Assignment 1

Due Date: 3/10/2023

Time: 23.55

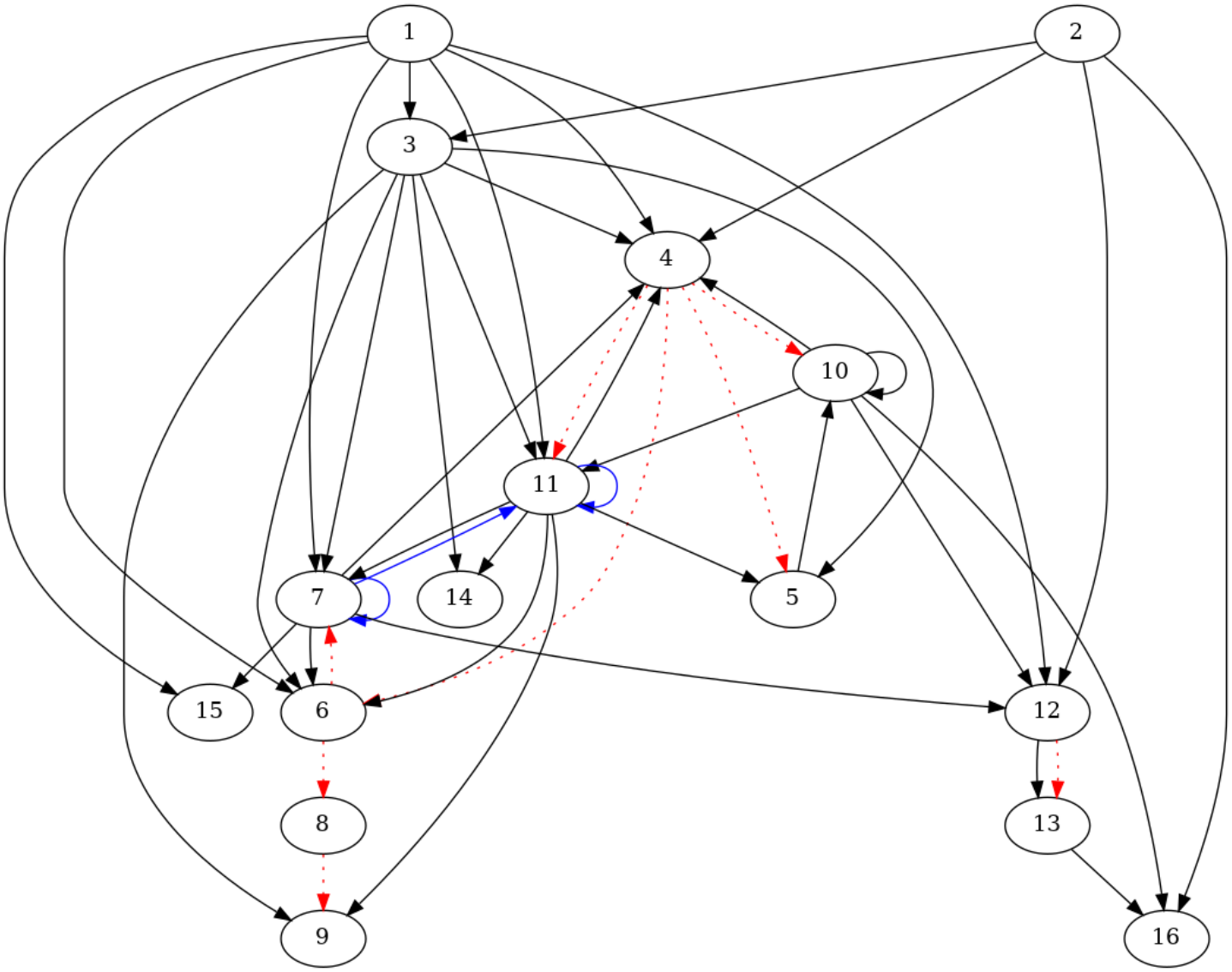
Please note:

This is an individual assignment

No late submissions are accepted.

QUESTION #1 STATIC BACKWARD SLICING

A. Create a static PDG for the program below



```
1. CIN >> B;
2. CIN >> X;
3. A=X+B;
4. WHILE (A<(X+B))

{
5. X=A;
6. IF (A>B)
   {
7.    B=10+A+B;
   }
8.  ELSE
   {
9.      X= A-1;
   }
10.         X--;
11.         A=A-B-X;
}
12.        IF B < X)
{
13.           X=10;
}
14.        COUT <<A;
15.        COUT <<B;
16.        COUT <<X;
```

B. Compute a static slice for the following variables.

Compute Slice (B,15)= { 1,2,3,4,5,6,7,8,9,10,11,15 }

Compute Slice S(X,16) = { 1,2,3,4,5,6,7,8,9,10,11,12,13,16 }

C. COMPUTE A STATIC FORWARD SLICE

for statement 9 => Static slice S(9) ={ 4,5,6,7,8,9,10,11,12,13,14,15,16 }
for statement 13 => Static slice s(13) ={ 13,16 }

