ECE540S Optimizing Compilers

University of Toronto, Faculty of Applied Science and Engineering
Department of Electrical and Computer Engineering

Midterm: February 28, 2002
25% of the final grade

Student Name: _____

Student Number: _____

Email Address: _____

Score:				
	Q1 =	/ 20	Q2 =	/ 10
	Q3 =	/ 10	Q4 =	/ 10
	Q5 =	/ 5	Q6 =	/ 15
	Q7 =	/ 15	Q8 =	/ 15

TOTAL = / 100

Student Name: _____

Control Flow Analysis (30 points)

1) Use the code below to answer parts A – D. (20 points)

(A) Identify all leader instructions and their corresponding basic blocks in the code below (number the basic blocks). (5 points)

```
S1:  GOTO S3
S2:  IF I > 100 GOTO S7
S3:  J = J + 1
S4:  IF (J > 2) GOTO S6
S5:  I = I + 1
S6:  GOTO S2
S7:  PRINT J
```

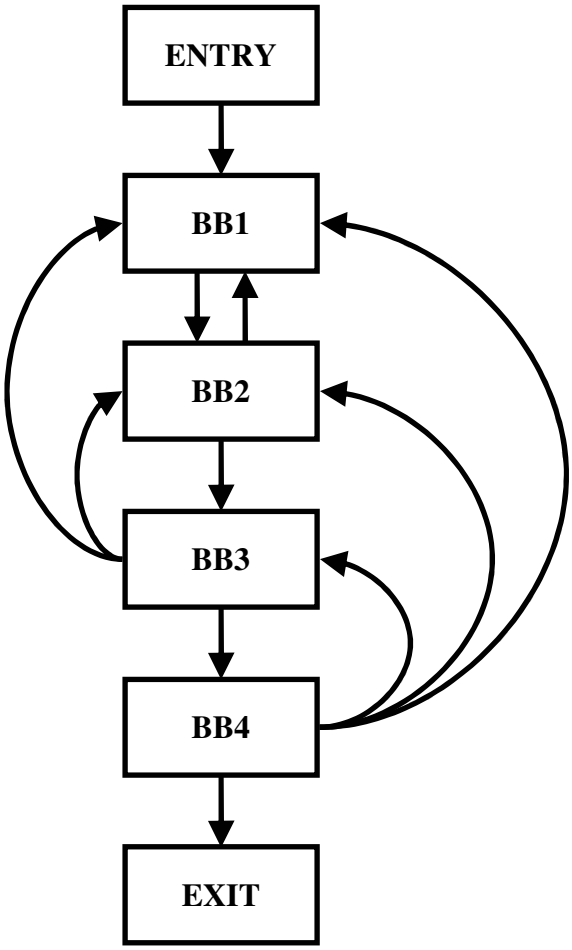
(B) Draw the control flow graph. (5 points)

Student Name: _____

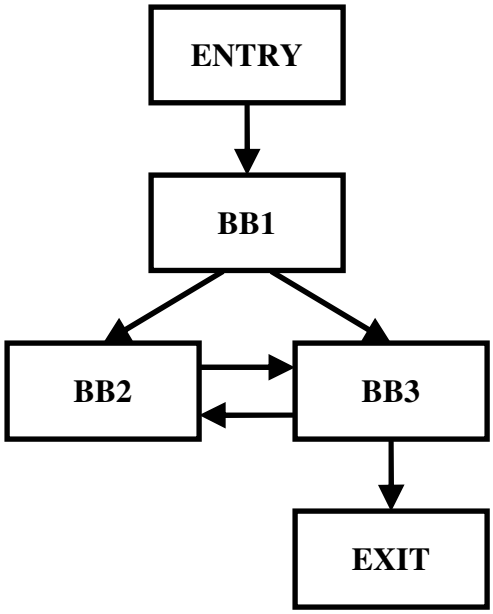
(C) Identify the back-edge in the CFG from (1) and list the nodes contained within its natural loop. (5 points)

(D) Rewrite the code (not the CFG) from Question (1) with a preheader inserted. Clearly identify which instruction(s) is/are the preheader (at a minimum there should be a single empty statement). Do not perform any optimizations; just add the "empty" preheader. All new instructions added should be labeled L1 … L*n*, where *n* is the number of new instructions. Update or add control flow statements as necessary. (5 points)

