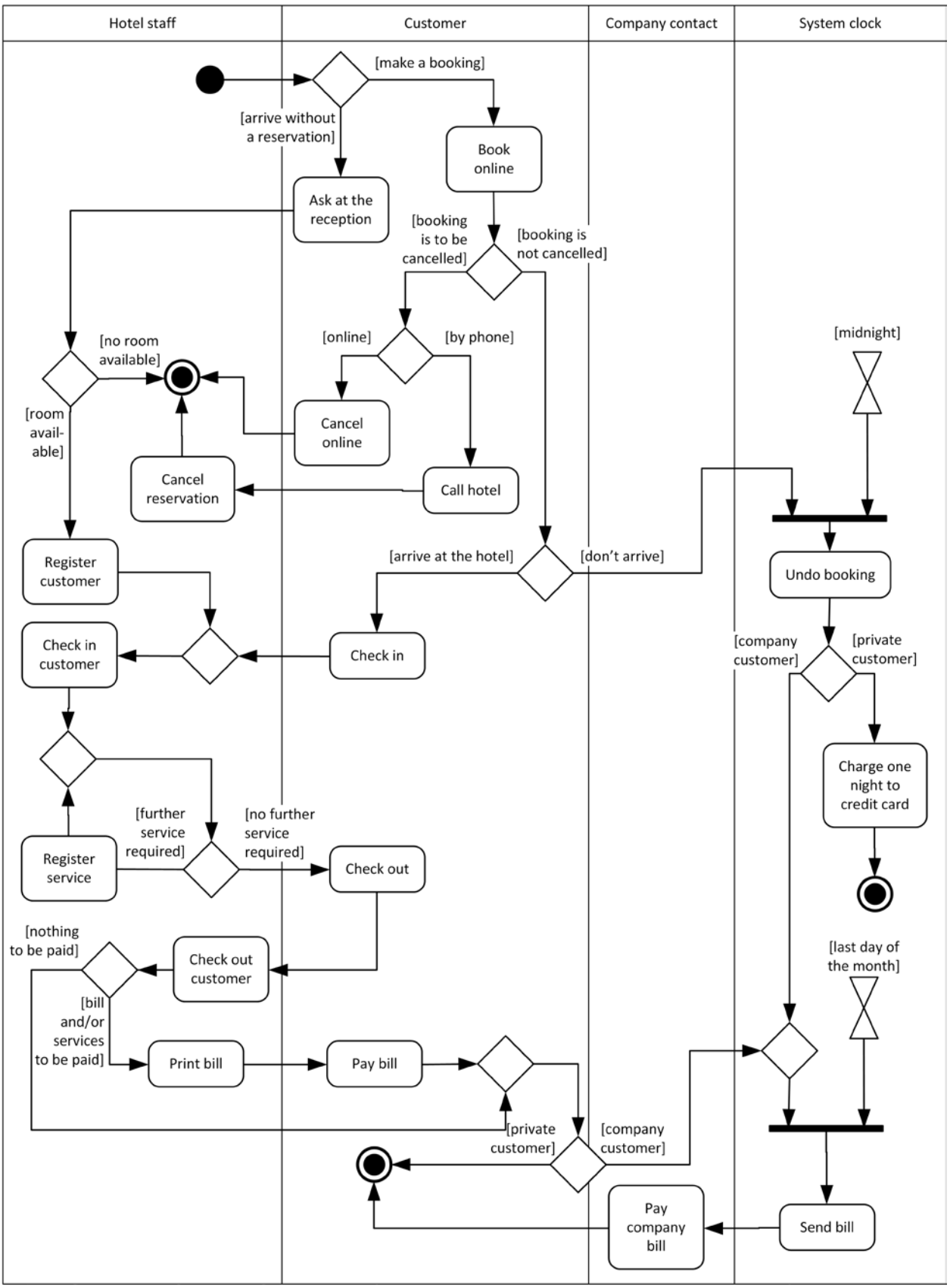
Software Systems Design Test 13 Dec 2018 – Solutions

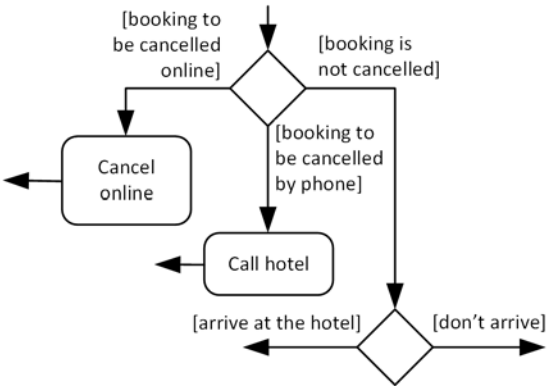
Version 2 (Additions to V1 are in blue)

