QUIZ NAVIGATION

1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22

Show one page at a time

Finish review

Started on	Thursday, 18 November 2021, 2:30 PM
State	Finished
Completed on	Thursday, 18 November 2021, 3:45 PM
Time taken	1 hour 14 mins
Grade	**25.00** out of 25.00 (**100**%)

Question 1

Complete

Mark 0.50 out of 0.50

Flag question

What is the direction of data flow in a DFA formulation of the Reaching Definitions?

forward flow

Question 2

Complete

Mark 0.50 out of 0.50

Flag question

What is the direction of data flow in a DFA formulation of the Available Expressions?

Forward Flow

Question 3

Complete

Mark 0.50 out of 0.50

Flag question

What is the direction of data flow in a DFA formulation of the Live Variable Analysis?

Backward Flow

Question 4

Complete

Mark 0.50 out of 0.50

Flag question

What is the direction of data flow in a DFA formulation of the Copy Propagation?

Forward Flow

Question 5

Complete

Mark 0.50 out of 0.50

Flag question

What is the confluence operator in a DFA formulation of the Reaching Definitions?

Union

Question 6

Complete

Mark 0.50 out of 0.50

Flag question

What is the confluence operator in a DFA formulation of the Available Expressions?

Intersection

Question 7

Complete

Mark 0.50 out of 0.50

What is the confluence operator in a DFA formulation of the Live Variable Analysis?

Union

Flag question

Question 8

Complete

Mark 0.50 out of 0.50

Flag question

What is the confluence operator in a DFA formulation of the Copy Propagation?

Intersection

Question 9

Correct

Mark 2.00 out of 2.00

Flag question

For a DFA for Reaching Definition, consider the following assignments in a block B and those
in other blocks:

==========

Block B

d0: y = 3
d1: x = 10
d2: y = 11

==========

Other Blocks

d3: x = 1
d4: y = 2
d5: z = x
d6: x = 4

==========

Choose the correct KILL[B]:

Select one:

a. {d3, d4, d6}

b. {d0, d3, d4, d6} ✓

c. {d0, d1, d2}

d. {d0, d1, d2, d3, d4, d6}

The correct answer is: {d0, d3, d4, d6}

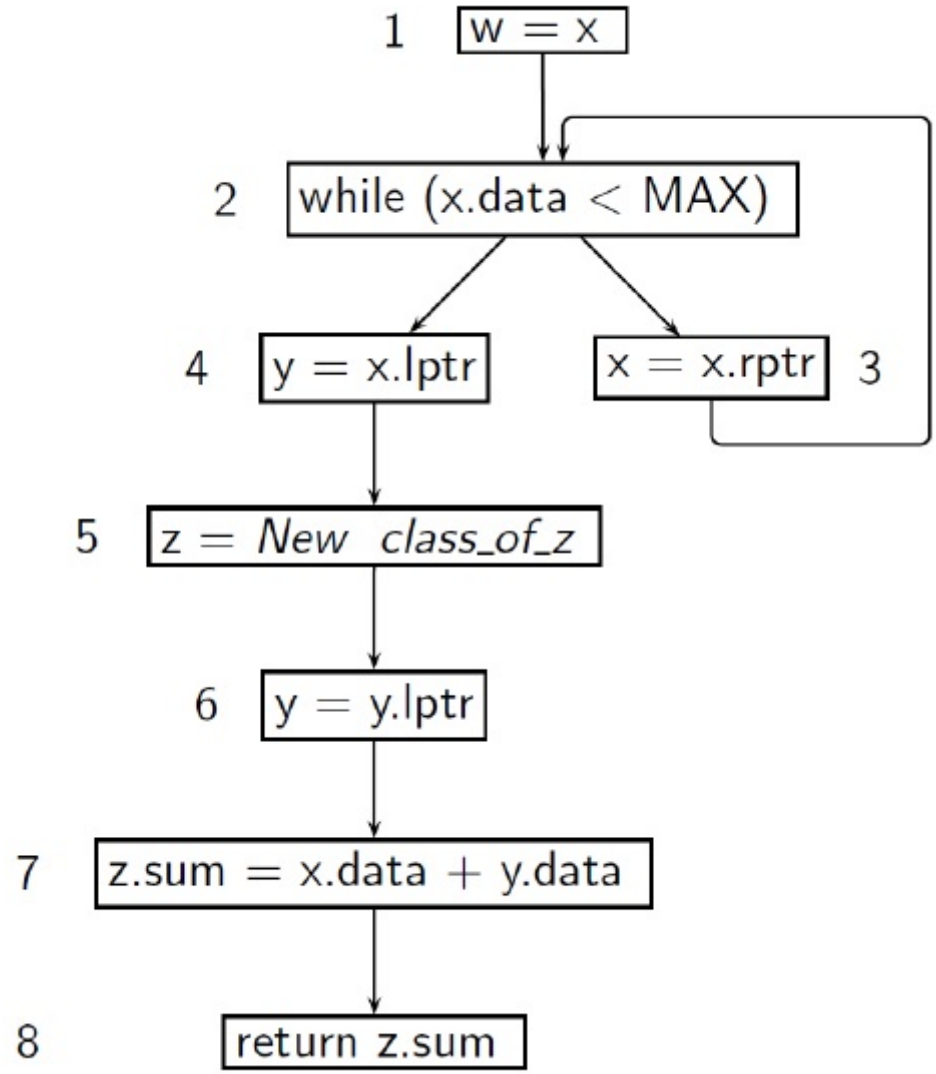
Question 10

Complete

Mark 0.50 out of 0.50

Flag question

Consider the following CFG and match the entry points of the **Block 1** with the list of live variables at that program point. Note that x, y and z are objects some of whose components are accessed / updated.



x, MAX

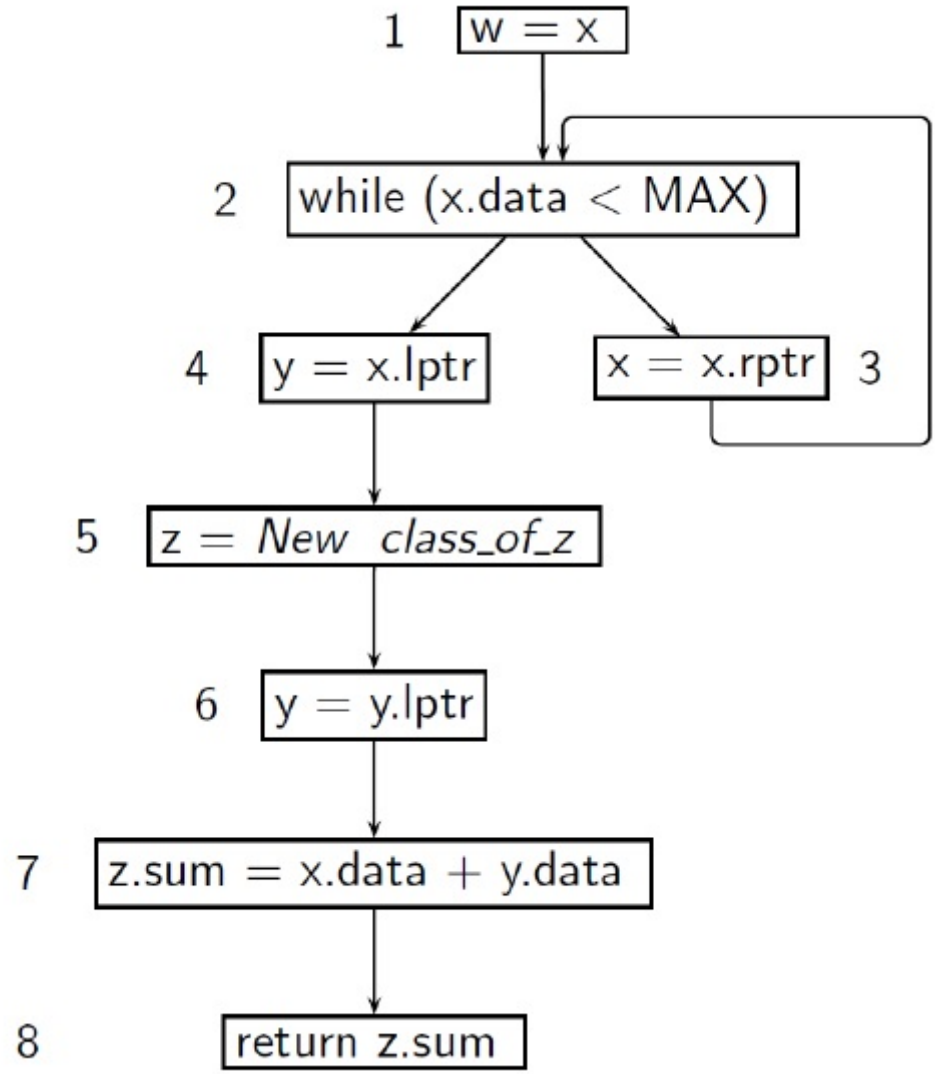
Question 11

Complete

Consider the following CFG and match the entry points of the **Block 2** with the list of live variables at that program point. Note that x, y and z are objects some of whose components are accessed / updated.