Student Name: _____



(A)



(B)

2) For each of the CFGs above state whether the graph is reducible or irreducible. (10 points)

(A)

(B)

Student Name: _____

Data Flow Analysis (20 points)

3) The following are data flow equations where BVin(b), Set1(b), Set2(b) and BVout(b) are all bit vectors. Set1(b) and Set2(b) are locally defined sets, that is they are functions only of the basic block (similar to the Gen and Kill sets of Reaching Definitions). Assume that the domain is all possible bit vectors of size N. Answer the following questions based on these equations. (10 points)

$$\mathbf{BVin(b) = Set1(b) \cup (Set2(b) \cap BVout(b))}$$

$$\mathbf{BVout(b) = \bigcap_{j \in Succ(b)} BVin(j)}$$

A) What type of data flow problem is this? (2 points)

(a) An any-path forward-analysis

(b) An any-path backward-analysis

(c) An all-path forward-analysis

(d) An all-path backward-analysis

B) What is the confluence operator for the above framework? (2 points)

C) If N is 4, what is the $\top$ and $\bot$ element of the domain? (3 points)

D) This framework is both monotone and distributive. What does this imply about the framework? (3 points)

Student Name: _____

4) Circle all statements that are true. (10 points)

A) Reaching Definitions (2 points)

a. **is an all-path forward data flow problem**
b. **is an any-path forward data flow problem**
c. **Reaching definition sets can be represented as bit vectors**
d. **A definition reaches a point p if it appears on all paths from Entry to p**

B) Live Variables (2 points)

a. **is a forward-path problem**
b. **is an any-path problem**
c. **Live variable sets can be represented as bit vectors**
d. **A variable is live at point p if it is defined on all paths from p to Exit**

C) Available Expressions (2 points)

a. **is a forward-path data flow problem**
b. **is an any-path data flow problem**
c. **Available expression sets can be represented as bit vectors**
d. **An expression is available at point p if it appears on all paths from p to Exit**

D) Very Busy Expressions (2 points)

a. **is an all-path backward data flow problem**
b. **is an any-path backward data flow problem**
c. **Very Busy Expression sets can be represented as bit vectors**
d. **An expression is very busy if it appears on more than 1 path from p to Exit**

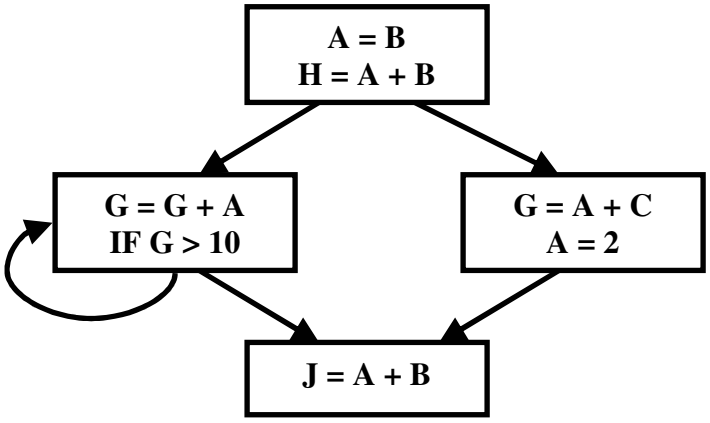
D) A DU chain (2 points)

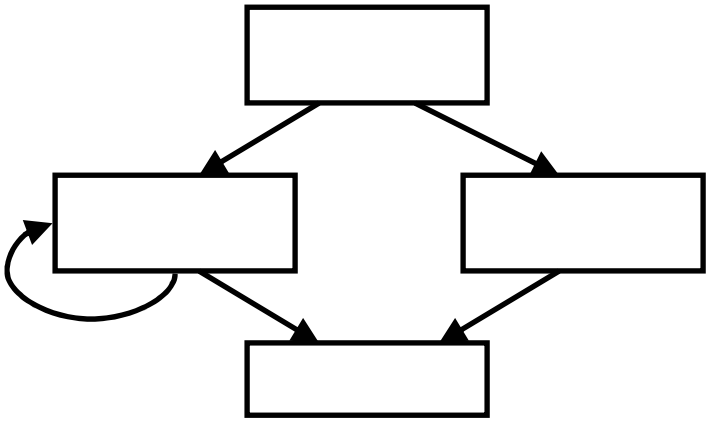
a. **links a use of a variable with all of its possible definitions**
b. **links a definition of a variable with all of its possible uses**
c. **contains uses and definitions that must occur at runtime**