Question 1 (Activity Diagram) [2.8 points]



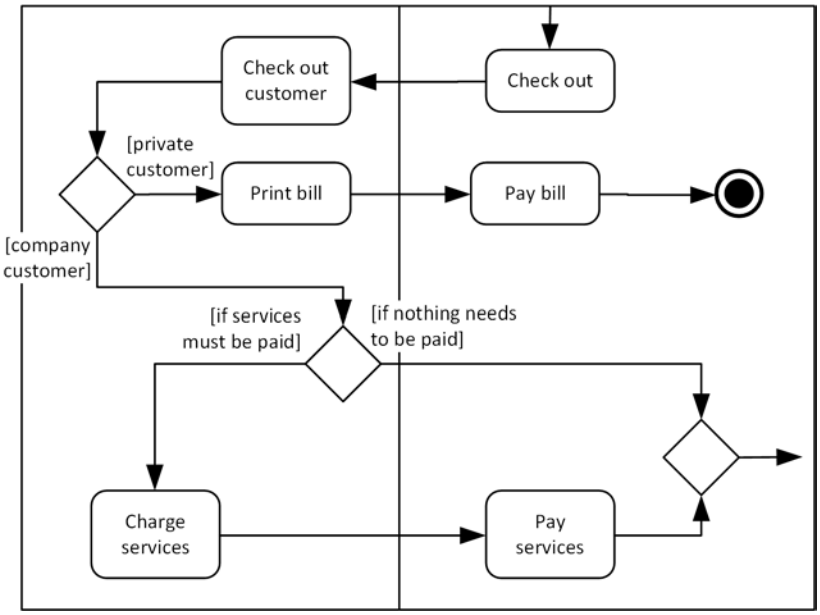
Remarks:

- Using a three-way branch as shown below is also correct (although it seems more logical to split it into two parts as shown above)



- "Assign room, program key" could be modelled as two different activities.

- A different logic is possible for the payment part. It is equally correct, but has more activities.



Question 2 (Use Cases) [1.7 points]

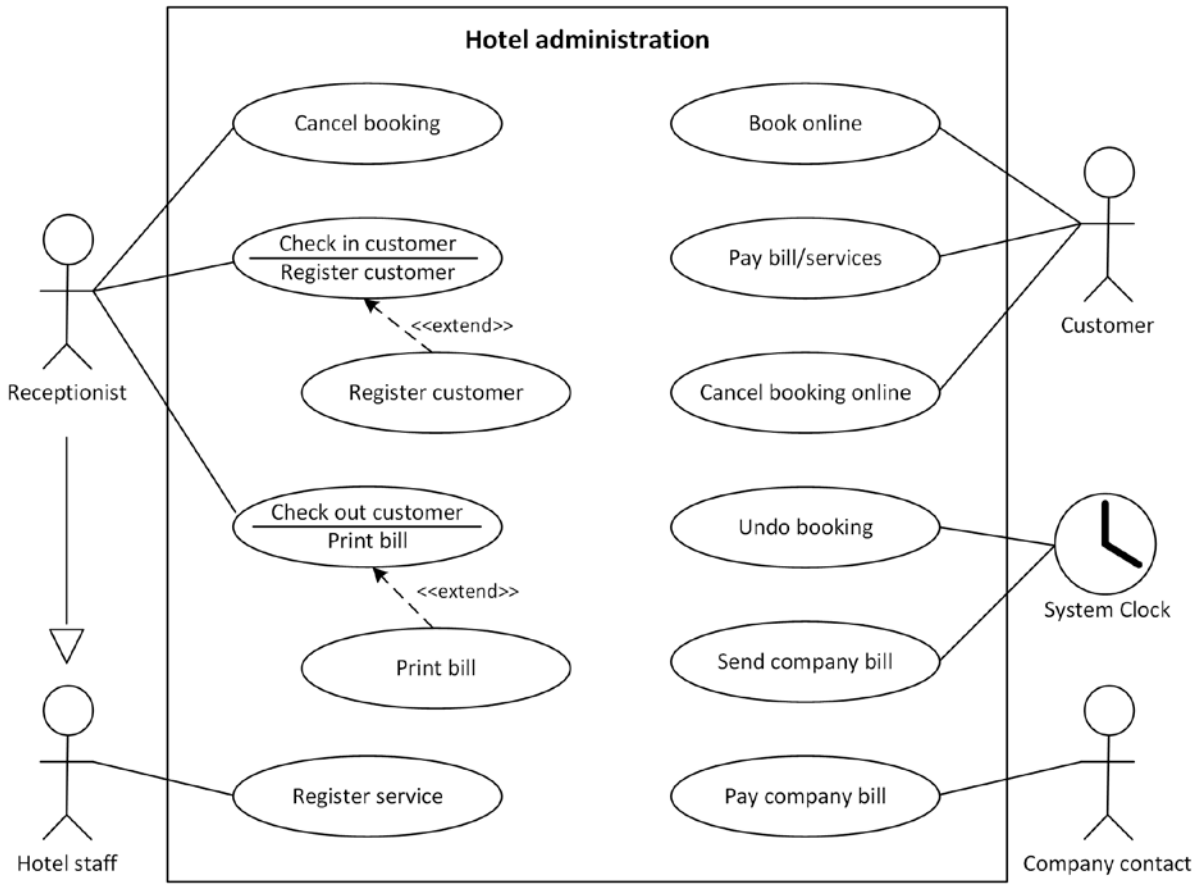
2a (Actor list) [0.4 points]

