

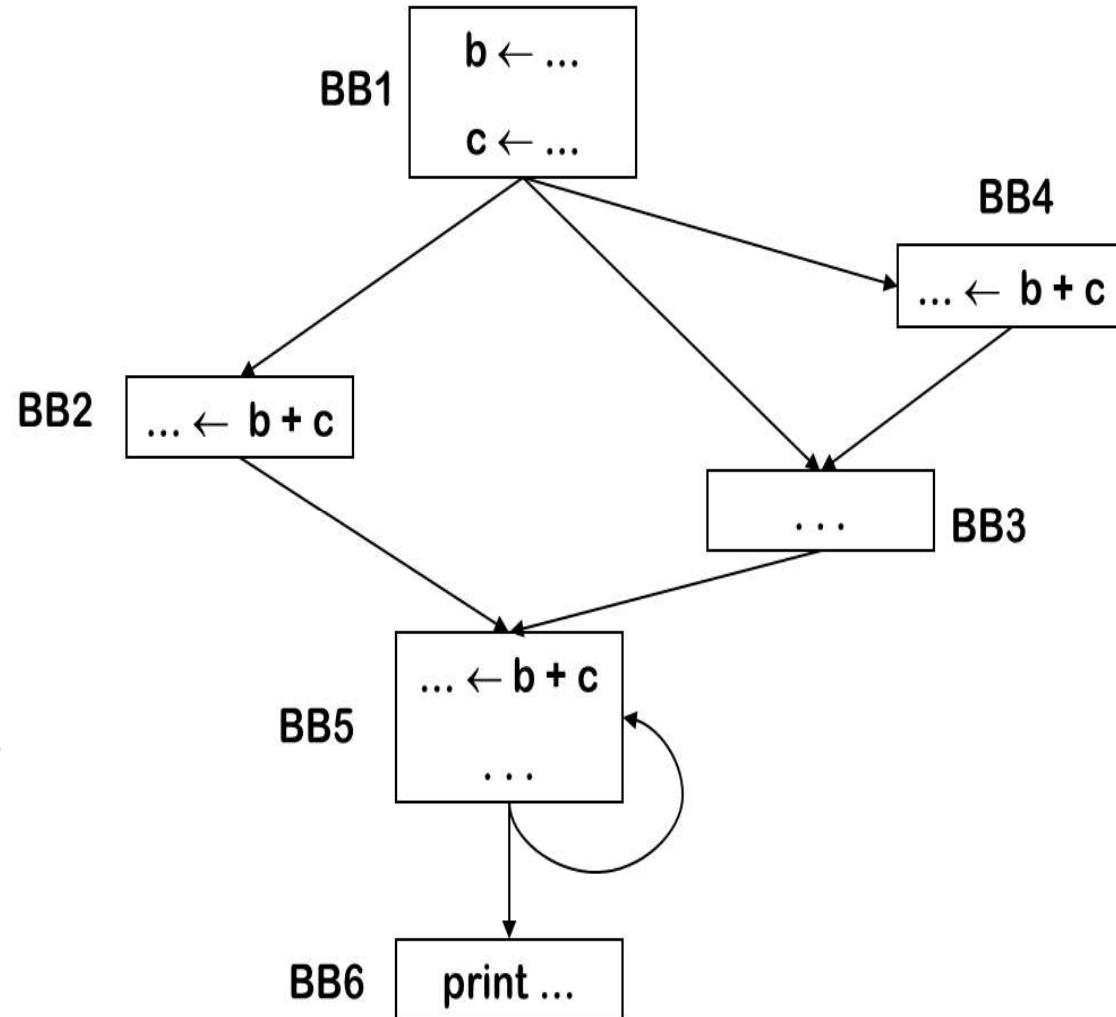
Worksheet–5 Solution

From lecture given on 01/23/2019

Question1.