Student Name: _____

Optimizations (50 points)

5) Perform copy propagation on the following graph. Place the resulting code in the graph at the bottom of the page. Perform no other optimizations. (5 points)





Student Name: _____

6) Perform local value numbering on the code below. Next to each line in the original code, provide the value number associated with each variable and expression. Below each line, write the statement after any substitutions made possible through value numbering are done. The first line is done for you. (15 points)

A = B + C *B -> 1; C -> 2; B + C -> 3; A -> 3*
A = B + C

D = E – F

G = B + C

H = A + G

A = A + A

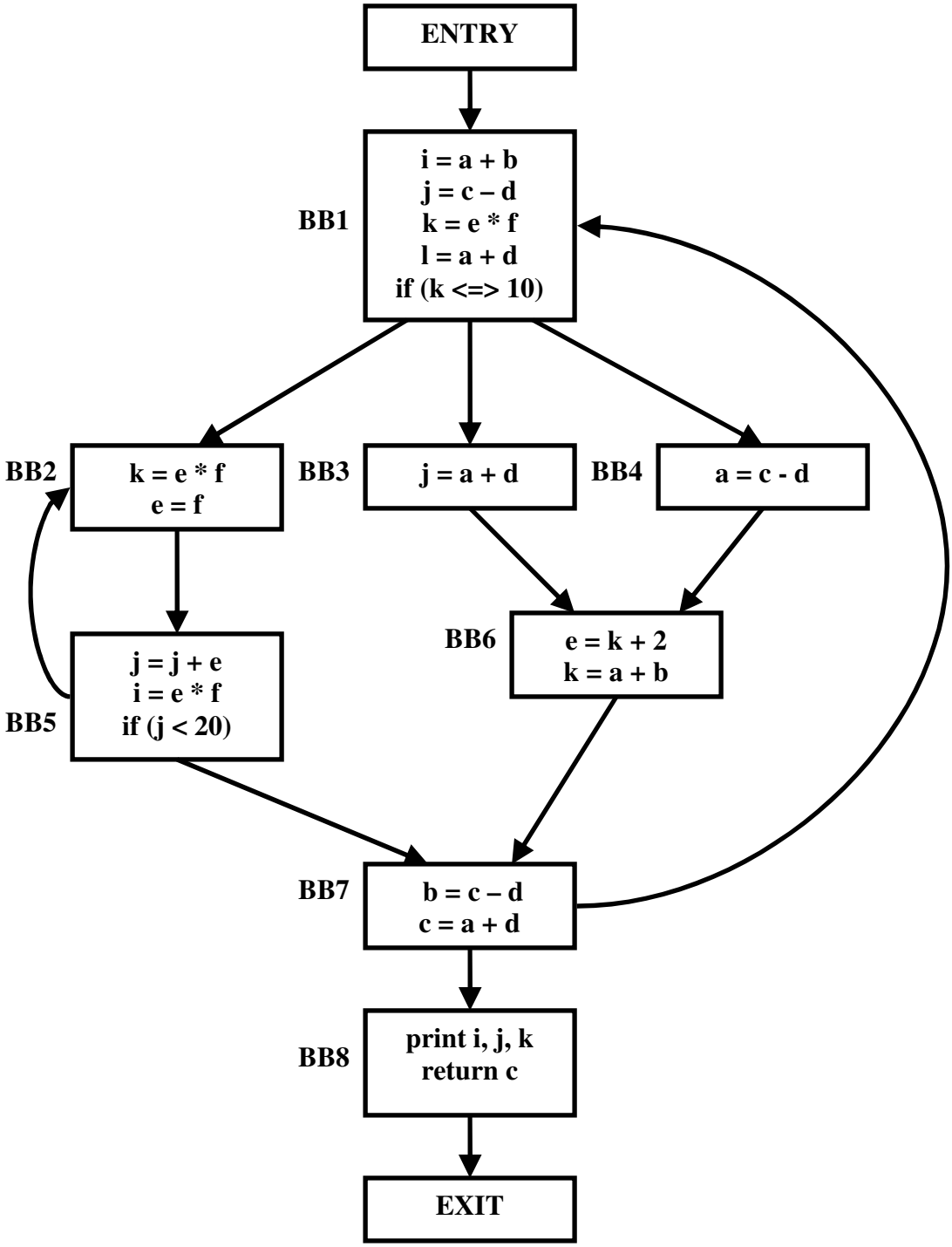
E = G

G = A + A

I = E - F

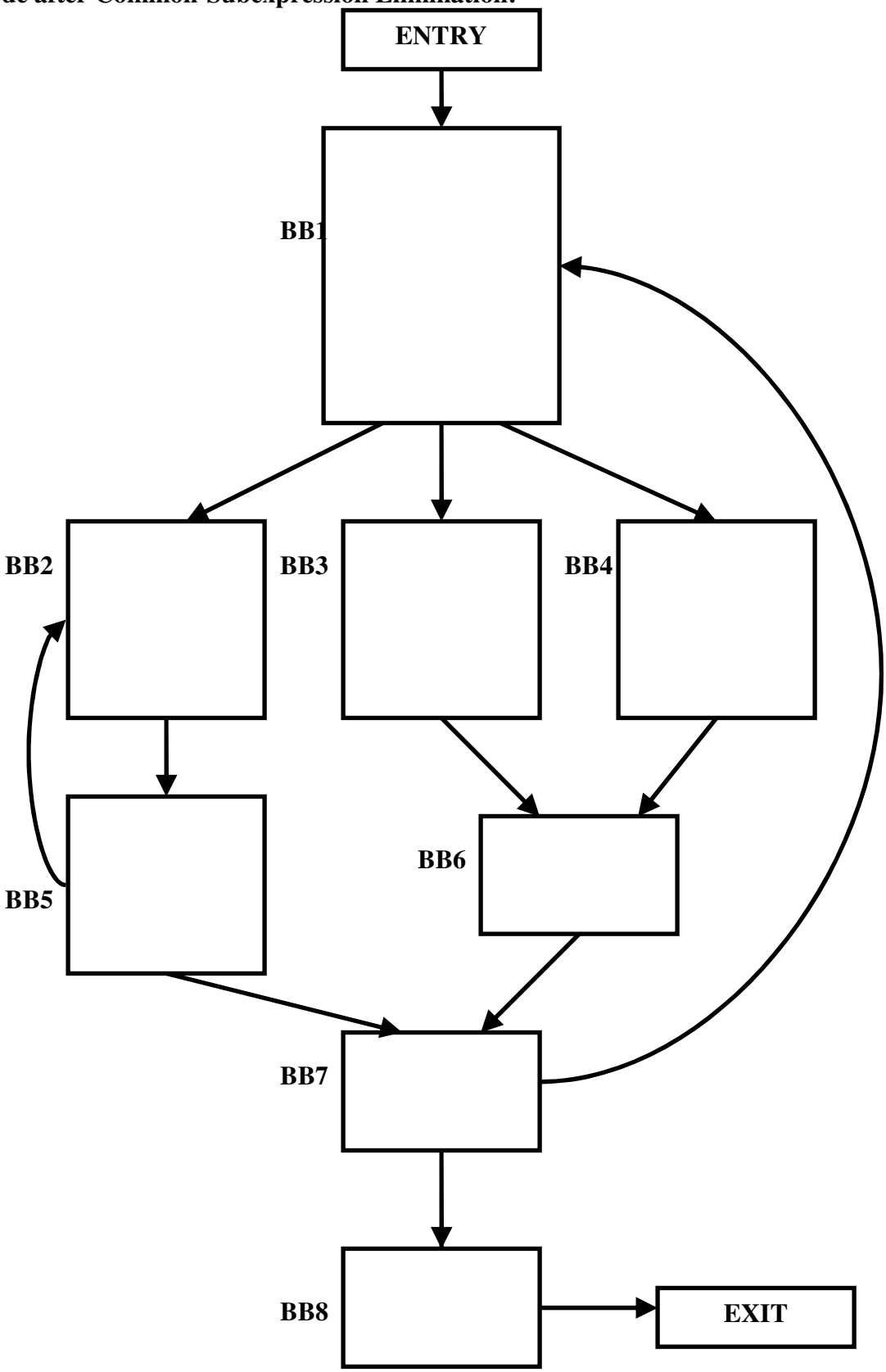
Student Name: _____

7) Perform Common Subexpression Elimination on the code below. For the original graph, identify the set of Available Expressions at the input to each basic block. Write these sets on the graph below. Place the code resulting from CSE in the graph on the next page (if a basic block is unchanged, you may write SAME in the node). Perform no other optimizations, only CSE. (15 points)



