Name _____ Student No. _____

No aids allowed. Answer all questions on test paper. Use backs of sheets for scratch work. ***Show all your work; there will be no credit for answers without a justification.***

Total Marks 100: four questions, each worth 25.

1. Use induction to prove that for $n \geq 1$,

$$1^3 + 2^3 + 3^3 + \cdots + n^3 = (1 + 2 + 3 + \cdots + n)^2.$$

SOLUTION: Basis case: $n = 1$, then $1^3 = 1^2$. For the induction step:

$$\begin{aligned}
&(1 + 2 + 3 + \cdots + n + (n+1))^2 \\
&= (1 + 2 + 3 + \cdots + n)^2 + 2(1 + 2 + 3 + \cdots + n)(n+1) + (n+1)^2 \\
&= (1^3 + 2^3 + 3^3 + \cdots + n^3) + 2(1 + 2 + 3 + \cdots + n)(n+1) + (n+1)^2 \qquad \text{ind hyp} \\
&= (1^3 + 2^3 + 3^3 + \cdots + n^3) + 2\frac{n(n+1)}{2}(n+1) + (n+1)^2 \\
&= (1^3 + 2^3 + 3^3 + \cdots + n^3) + n(n+1)^2 + (n+1)^2 \\
&= (1^3 + 2^3 + 3^3 + \cdots + n^3) + (n+1)^3
\end{aligned}$$

2. Use the invariance principle to show that if in a group of people each person has at most 3 enemies, then they can be distributed into two groups so that each person has at most one enemy in their group.

   SOLUTION: To solve this problem we must provide both an algorithm and an invariant for it. The algorithm works as follows: initially divide the people into any two groups.

   Let H be the total sum of enemies that each person has in his own group. Now repeat the following loop: while there is an m which has at least two enemies in his own group, move m to the other group (where m must have at most one enemy). Thus, when m switches houses, H decreases. Here the invariant is "H decreases monotonically."

   Now we know that a sequence of positive integers cannot decrease for ever, so when H reaches its absolute minimum, we obtain the required distribution.

3. Write an algorithm which when given a positive integer n (i.e., $n \in \mathbb{N} - \{0\}$), outputs "yes" if $n = 2^k$ (i.e., n is a power of 2), and "no" otherwise.

SOLUTION:

1. $x \leftarrow n$
2. while $(x > 1)$
3. if $(2|x)$ then
4. $x \leftarrow x/2$
5. else
6. stop and return "no"
7. return "yes"

4. Prove that the algorithm you gave in question 3 is correct; give an appropriate pre-condition, post-condition, and loop invariant.

   Remember, *correctness = partial correctness + termination.*

   SOLUTION:

   pre-condition: $n \geq 1$

   post-condition: return "yes" $\iff$ n is a power of 2.

   loop invariant: x is a power of 2 iff n is a power of 2.

   We show the loop invariant by induction on the number of iterations of the main loop. Basis case: zero iterations, and since $x \leftarrow n$, $x = n$, so obviously x is a power of 2 iff n is a power of 2.

   For the induction step, note that if we ever get to update x, we have $x' = x/2$, and clearly x' is a power of 2 iff x is.

   Note that the algorithm always terminates (let $x_0 = n$, and $x_{i+1} = x_i/2$, and apply the LNP as usual). We can now prove correctness: if the algorithms returns "yes", then after the last iteration of the loop $x = 1 = 2^0$, and by the loop invariant n is a power of 2. If, on the other hand, n is a power of 2, then so is every x, so eventually $x = 1$, and so the algorithm returns "yes".