Burton Rosenberg
Final

There are three problems for a total of 99 points. Show all your work, partial credit will be awarded. When there is not enough room on the test page itself, write in the provided blue books and write and sign your name on each one. No notes, no collaboration.

Name: _________________________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
</tr>
</tbody>
</table>
1. Finish writing the following Quicksort program:

```pascal
type MyArray = array [0..1000] of integer;

procedure QuickSort( var A : MyArray ; i,j : integer ) ;
var M: integer ; (* index of split *)
i_, j_: integer ; (* helper indices *)
s : integer ; (* splitter value *)
begin
  if i<j then begin
    (* Split A[i..j], setting M such that:
    1) i <= M <= j
    2) A[k] <= A[M], for all i <= k < M
    3) A[M] <= A[k], for all M < k <= j
    *)
      (* recurse *)
      QuickSort( A, i, M-1 ) ;
      QuickSort( A, M+1, j ) ;
    end (* if statment *).
  end ; (* procedure QuickSort *)
```
2. Finish writing the following tree insert program:

```pascal
type tree = ^tree_rec
  tree_rec = record
    left, right : tree ;
    data : integer ;
  end ;

procedure tree_insert( var t : Tree ; v : integer ) ;
begin
  if (t<>NIL) then begin
  end (* if *)
  else begin
    new(t) ;
    t^.left := NIL ;
    t^.right := NIL ;
    t^.data := v ;
  end (* else *)
end ; (* procedure tree_insert *)
```
3. Rewrite the following function, using only recursion (no loop constructs):

```pascal
function sum(N) : integer ;
var i : integer ;
begin
  {Precondition: N is any integer.}
  i := 0 ;
  repeat
    i := i + 1
  until (i*i) > N ;
  sum := i
end ;
```