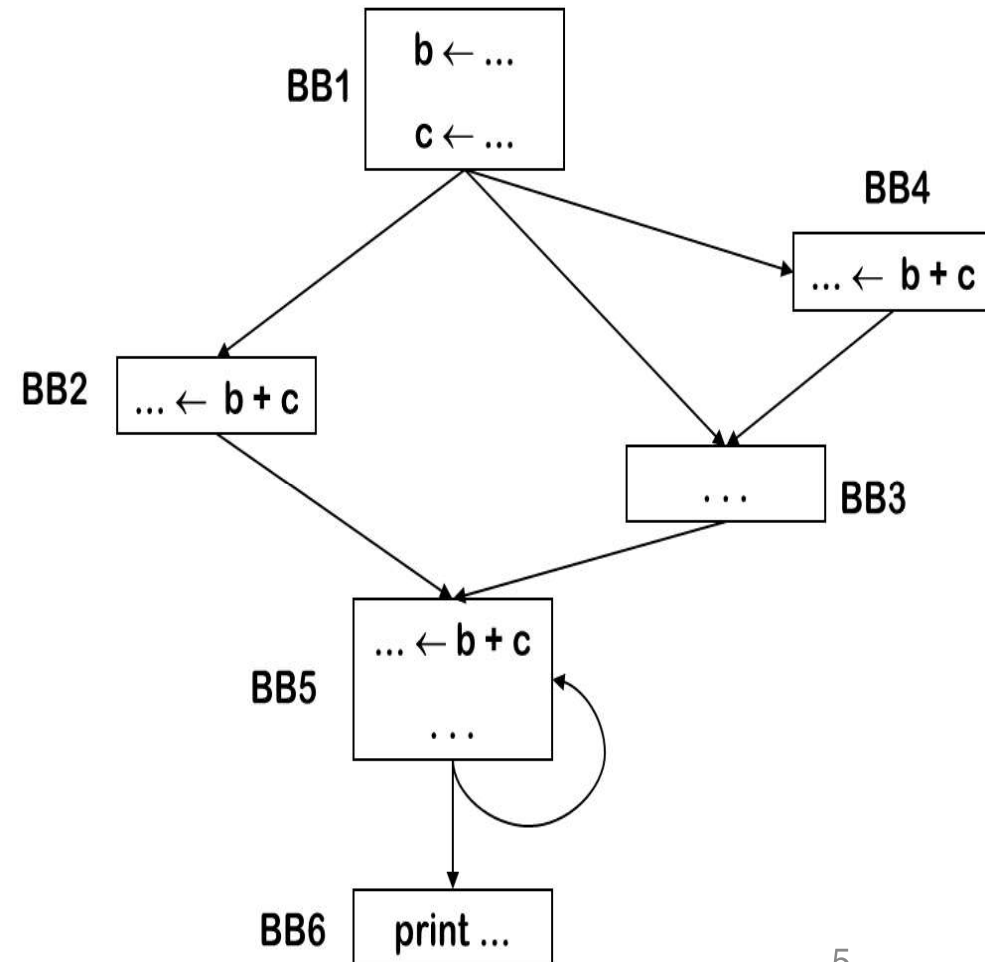
Consider the control flow graph shown below. Indicate where computations of **$b+c$** can be inserted and deleted to minimize the number of times it is computed.

Assume that there are no other defs of b and c , and do not worry about dead code elimination in this example.



