Theory of Programming
midterm exam - sample

1. (6 points)

Let $A = [1..5]$ be a statespace, $S \subseteq A \times (A \cup \{fail\})^{**}$ a program over the statespace A.

$$S = \left\{ \begin{array}{lll} 1 \to < 1,2,5,1 > & 1 \to < 1,4,3,5,2 > & 1 \to < 1,3,2,3,\dots > \\ 2 \to < 2,1 > & 2 \to < 2,4 > & 3 \to < 3,3,3,\dots > \\ 4 \to < 4,1,5,4,2 > & 4 \to < 4,3,1,2,5,1 > & 5 \to < 5,2,3,4 > \\ 5 \to < 5,2,fail > & 5 \to < 5,3,4 > & \end{array} \right\}$$

Let $F \subseteq A \times A$ denote the following problem: $F = \{ (2,1), (2,4), (4,1), (4,2), (4,5) \}$

- Determine the program function of S and its domain.
- Decide whether S is totally correct with respect to the given problem F.

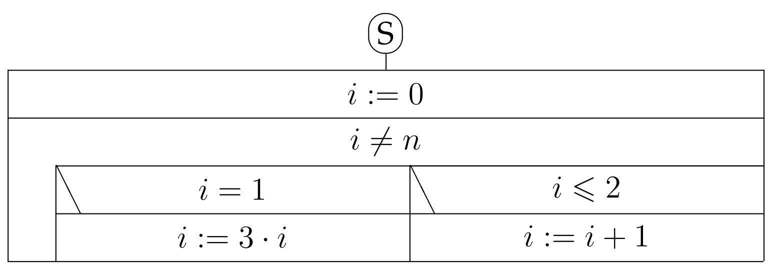
2. (6 points)

$A = (i : \mathbb{N}_0, n : \mathbb{N}_0)$

Write down the sequences that are assigned to the states

- $(1,3)$ and
- $(4,6)$

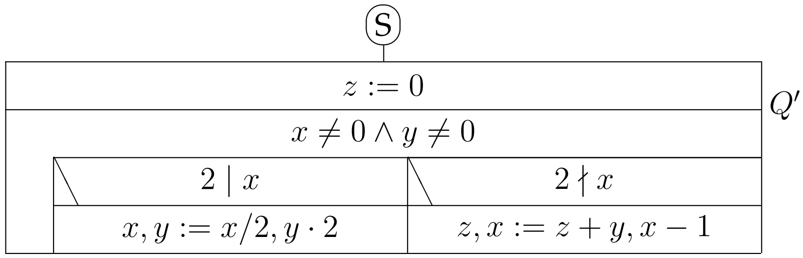
by the following S program.



3. (20 points)

Given the following problem:
A problem is given by its specification:
$A = (x\colon \mathbb{N}_0, y\colon \mathbb{N}_0, z\colon \mathbb{N}_0)$
$Pre = (x = x' \wedge y = y')$
$Post = (z = x' \cdot y')$

Let S denote the following program:



$Inv = (z + x \cdot y = x' \cdot y')$
$Q' = (z = 0 \wedge x \cdot y = x' \cdot y')$
variant function: x

Write down the conditions that are sufficient to prove that the given S program solves the given problem.
Prove that S is a solution of the given problem. Detailed explanation is required.

4. (28 points)

Given an array x of n integer numbers. Calculate the sum of the elements in such a way that if index i is an odd number, then the additive inverse of $x[i]$ has to be considered in the sum.
Examples:

(a) If the input array is x=[1,2,3,4] then the result should be -1+2-3+4=2.
(b) If the input array is x=[-1,-2,-3,4] then the result should be 1+(-2)+3+4=6.
(c) If the input array is x=[1] then the result should be -1.
(d) If the size of the input array is 0 then the result should be 0.

The specifications of the problem is given:
$A = (x : \mathbb{Z}^n, s : \mathbb{Z})$
$Pre = (x = x')$
$Post = (Pre \wedge s = \sum\limits_{i=1}^{n} (-1)^i \cdot x[i])$