Problem 1. Find the number of times (runtime complexity) function g is called in the pseudo-code shown below:

```
function g(n)
    if n = 1 then return
    g(n-1)
    g(n-1)
end function
```

Your answer should be in terms of the input n in the most simplified form. Show your work. In addition, write the asymptotic runtime of the algorithm.

Problem 2. Suppose you're given a positive integer $x > 0$ and two sorted arrays A and B, each contains n positive integers. Write an efficient pseudo-code to find two numbers $A[i]$ and $B[j]$ such that $A[i] + B[j] = x$. You should output "yes" if there exist indices $1 \leq i, j \leq n$ for which $A[i] + B[j] = x$ and "No" otherwise. Find the exact worst-case number of comparisons. Show your work.

Problem 3. Selection sort finds the largest element and puts it at the end of the array. Consider a version of Selection sort that finds the two largest elements and puts both of them at the end of the array (in order). Write the pseudo-code for this version of selection sort. Make sure that it works when the size of the array is odd.

Problem 4. Both selection and insertion sort are quadratic runtime algorithms in the worst case. List an advantage of insertion sort over selection sort and one advantage of selection sort over insertion sort? Be specific.