Mark 0.50 out of 0.50

Flag question



x, MAX

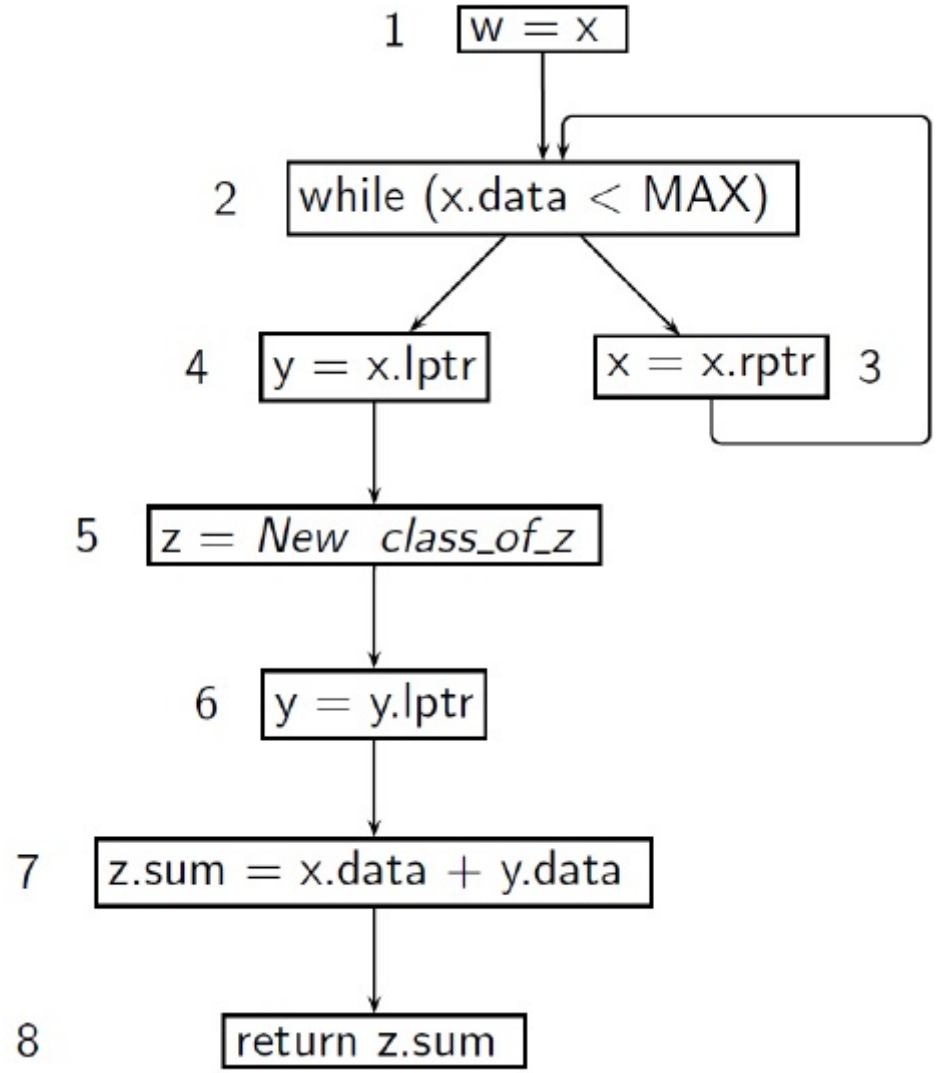
Question 12

Complete

Consider the following CFG and match the entry points of the **Block 3** with the list of live variables at that program point. Note that x, y and z are objects some of whose components are accessed / updated.

