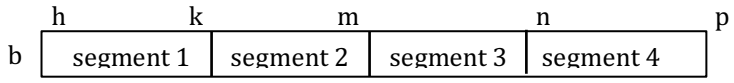
NAME _____

Homework on program correctness: Solutions

Question 1. Write the formula for the number of values in the range b..c: _____c+1–b_____

Question 2. In the video on ranges, we gave a mnemonic for remembering the number of values in a range. Write that formula here: Follower – First

Question 3. Below are four array segments. To the right, using what you wrote in answering question 2, write the number of values in each segment in terms of the relevant variables.



b[h..k] _____k+1–h_____

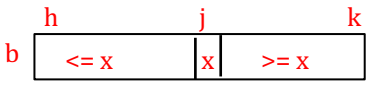
b[k+1..m] _m+1 – (k+1)_____

b[m+1..n–1] __n – (m+1)___

Question 4. State the formula that says segment b[p..q] is empty: __p = q+1_____

Question 5. Below, draw an array diagram that represents this assertion:

b[h..j-1] ≤ x && b[j] = x && b[j+1..k ≥ x]



Question 6. Write down the meaning of the Hoare triple {B} C {D} :

Execution of C in a state in which B is true is guaranteed to terminate, and when it does, D is true.

Question 7. Using the definition of the assignment statement {R[x:= e]} x= e; {R}, calculate the preconditions of the following assignment statements. You do not have to simplify them.

{ (y+1) * y = z }
x= y+1;
{x * y = z}

{ x + 2*x + z = 2*x }
y= 2*x;
{x + y + z = 2*x}

{ x + y + 2 = 8}
y= y+2;
{x + y = 8}

Question 8. Calculate the precondition of the following two sequences of assignments. It's recommended to simplify a precondition after calculating it before moving on the next step. Here's one reason to do that. Since x and y are being replaced in each one, it helps to keep the number of occurrences of them to a minimum. For example, you can rewrite

x = B && y = x + A as x = B && y = B + A .

{ y = B and x = C }	{ x = C and y = B }
	{ x + B = B + C and y = B }
	{ x + y = B + C and y = B }
t= x;	x= x + y;
	{ x = B + C and y = B }
	{ x = B + C and B + C – y = C }
{ y = B and t = C }	{ x – C = B and x – y = C }
x= y;	y= x – y;
	{ x - C = B and y = C }
{ x = B and t = C }	{ x - y = B and y = C }
y= t;	x= x – y;
{x = B and y = C}	{x = B and y = C}

Question 9. We gave the following rule for determining when an if-else statement is correct:

> **Hoare triple for if-else**:
> If {Q && B} S1{R} and {Q && !B} S2 {R}
> then {Q} **if** (B) S1 **else** S2 {R}

Write below a similar rule for determining when an if-statement is correct:

> **Hoare triple for the if-statement**:
>
> If {Q && B} S1{R} and Q && !B => R
>
> then {Q} **if** (B) S1 {R}