

Name: Mujahid Khan

ID: 14582 -

Subject - software eng.

Section = (A) BS (SE)

Final term Assignment.

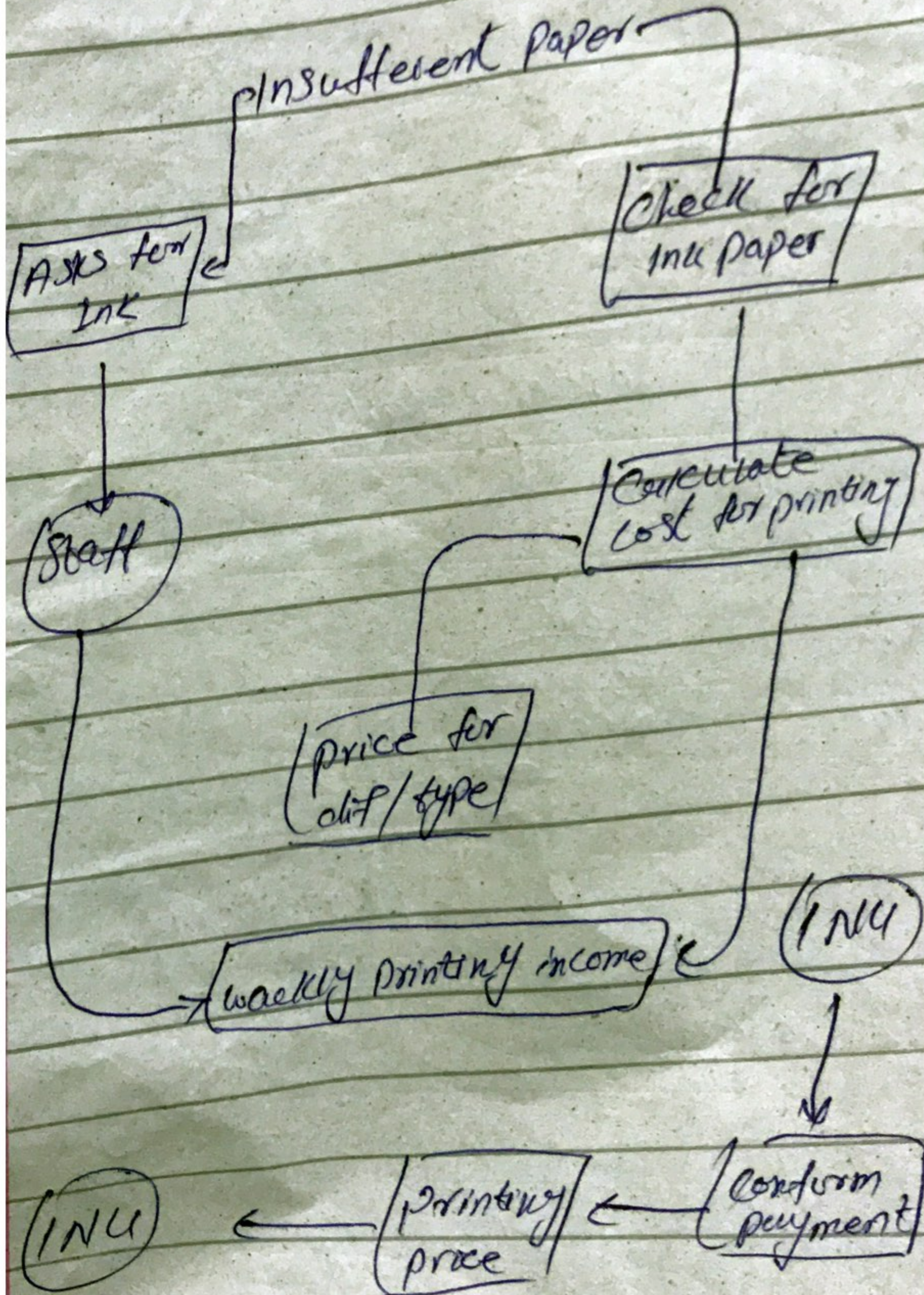
Submitted to: SIR Ghassan Sh.