Mark 0.50 out of 0.50

Flag question



x, MAX

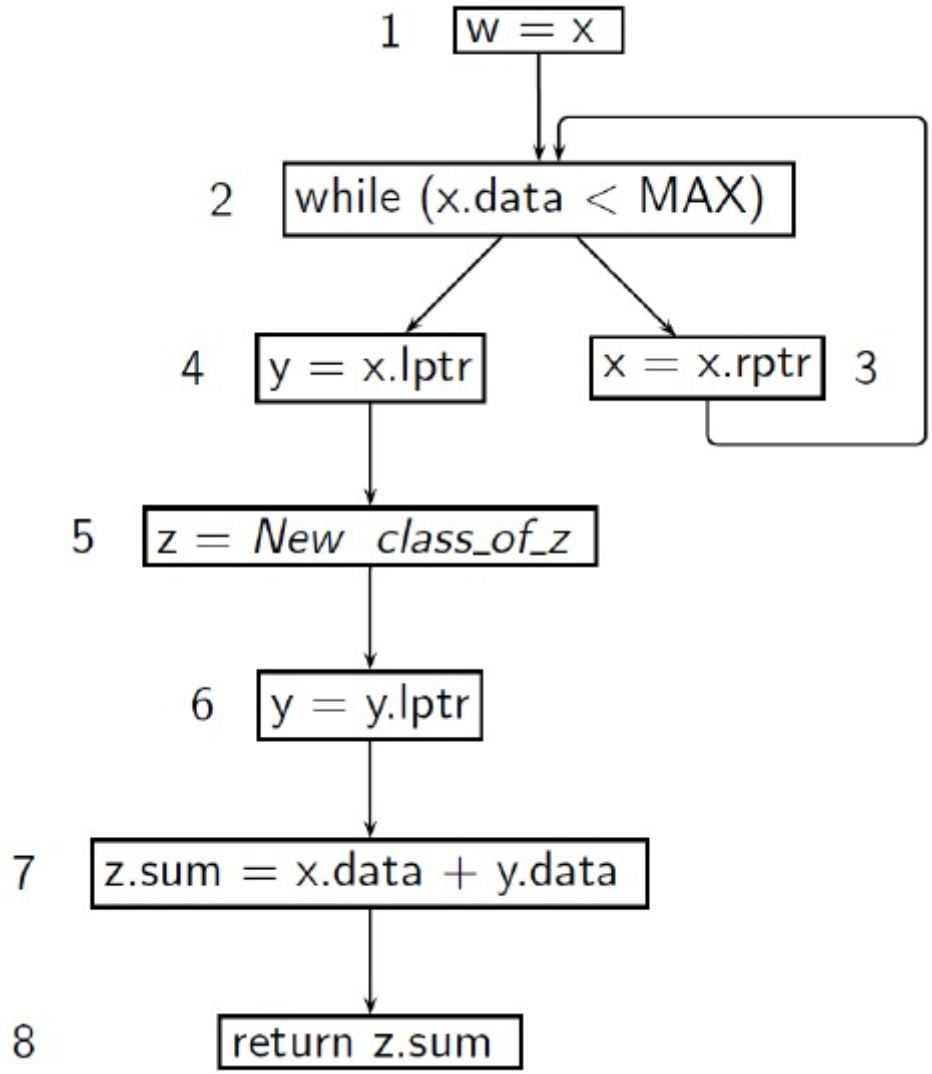
Question 13

Complete

Mark 0.50 out of 0.50

Flag question

Consider the following CFG and match the entry points of the **Block 4** with the list of live variables at that program point. Note that x, y and z are objects some of whose components are accessed / updated.