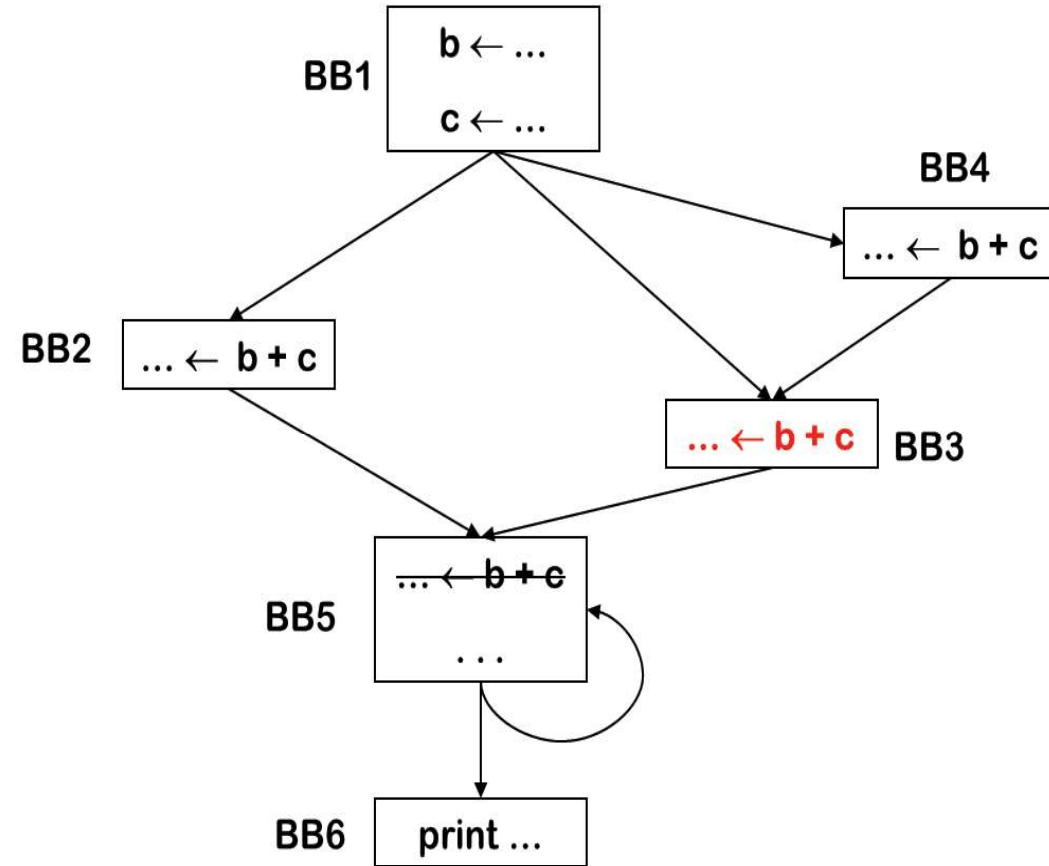
Sample solution : 1st step

- Computation of **$b+c$** in BB5 is partially redundant.
- We can remove the redundancy by moving the computation of **$b+c$** from BB5 to a location before BB5.



Sample solution : 2nd step

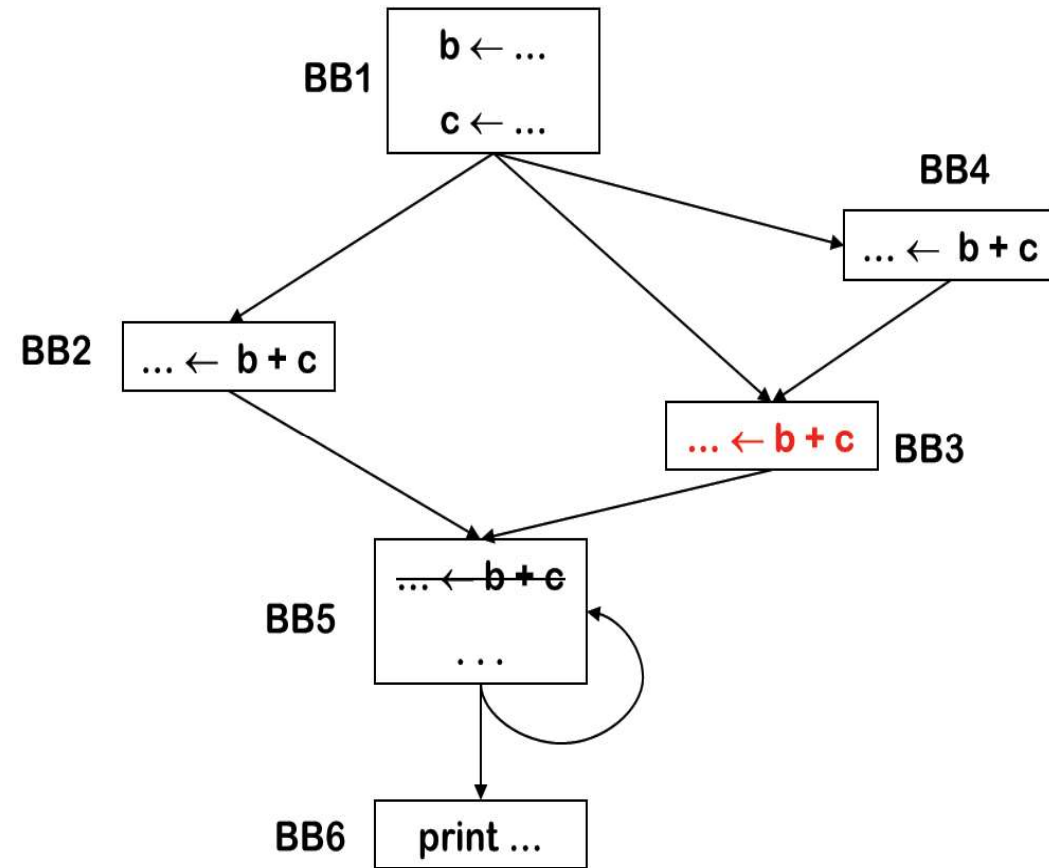
- By moving the computation of **$b+c$** to the end of BB3, redundancy of computing **$b+c$** along the path **$[BB1 \rightarrow BB2 \rightarrow BB5 \rightarrow^* BB6]$** is removed.
- Redundancy still remains along the path **$[BB1 \rightarrow BB4 \rightarrow BB3 \rightarrow BB5 \rightarrow^* BB6]$** .