Date: 22/06/2020



①

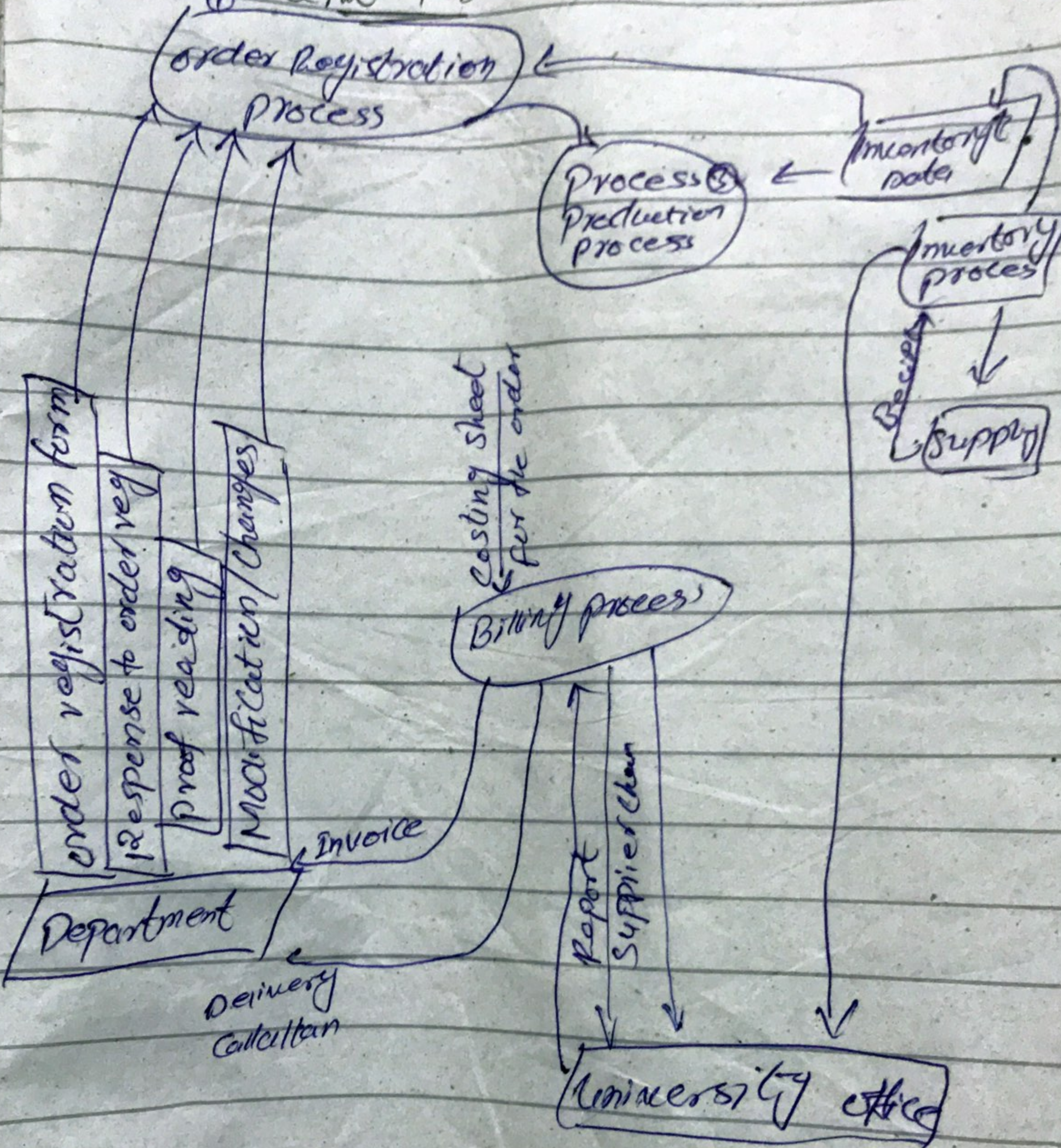
Q No 1-1 =





(2)