Actor	***Description***
Receptionist	Hotel staff in charge of checking in checking out customers
Hotel staff	(Relevant here): Can enter services provided to a customer
Customer	Hotel guest, can be private customer or company customer
Company contact	Responsible for paying the bills of company customers
System clock	Handles no-shows and monthly bills

Remarks:

- Private customers and company customers have exactly the same use cases, therefore it is easier to model them simply as “Customer”
- The actors in the actor list exactly match the actors in the use case diagram

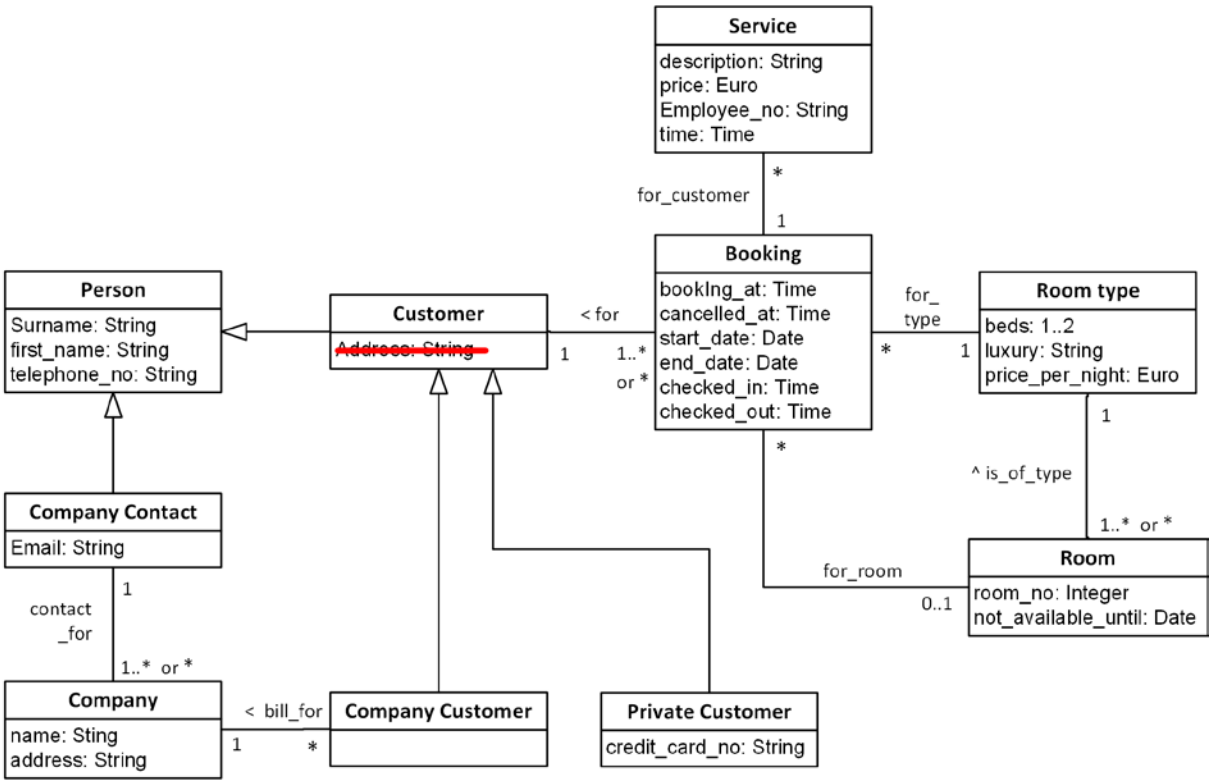
2b (Use case diagram) [1.3 points]



Remarks:

- Extensions are “Print bill” (does not apply to company customers without services) and “Register customer” (only if a customer has not made a booking)
- If you made no distinction between Receptionist and other Hotel staff, that is not a problem. Hotel staff has been graded as the correct actor for the Receptionist’s use cases.