Sample solution : 2nd step

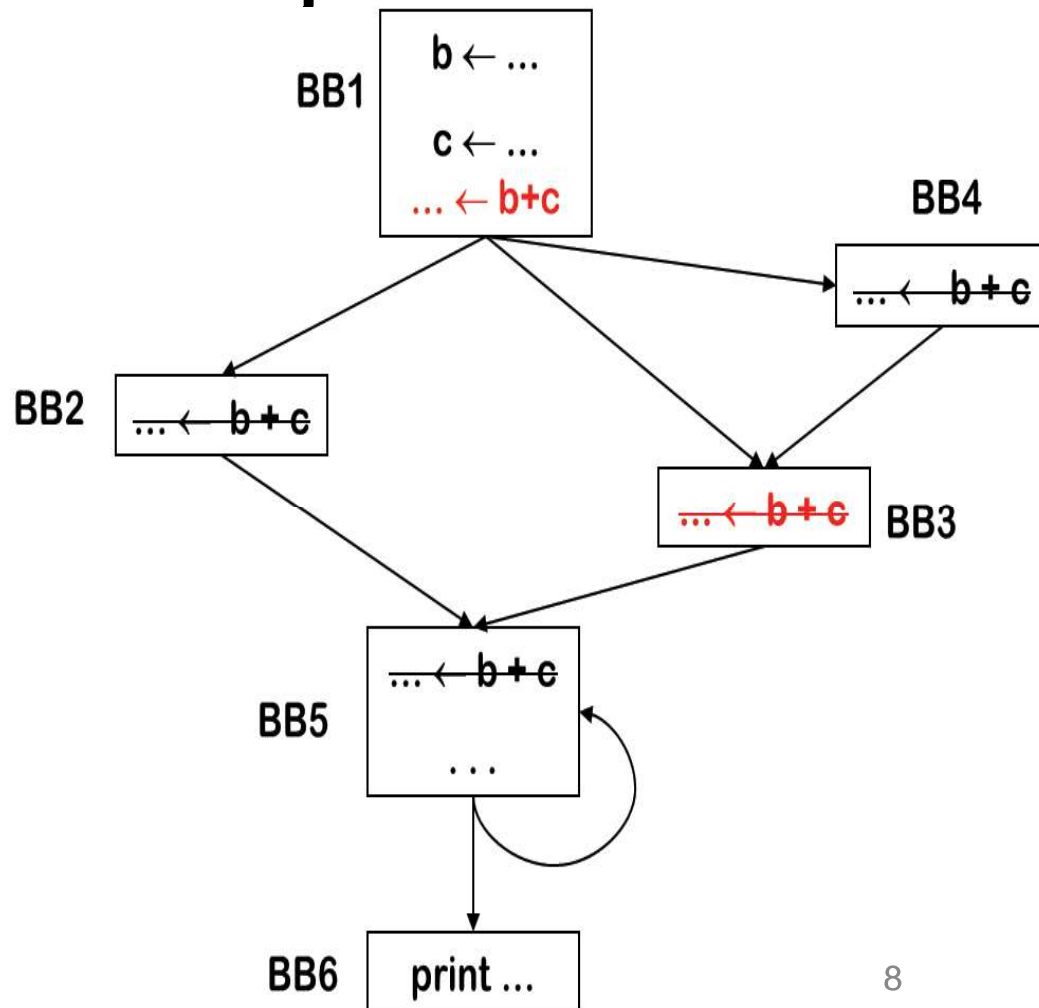
- **$b+c$** is computed on every path that leaves **BB1** and produces the same value at each of those computations.