Hint: There is no need to create a new PDG, you can reuse the one from question A)

QUESTION #2

PLEASE USE THE PROGRAM SHOWN ON THE LEFT AND COMPLETE THE PROGRAM DEPENDENCIES IN THE TABLE–

```
1.  Y =10
2.  IF (Y)
3.      Y=Y+1;
```

Node	Is **data** dependent on			
		1	2	3
	1			
	2	X		
	3	X		

Node	Is **control** dependent on			
		1	2	3
	1			
	2			
	3		X	

```
1.  CIN >> B;
2.  CIN >> X;
3.  A=X+B;
4.  WHILE (A<(X+B))
    {
5.      X=B+X;
6.      IF (A>(B+C))
        {
7.          B=10+A+B;
        }
8.      ELSE
        {
9.          X= 10;
        }
10.     X--;
11.     A=A-B-X;
      }
12.COUT <<A;
13. COUT <<B;
14. COUT <<X;
```

Data Dependencies

Node		1	2	3	4	5	6	7	8	9	10	11	12	13	14
	1														
	2														
	3	X	X												
	4	X	X	X				X			X	X			
	5	X	X					X			X				
	6	X		X				X				X			
	7	X		X				X				X			
	8														
	9														
	10					X					X				
	11	X		X				X			X	X			
	12			X								X			
	13	X						X							
	14		X								X				

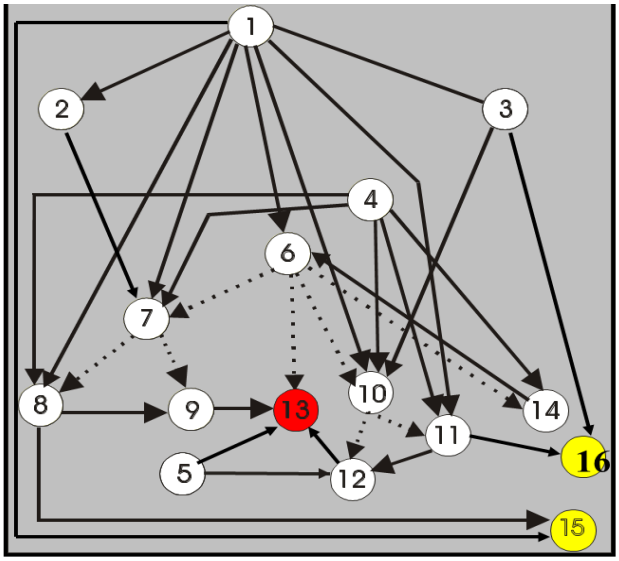
Control Dependencies

Node		1	2	3	4	5	6	7	8	9	10	11	12	13	14
	1														
	2														
	3														
	4														
	5				X										
	6				X										
	7						X								
	8						X								
	9								X						
	10				X										
	11				X										
	12														
	13														
	14														

Question #3

Given is the following program and PDG – identify all the problems in this PDG (wrong/missing dependencies)

```
1    input (n,a);
2    max := a[1];
3    min  := a[1];
4     i := 2;
5    s:= 0;
6    while i ≤ n do
     begin
7         if max < a[i] then
          begin
8              max := a[i];
9              s := max;
          end;
10        if min > a[i] then
          begin
11             min := a[i];
12             s := min;
          end;
13        output (s);
14        i := i +2;
     end;
15   output (max);
16   output (min);
```



Data Dependency

Control Dependency

Any problems within this PDG?

45

Data dependencies:	**Missing: (2->15),(4->6),(14->14),(14->10),(14->7),(14->11),(14->8),(11->10),(8->7)** **Should be removed: (1->15),(5->12)**
Control dependencies:	**Missing: NA** **Should be removed: NA**

Question #4

In a recent department meeting your new boss made the following statement. I just read a research paper which discussed software aging and I am not sure if the claims in the paper are correct. In their paper the authors state that the cause for software aging is: (1.) ignorant surgery – that is modifications being performed to a software product by people who are not necessarily skilled/trained enough to perform such software changes; as well as by (2.) too much movement, that is, software is changed to remove technical debt.

Your boss is asking if you agree/disagree with the two claims made in the paper. Clearly state if you agree/disagree with each claim (1.) and (2.) and briefly justify your decisions (max. 50 words)

Question #4:

Claim #1 – Correct. Ignorant surgery can cause software aging because less skilled programmers may inadvertently introduce errors and design flaws, leading to deterioration of the software's quality and maintainability.

Claim #2 – Incorrect. Software aging is more likely to occur when a system doesn't receive necessary updates to adapt to changing requirements. Implementing changes correctly can actually prevent or mitigate software aging by keeping the software aligned with evolving needs.

Question #5

You were reading in an article the following statement: The major objective of perfective maintenance is to reduce technical debt in a software system.
Is the above statement, correct? Clearly indicate if you agree/disagree. Briefly justify your answer (1-2 sentences)

Question #5:

Disagree - The statement is incorrect. The primary goal of perfective maintenance is to enhance or add new features to a software system. Reducing technical debt is typically addressed through corrective or preventive maintenance, not perfective maintenance.