Question 3 (Class Diagram) [3.2 points]



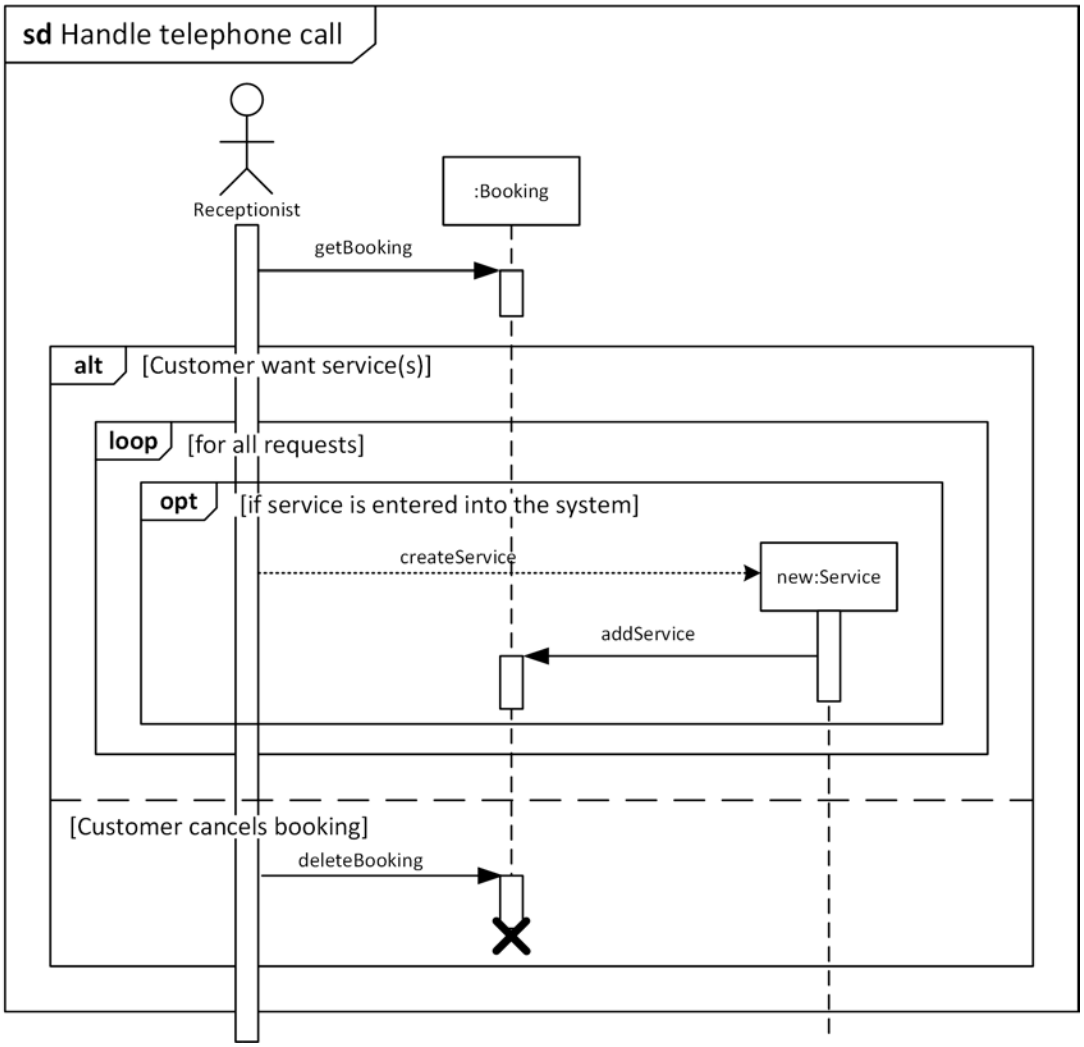
Remarks:

- The attribute “Address” in the class “Customer” (in the original version of the solution) was not mentioned in the text of the test, so it should not be part of the solution.
- Room type is not mentioned explicitly in the text but it should be clear from the description that each room is of a particular type. A booking is made for a room type. The specific room is allocated later, during check-in – therefore a booking is associated with 0 or 1 room.
- “Service” can be associated with “Booking” or with “Customer”, both are considered to be correct. An assocation with “Booking” is the most logical, because at the end of the stay the customer has to pay services related to *that* booking (the customer could stay multiple times in the hotel). If “Service” is associated with “Customer” then – assuming that bookings for one customer don’t overlap – the timestamp of the service will indicate with which booking it is to be associated.
- A booking is associated with one customer, therefore “Customer” is included as a superclass of Priave Customer and Company Customer. Otherwise you would need (identical) assocations between “Booking” and either customer class.