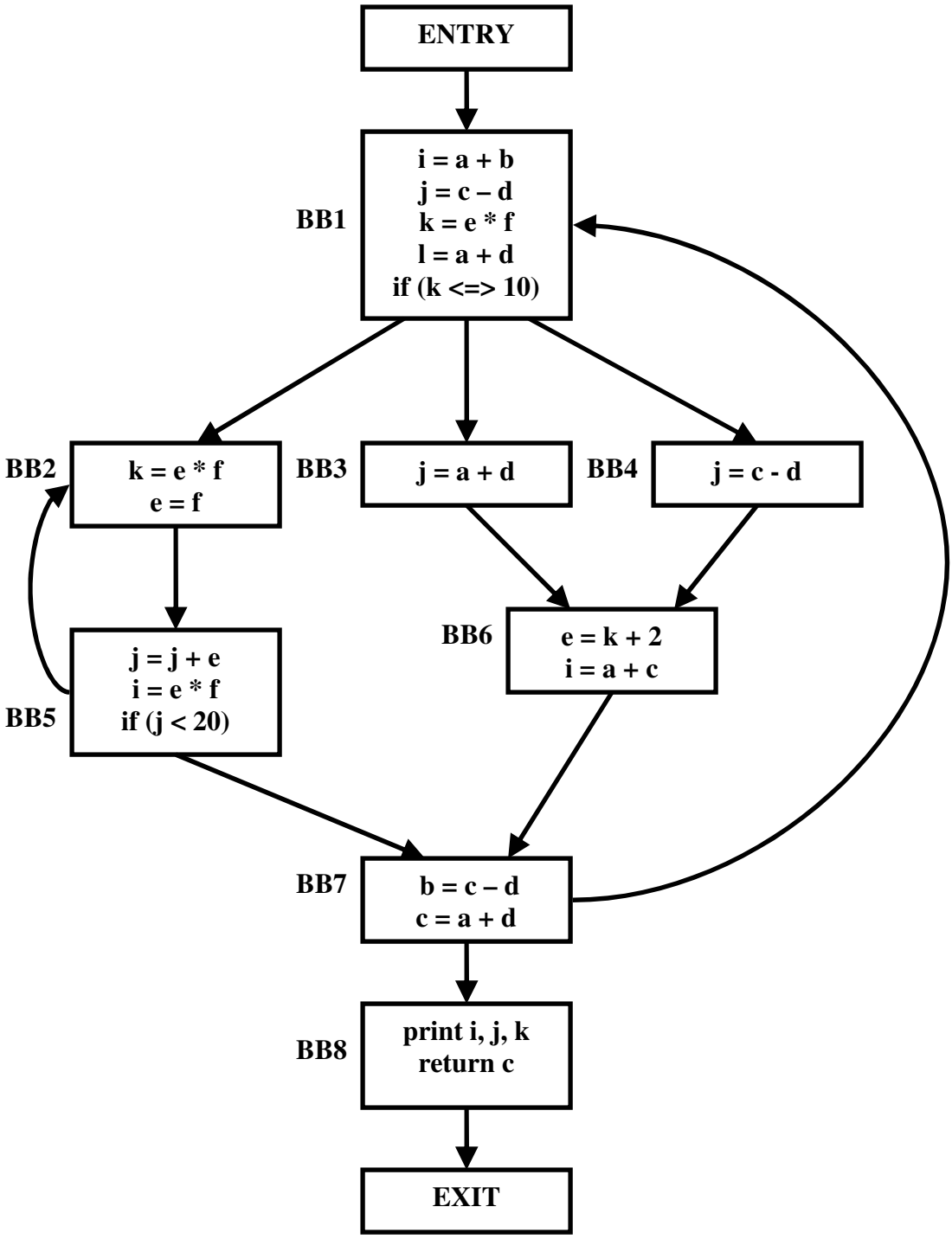
Student Name: _____

Code after Common-Subexpression Elimination:



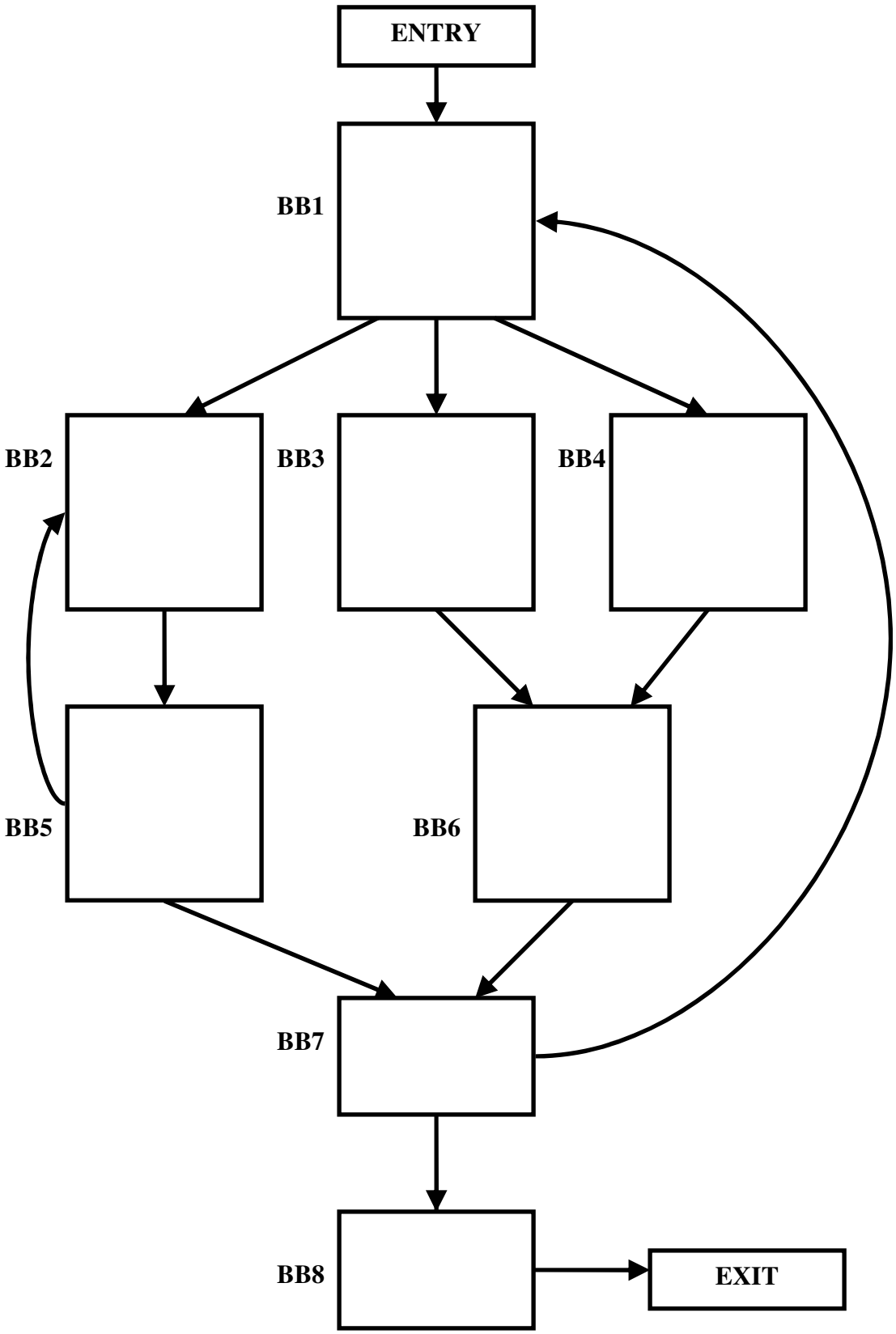
Student Name: _____

8) Perform dead-code elimination on the code below. Show the code resulting from dead-code elimination in the graph on the next page. If a basic block is unchanged, you may write SAME in the node. Perform no other optimizations, only dead-code elimination. (15 points)



Student Name: _____

Code after Dead-Code Elimination:



Student Name: _____

INTENTIONALLY BLANK FOR EXTRA WORK SPACE