x

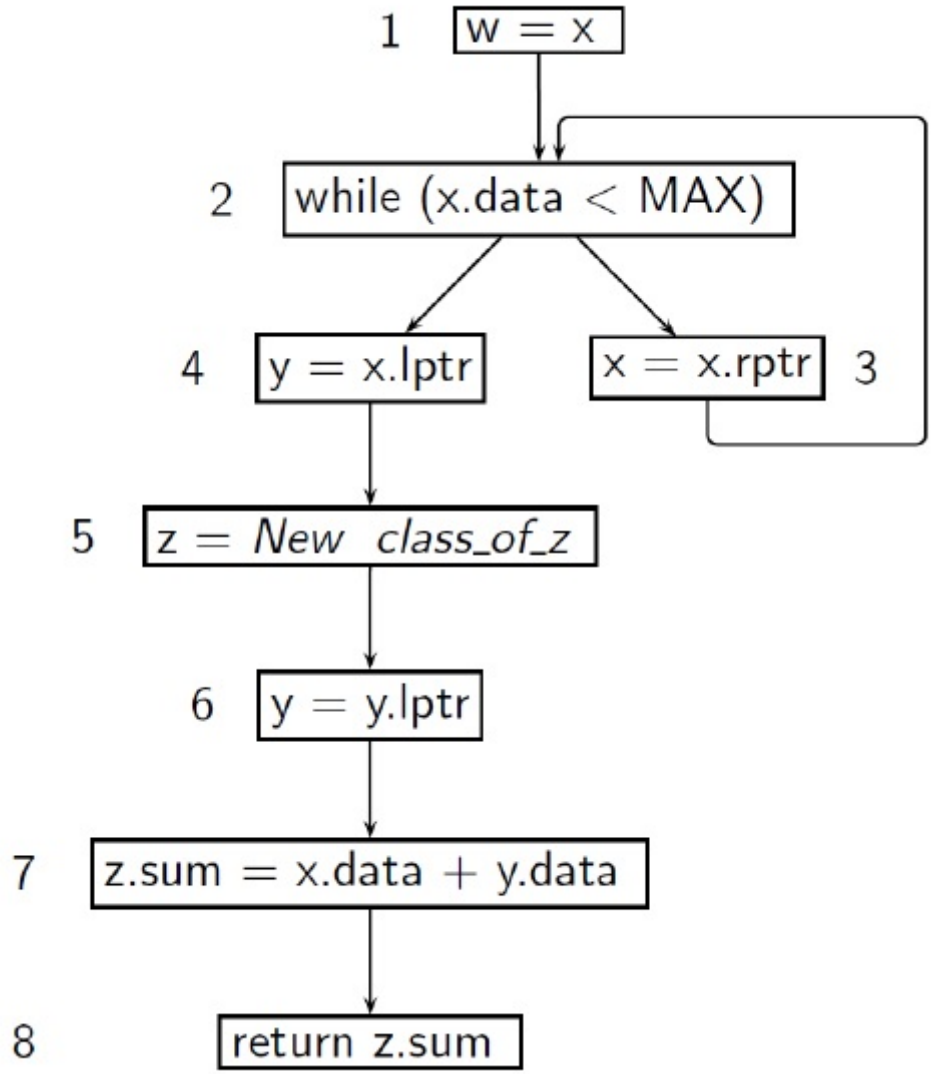
Question 14

Complete

Mark 0.50 out of 0.50

Flag question

Consider the following CFG and match the entry points of the **Block 5** with the list of live variables at that program point. Note that x, y and z are objects some of whose components are accessed / updated.



x,y

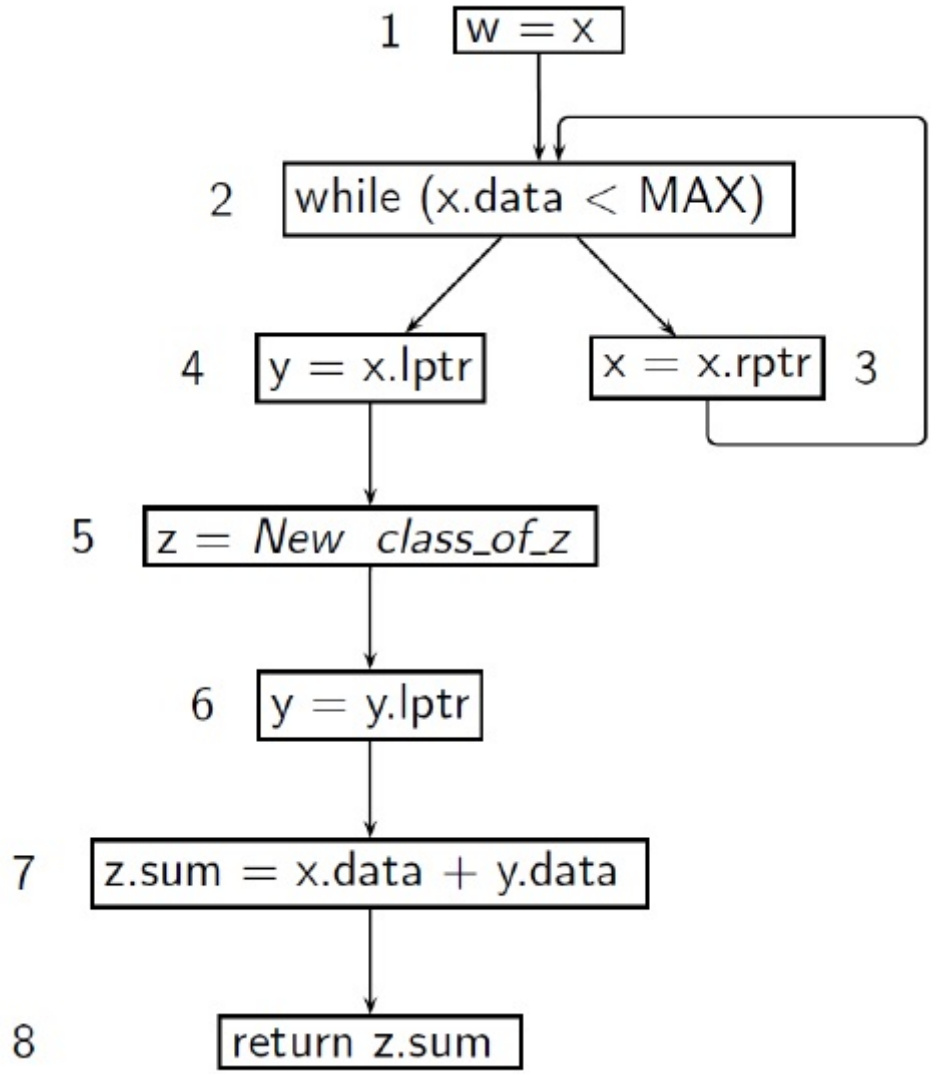
Question 15

Complete

Mark 0.50 out of 0.50

Flag question

Consider the following CFG and match the entry points of the **Block 6** with the list of live variables at that program point. Note that x, y and z are objects some of whose components are accessed / updated.