(= **$b+c$** is an anticipable expression from the end of BB1)



Sample solution : 3rd step

- Since **$b+c$** is anticipable from the end of BB1, it is safe to append the computations of **$b+c$** to the end of BB1, and delete others.
- After the modification, there are no redundancies remaining in any control path.

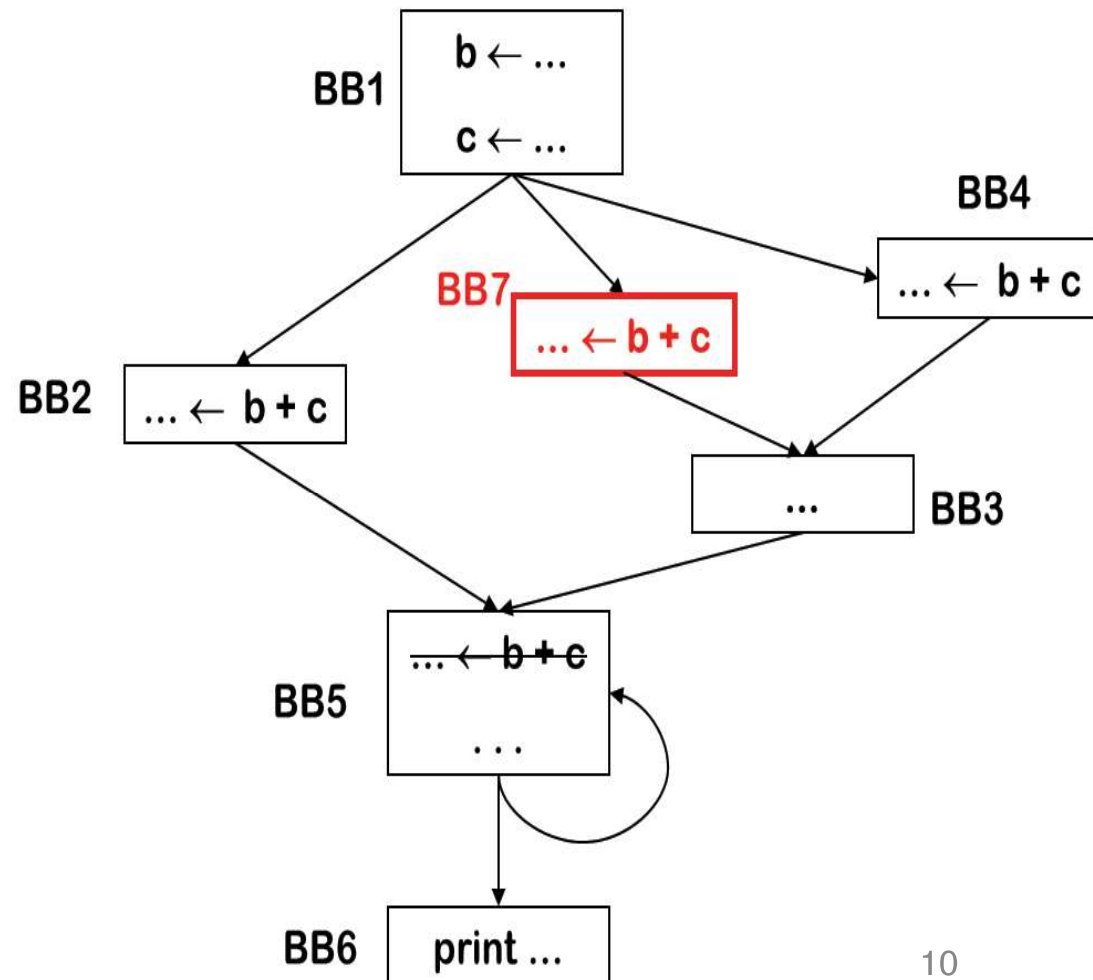


Comments on students' answers

- Almost all students submitted same answers as the sample solution.
- We will discuss some of the different solutions from students.

Alternate student solutions #1

- Computation of **$b+c$** in BB5 is deleted, and a new basic block(BB7) is added with computation of **$b+c$** .
- **$b+c$** is computed only once in every control path, so there are no redundancies.
- However, this solution would result in **longer code length** than the sample solution.



Alternate student solutions #2

- A new basic block(BB7) is added before BB5, computation of **$b+c$** is deleted from BB4 and BB2.
- Adding a new block before BB5 can reduce the redundancy caused by the loop in BB5, but computation of **$b+c$** is done twice in every control flow.

