would report `#t`
---|---
term |
a |
next |
b |

**Question 2.** Using `sum`, write a procedure that will calculate the sum of every other integers over a range. Name this procedure `sum-alt-ints`, and then invoke it to calculate the sum 3 + 5 + 7 + … + 99.
Question 3. Using `sum`, write a procedure that will calculate the sum of the sequence \( \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \) etc. Name this procedure `approaching-one`, and then invoke it to calculate the sum through (and including) \( \frac{1}{512} \).

Question 4. Using `sum`, write a procedure that will calculate the sum of the sequence \( 1^2 – 2^2 + 3^2 – 4^2 \ldots \) etc. Name this procedure `diff-alt-squares`, and then invoke it to calculate the series from 1 to 100.