x,y,z

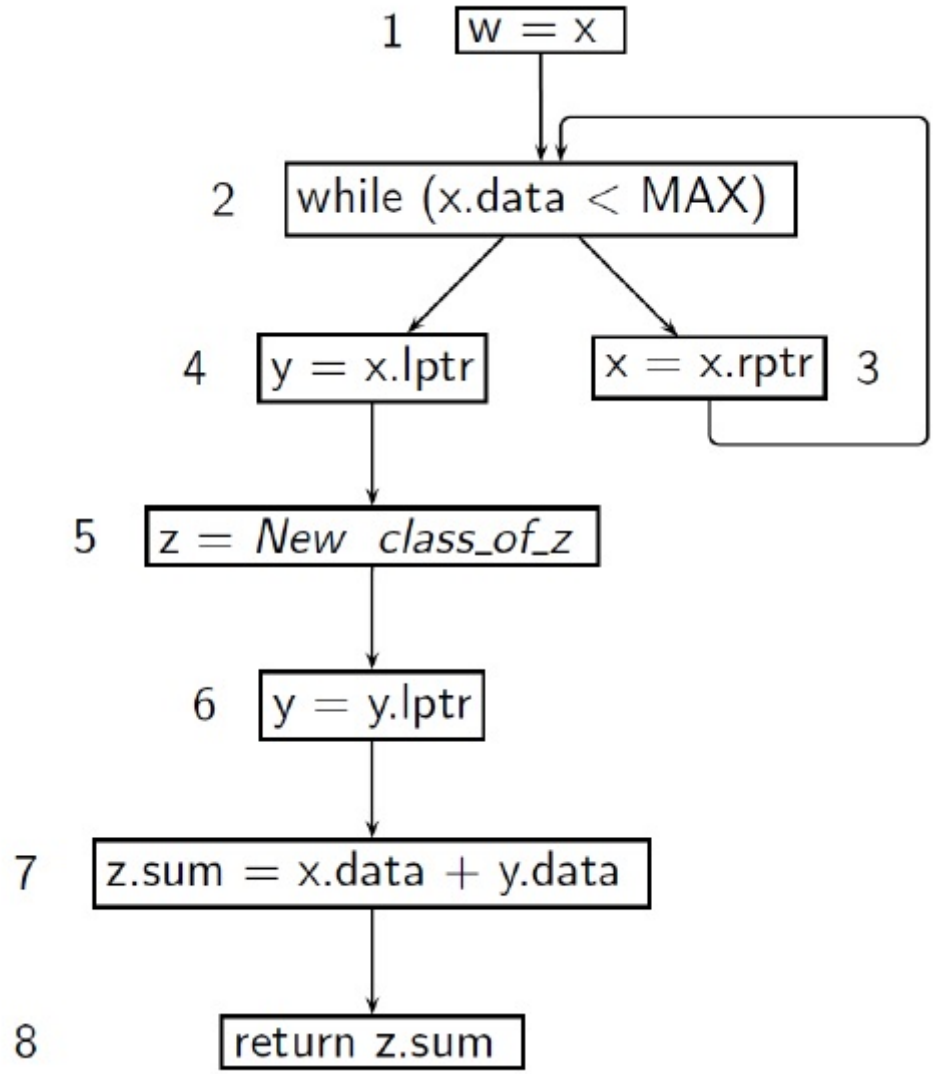
Question 16

Complete

Mark 0.50 out of 0.50

Flag question

Consider the following CFG and match the entry points of the **Block 7** with the list of live variables at that program point. Note that x, y and z are objects some of whose components are accessed / updated.



