1. Write a program by using an user defined function for computing √n for any positive integer n. Besides assignment and comparison, your algorithm may only use the four basic arithmetical operations.

2. Design a data structure to maintain a set S of n distinct integers that supports the following two operations:

a) INSERT(x, S): insert integer x into S

b) REMOVE(S): remove the smallest n/2 integers from S.

c) Write a program by using UDFs that give the worse-case time complexity of the two operations INSERT(x, S) in O(1) time and REMOVE-BOTTOM-HALF(S) in O(n) time.

3. Let A be an array of n integers a0,a1,... ,an-1 (negative integers are allowed), denoted, by A[i... j], the sub-array ai, ai+1,... ,aj for i≤j. Also let Si-j denote the sum ai + ai+1 +· · · + aj. Write a program by using an udf that must run in O(n2) time to find out the maximum value of Si-j for all the pair i, j with 0 ≤ i ≤ j ≤ n-1. empty sub-array will have sum Si-j = 0. Indeed, if all the elements of A are negative, then one returns 0 as the maximum sub-array sum.