The Exam

Problem 1: Termination

(15 points) Consider the Dafny method below:

```dafny
method termination(a: nat, b: nat)
{
    var x, y := a, b;
    while x + y > 0
        decreases ?
        {
            if x > y + 1 {
                x, y := x - 1, y + 1;
            } else if y > x + 1 {
                x, y := x + 1, y - 1;
            } else {
                x, y := x + y - 1, 0;
            }
        }
}
```

What is the appropriate decreases clause (termination measure) that proves the above loop terminating?
Problem 2: Iterative Correctness

(15 points) Consider the Dafny method below:

```drazn
method SumOfOdds(n: nat) returns (sum: nat)
    ensures exists k :: sum == k * k
{
    var i: nat := 0;
    sum := 0;
    while i < n
        invariant ?
        {
            sum := sum + 2 * i + 1;
            i := i + 1;
        }
}
```

What is the appropriate loop invariant that will prove the postcondition as stated?

Problem 3: Recursive Correctness

(20 points) Consider the recursive function defined on sequences of natural numbers below:

```drazn
function maxSeq(s: seq<nat>): nat
    decreases |s|
    requires |s| > 0
    requires ?
    ensures forall k :: 0 <= k < |s| ==> s[k] <= maxSeq(s)
{
    if |s| == 1 then s[0]
    else
        var i :| 0 <= i < |s| - 1;
        if s[i] < s[i + 1] then maxSeq(s[i+1..])
            else maxSeq(s[..i+1])
}
```

Provide the missing precondition such that the postcondition of function maxSeq is correct for all inputs satisfying your precondition together with the ones already given. Be as liberal as possible. Full marks will go to a precondition that will make this code work for as many input sequences as possible.