x,y,z

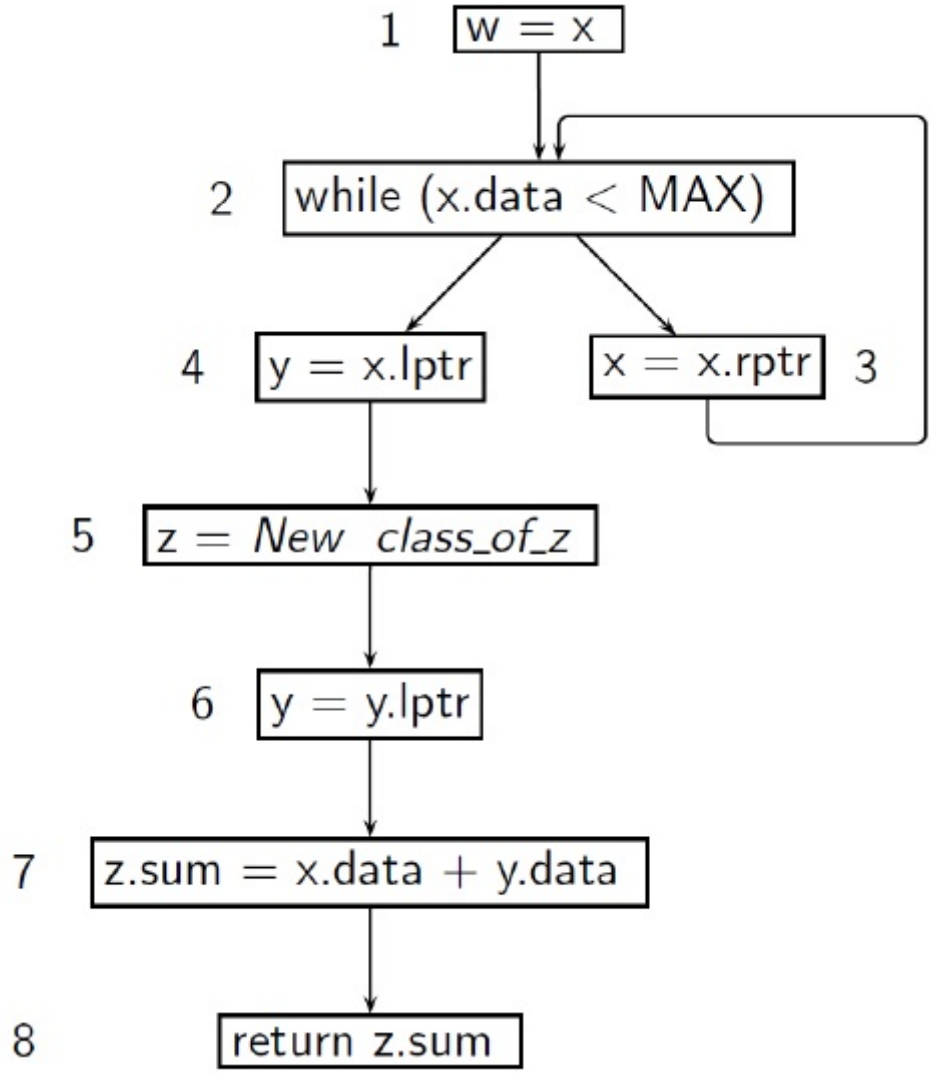
Question 17

Complete

Mark 0.50 out of 0.50

Flag question

Consider the following CFG and match the entry points of the **Block 8** with the list of live variables at that program point. Note that x, y and z are objects some of whose components are accessed / updated.



z

Question 18

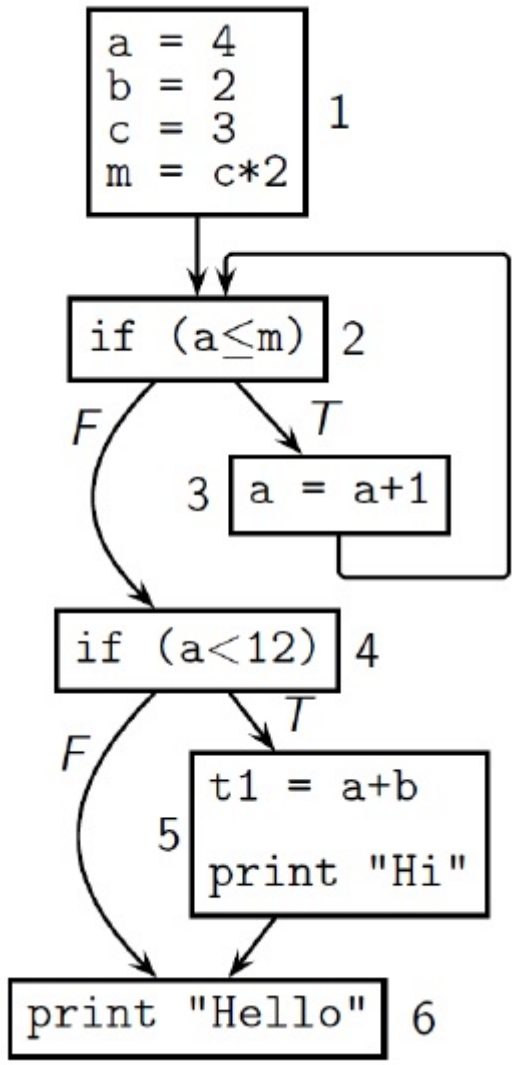
Correct

Mark 1.00 out of 1.00

Flag question

We define a symbol to be Useless in a CFG if its removal from all instructions has no eff ect on the computation.

Identify the useless variable(s) in the following CFG.



Select one or more:

- [ ] a. {m}
- [x] b. {b}
- [ ] c. {c}

d. {a}

e. {t1} ✓

The correct answer is: {b}, {t1}

Question 19

Correct

Mark 2.00 out of 2.00

Flag question

Consider the following expression:

(a + b) * (a + b) + (a + b) * (a + b) + (a + b) * (a + b) + (a + b) * (a + b)

What is the least number of registers needed to generate code using the Sethi-Ullman Algorithm?