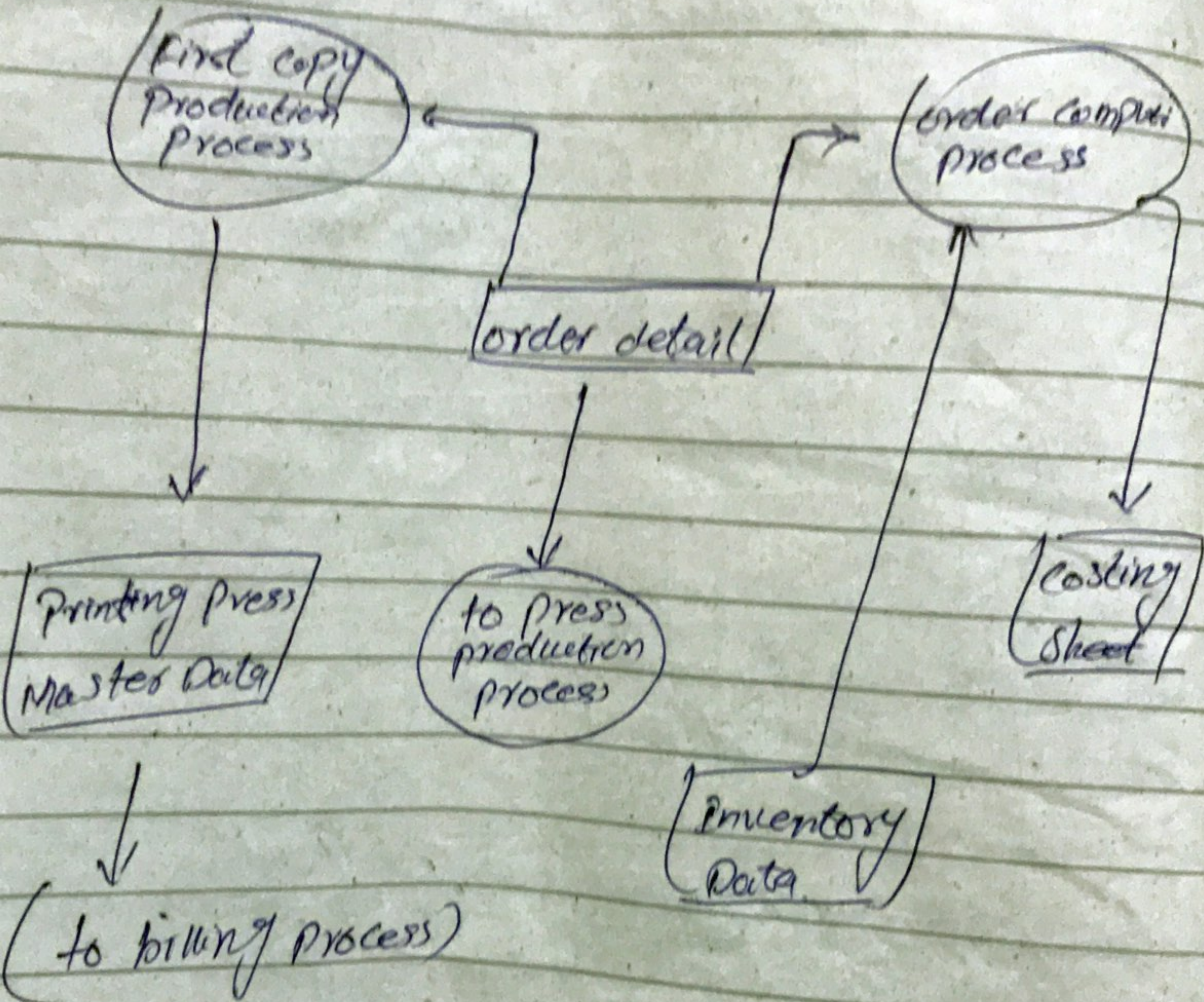
Q No 1-2





3

Q No 1.3





(4)

Question 2:

Q2: Q.2 → (1): Explain why testing can only detect the presence of errors, not their absence?

Ans 2

(1) Testing can detect only the presence of errors, not their absence because the main goal of testing is that -  
Testing is a part of broader process of software verification and validation. It consists of a set of activities where the testers try to make the software behave anomalous in order to detect or anomaly to be later fix. Testing cannot demonstrate the faults other than specified in every circumstance. It is always possible that a



(5)

have overlooked could discover  
Another problem with the system -

Q No 2:

Part (2)

Define the following terms -

- ① Unit Testing -
- ② System testing -
- ③ Black box testing -
- ④ White Box Testing -

Ans (2) Unit Testing -

Unit Testing  
is a level of software testing  
where individual units / compo-  
nents of a software are  
tested. The purpose is to  
validate that each unit of  
the software performs as  
designed. A unit is the  
smallest testable part of  
any software. It usually  
has one or a few input and  
usually a single output -



(16)

## System testing

System testing is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In system testing the functionalities of the system are tested from an end-to-end perspective.

## Black box testing

Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing. Unit integration, system and acceptance.



(7)

White box testing =

White box testing is testing of a software solution's internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security -

Question 3:

Q.3 (4) Briefly describe the three main types of software maintenance. Why is it sometimes difficult to distinguish b/w them?

Ans 3

① The three main types of software maintenance -



(8)

① Fault repairs =

Coding errors are usually relatively cheap to correct design errors are more expensive as they may involve rewriting several program components. Requirements errors are the most expensive system redesign which be necessary.

② Environmental adaptation:

This type of maintenance is required when some aspect of the system environment such as the hardware the platform operating system, or other support software changes the application system must be modified to adapt it to cope with these environmental changes.



(9)

③ Functionality addition -

This type of maintenance is necessary when the system requirements change in response to organizational or business change. The scale of the changes required to the software is often much greater than for the other types of maintenance -

Differentiate b/w the types of maintenance -

In Practice, there is not a clear-cut distinction b/w these type of maintenance. When the system adapt to new environment then add functionality to take advantage of new environmental features. Software faults are often exposed because users use



(10)

the system in ~~un~~ maintenance  
~~and~~ unanticipated ways.

These types of maintenance  
are recognized but a diff-  
erent person sometimes  
gives them different names.

"Corrective maintenance" is  
universally used to refer to  
maintenance for fault repair.

"Adaptive maintenance" some-  
times means adapting to new  
environment and sometimes  
means adapting the software  
to new requirements.

"Perfective maintenance" Some-  
times means perfecting the  
software by implementing new  
requirements in other cases  
it means maintaining the  
functionality of the system but  
improving its structure and  
Performance.



(11)

Part (2) What are the principal factors that affect the costs of system reengineering? Also briefly explain the reengineering process with the help of diagram?

Ans (2) The quality of the software to be reengineering. The tool support availability for engineering. Extent of the data conversion which is required. The availability of expert staff for reengineering. The lower the quality of the software and its associated documentation.



# Reverse Engineering Process

(1a)

