This test consists of 3 questions on 5 pages (including this one). When you receive the signal to start, please make sure that your copy of the test is complete.

Please answer questions in the space provided. You will earn 20% for any question you leave blank or write "I cannot answer this question," on. You will earn substantial part marks for writing down the outline of a solution and indicating which steps are missing.

If you’re worried that the staple holding these sheets together will fail, write your student number at the bottom of pages 2-5 of this test.

Good Luck!
**QUESTION 1. [6 marks]**

Use any technique you like to guess a closed form for $T(n)$. State your guess, and prove it correct.

$$
\forall n \in \mathbb{N} \quad T(n) = \begin{cases} 
1, & \text{if } n = 0 \\
2T(n-1), & \text{if } n > 0
\end{cases}
$$

**Sample solution:**

$P(n) : T(n) = 2^n$. Claim: $\forall n \in \mathbb{N}, P(n)$.

**Proof — Base case:** If $n = 0$, then $T(n) = 1$ (by definition), which is $2^0$, so $P(0)$ is true.

**Induction step:** Assume $n$ is an arbitrary natural number, and that $P(n)$ is true. Then

$$
T(n + 1) = 2T(n) \quad \text{[definition of } T(n), \text{ since } n + 1 > 0] \\
= 2 \times 2^n \quad \text{[by assumed } P(n)\text{]} \\
= 2^{n+1}
$$

So, $\forall n \in \mathbb{N}, P(n) \Rightarrow P(n + 1)$.

I conclude that $\forall n \in \mathbb{N}, P(n)$.
QUESTION 2. [8 marks]

String s is a palindrome means for every natural number i from 0 to \( \text{len}(s) - 1 \), \( s[i] == s[\text{len}(s) - 1 - i] \). Prove that if the precondition for \( \text{palCheck}(s) \) is true, the postcondition will be satisfied.

def palCheck(s):
    # Precondition: s is a python string
    1. if len(s) < 2 : return True
    2. else :
    3.     s_inner = s[1:len(s) - 1] # subarray s[1..len(s)-2]
    4.     return s[0] == s[len(s) - 1] and palCheck(s_inner)
    # Postcondition: Returns True if s is a palindrome, False otherwise

SAMPLE SOLUTION: \( P(n) \) if \( \text{palCheck}(s) \) is executed on a string \( s \), of length \( n \), then it returns true if s is a palindrome, false otherwise.

Claim: \( \forall n \in \mathbb{N}, P(n) \).

COMPLETE INDUCTION. Assume \( n \) is an arbitrary natural number, and that \( P(0) \land \ldots \land P(n - 1) \) are true.

CASE \( n = 0 \): The only string \( s \) of length \( n = 0 \) is the empty string. Since there are no indices \( i \) from 0 to 0-1, it is vacuously true that for every natural number from 0 to \( \text{len}(s) - 1 \), \( s[i] == s[\text{len}(s) - 1 - i] \), and \( s \) is a palindrome. In this case, the condition on line 1 succeeds, and True is returned, so \( P(0) \) holds.

CASE \( n = 1 \): There is only one index \( i = 0 \) in 0..\( \text{len}(s) - 1 = 0 \), and \( s[0] == s[\text{len}(s) - 1 - 0 = 0] \), so \( s \) is a palindrome. In this case, the condition on line 1 succeeds, and True is returned, so \( P(0) \) holds.

CASE \( n \geq 2 \): Since \( n \geq 2 \), lines 3 and 4 execute, and \( s\_inner \) is a string of length \( n - 2 \geq 0 \). Line 4 returns True if and only if \( s[0] == s[n-1] \) (and vice versa) and \( \text{by} \ P(n-2) \), for every \( j \) from 0 to \( n-3 \), \( s\_inner[j] == s\_inner[n-3-j] \). Since \( \text{by} \ \text{the slice operator} \ s\_inner[j] == s[j+1] \), \( P(n-2) \) says that for every \( i \in [1, n - 2] \), \( s[i] == s[n-2-j] == s[n-1-i] \). Thus, line 4 returns True iff for all \( i \) in 0..\( n + 1 \), \( s[i] == s[n-1-i] \), that is iff \( s \) is a palindrome.

Since it is true in all three possible cases, \( \forall n \in \mathbb{N}, P(0) \land \ldots \land P(n - 1) \Rightarrow P(n) \). Conclude, \( \forall n \in \mathbb{N}, P(n) \).


**QUESTION 3.** [10 marks]

Read over the definition of `downward(n)`, below. The line numbers are not part of the python code, but were added for your convenience.

**PART (A) [8 marks]**

Prove that if the precondition is true, the loop in `downward(n)` terminates. You may want to state, and prove, a loop invariant that helps with your proof.

```python
def downward(n):
    # Precondition: n is a positive natural number
    # loop invariant:
    # loop invariant:
    1. m = 0
    2. while (not(n == 0)) :
    3. n = n/2 # this is integer division
    4. m = m + 1
    # Postcondition:
    # Postcondition:
    5. return m
```

**Sample solution:** # loop invariant: n is a natural number.

\[ P(i) : \text{If there is an } i \text{th iteration of the loop, then } n_i \in \mathbb{N}. \]

Claim: If the precondition is true, \( \forall i \in \mathbb{N}, P(i). \)

Proof (simple induction on \( i \)): Assume the precondition. Base case: If \( i = 0 \), then \( n_0 \) is guaranteed to be a natural number by the precondition, so \( P(0) \) holds.

**Induction Step:** Assume \( i \) is an arbitrary natural number and that \( P(i) \) is true. If there is an \( (i + 1) \)th iteration of the loop we know that \( n_i \neq 0 \), and by \( P(i), n_i \in \mathbb{N} \), so \( n_i \geq 1 \), and \( n_{i+1} = n_i/2 \geq 0 \), and integer division guarantees that \( n_i/2 \) is an integer. Thus, \( n_{i+1} \in \mathbb{N} \), and \( \forall i \in \mathbb{N}, P(i) \Rightarrow P(i + 1). \)

Conclude that \( \forall i \in \mathbb{N}, P(i). \)

If there is an \( (i + 1) \)th iteration of the loop, then (as observed above), \( n_i \geq 1 \), so \( 2n_i = n_i + n_i > n_i \), so \( n_i > \lceil n_i/2 \rceil = n_{i+1} \). Combining this with \( P(i) \) shows that the sequence \( \langle n_i \rangle \) is a strictly decreasing sequence of natural numbers, which (by the Principle of Well-Ordering) must be finite. This, in turn, implies there is a finite number of loop iterations — the loop terminates.
Part (b) [2 marks]

State, without proof, a postcondition that gives the value of \( m \) in terms of \( n \), after the loop terminates.

Sample solution: If \( n > 0 \), \( m = \lfloor \log_2(n) \rfloor + 1 \). Otherwise \( m = 0 \).

# 1: _____/ 6

# 2: _____/ 8

# 3: _____/10

TOTAL: _____/24

Total Marks = 24