Answer: 3 ✓

The correct answer is: 4

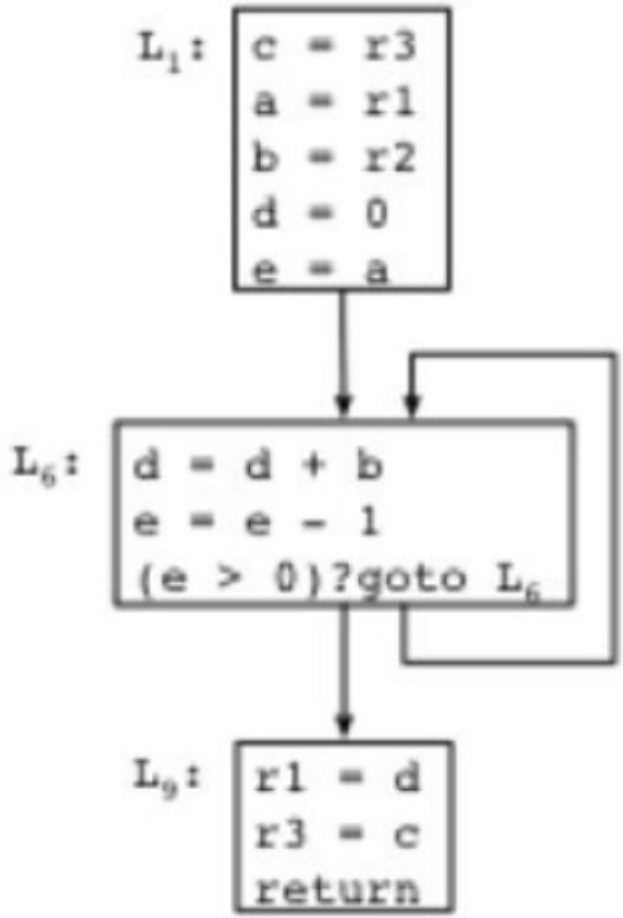
Question 20

Correct

Mark 5.00 out of 5.00

Flag question

Consider the following CFG where there are 3 registers (r1, r2, and r3) and 5 variables (a, b, c, d, and e). These three registers are the only registers available to generate code.



(a) Using Register Interference Graph and *Chaitin's Graph Coloring Algorithm*, determine the minimum number of registers needed to generate code.

(b) If the minimum is 3, you are done. You can generate the code using r1, r2, and r3.

(c) If the minimum is more than 3, you need to spill. So decide on the variable(s) to spill keeping the load-store overhead to the minimum.

(d) Choose the right option below based on your register allocation.

Select one:

- a. Min. Registers = 4, Variable to Spill = c, Load-Store Overhead = 2 ✓
- b. Min. Registers = 4, Variable to Spill = b, Load-Store Overhead = 2 * r1

c. Min. Registers = 5, Variable to Spill = b and c, Load-Store Overhead = 2 + 2 * r1

d. Min. Registers = 4, Variable to Spill = a, Load-Store Overhead = 0

e. Min. Registers = 3, Variable to Spill = None, Load-Store Overhead = 0

f. Min. Registers = 4, Variable to Spill = d, Load-Store Overhead = 4 * r1

The correct answer is: Min. Registers = 4, Variable to Spill = c, Load-Store Overhead = 2

Question 21

Correct

Mark 2.00 out of 2.00

Flag question

Among the following loops, identify the loop(s) for which a = *p; is a loop invariant.

Select one:

a.
```
const int * const p = 5;
// ...
for(i = 0; i < n; ++i) {
    a = *p;
    // ...
}
```

b.
```
int *p = 5;
// ...
for(i = 0; i < n; ++i) {
    a = *p;
    // ...
}
```

```
int * const p = 5;
// ...
for(i = 0; i < n; ++i) {
    a = *p;
    // ...
}
```

c.

```
const int *p = 5;
// ...
for(i = 0; i < n; ++i) {
    a = *p;
    // ...
}
```

d.

The correct answer is:

```
const int * const p = 5;
// ...
for(i = 0; i < n; ++i) {
    a = *p;
    // ...
}
```

Question 22

Complete

Mark 5.00 out of 5.00

Consider the following code:

Flag question

```
int a[100]; // sizeof(int) = 4

a[99] = 0;
for(i = 99; i > 0; --i)
a[i-1] = a[i] + 1;
```

(a) For a MIPS 5 stage pipeline, schedule with stall to avoid hazards. Assume:

```
R0 = &a[0] in memory
R1 = &a[99] in memory
Load <reg>, <mem> needs 1 cycle stall.
    <reg> <- <mem>
Inc <reg> does not need a stall.
    <reg> = <reg> + 1
Add <reg1>, <reg2>, <const> needs 1 cycle stall.
    <reg1> = <reg2> + <const>
Store <mem>, <reg>, does not need a stall.
    <mem> <- <reg>
Jne <reg1>, <reg2>, <label> needs 1 cycle stall.
    if (<reg1> != <reg2>) go to <label>
There is one cycle delay slot
There are 10 registers (R0 to R9) available
```

(b) Unroll the loop to optimize stalls in the generated code

What is the average number of cycles per iteration?

3

Finish review