A note about the mutiplicities: In some cases it can be argued that a mutiplicity 1..* is more precise than a multiplicity *. It would be strange if there is a room type but not any room of that type. Similarly it can be argued that a customer is associated with at least one booking, or a company contact is associated with at least one company, otherwise they would not in the system. In all of these cases * and 1..* are graded as correct.

However, there could be a company without a company customer (when the company has made a contract with the hotel but none of its employees stayed there so far).

Question 4 (Sequence Diagram) [1.1 points]



Remarks:

- The receptionist may also pass on the request to someone else, who will take care of the request as well as the administration. Then the receptionist leaves no traces in the system for such a request. Therefore, acting on the request is **opt**ional in the sequence diagram.
  Alternatively (also correct is to have an **alt**, with one part for action by the receptionist, another part for passing on the service request
- Instead of having an alt for either cancellation or service requests, you could have two **opt**s.
- If a new service is created by the receptionist, it should be linked to the booking. (Also acceptable: linking it to the customer).
- getBooking would perhaps not be needed in case the phone call entails a single service request, and the receptionist is not going to handle the request but will pass it on. However, if you delete it at the top of the diagram, you have to enter it at two other places: inside the **opt** as well as in the second half of the **alt**.

Question 5 (Software Metrics) [1.2 points]

5a (Lack of Cohesion in Methods) [0.4 points]

The accesses of methods to attributes is shown in the following table:

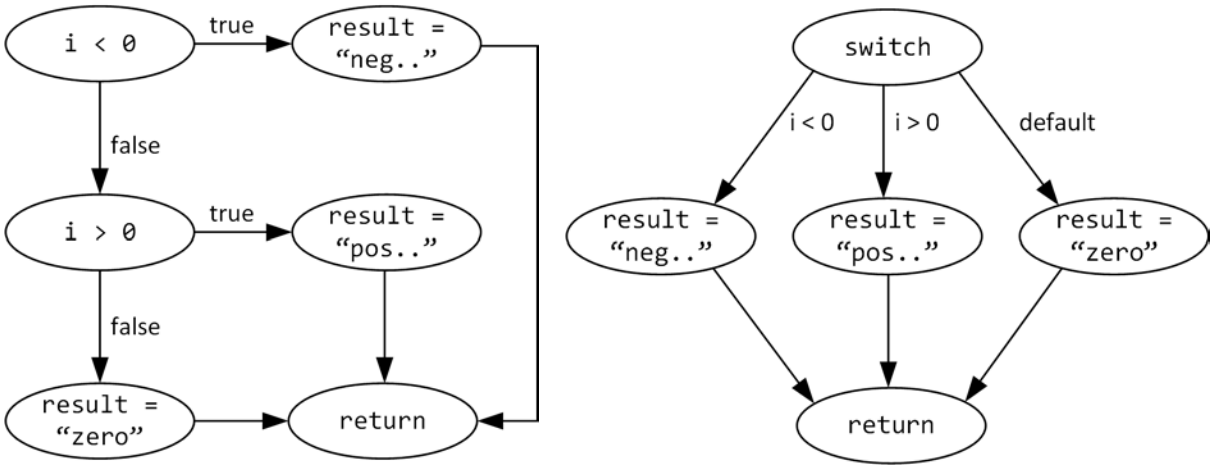
	h	v
Position	√	√
goUp		√
goDown		√
goLeft	√	
goRight	√	
getHorizontal	√	
getVertical		√

Both h and v are accessed by 4 methods, yielding avg(*mA*) = 4.
There are 7 methods, including the constructor, hence

LCOM2 = (7 – 4) / (7 – 1) = 3 / 6 = 0.5

5a (Cyclomatic Complexity) [0.8 points]

The flow graphs for sign1 and sign2:



For sign1 we find #nodes = 6, #edges = 7, CC = E – N + 2 = 7 – 6 + 2 = 3

For sign2 we find #nodes = 5, #edges = 6, CC = E – N + 2 = 6 – 5 + 2 = 3