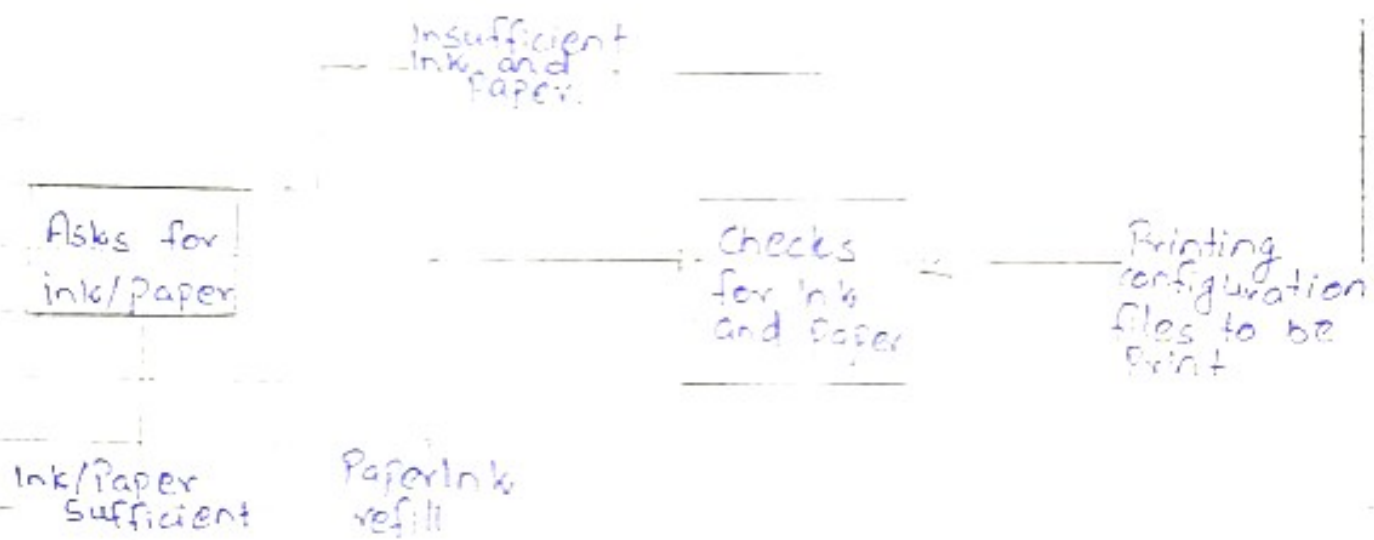


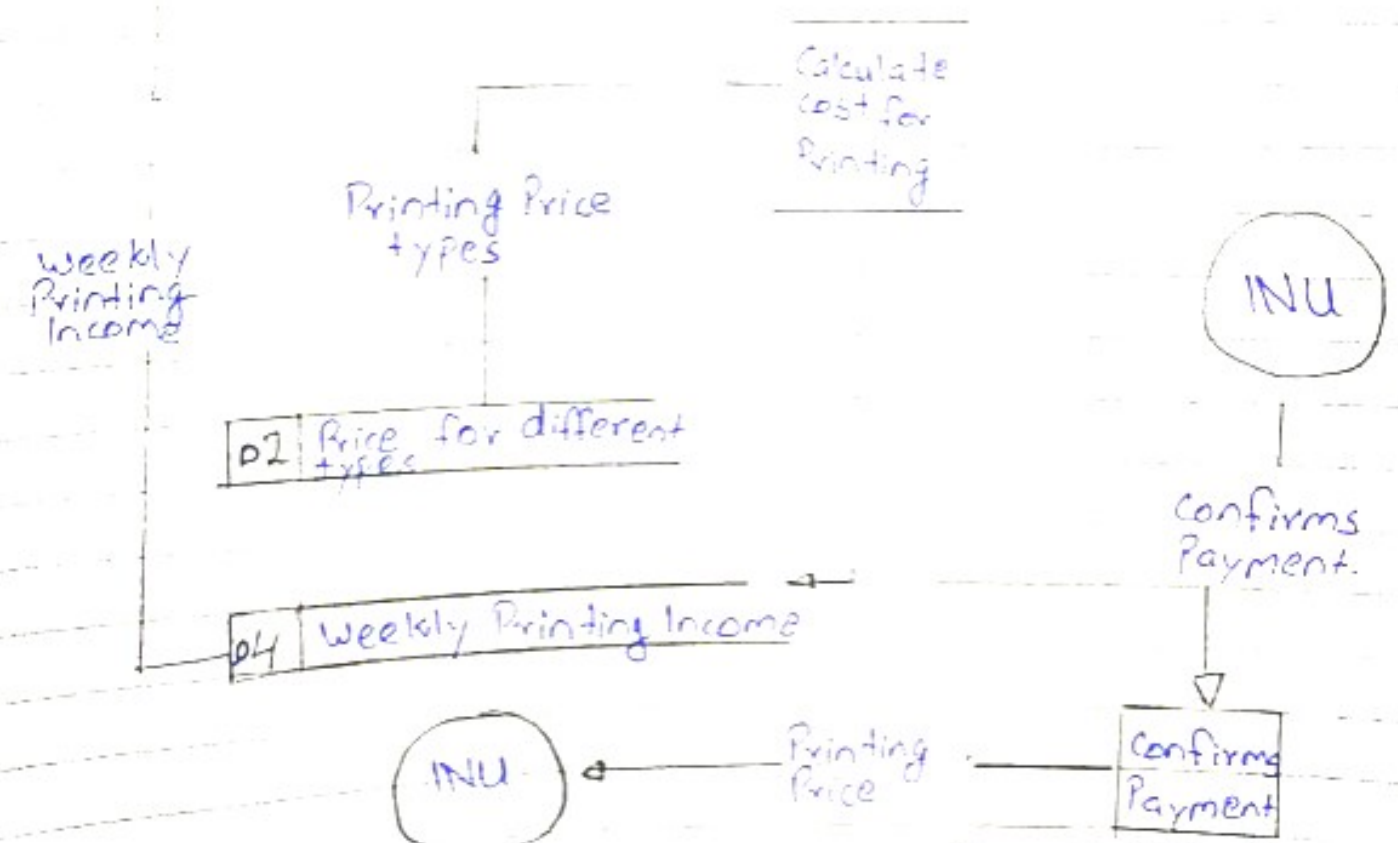
# Question no 1 (i):-



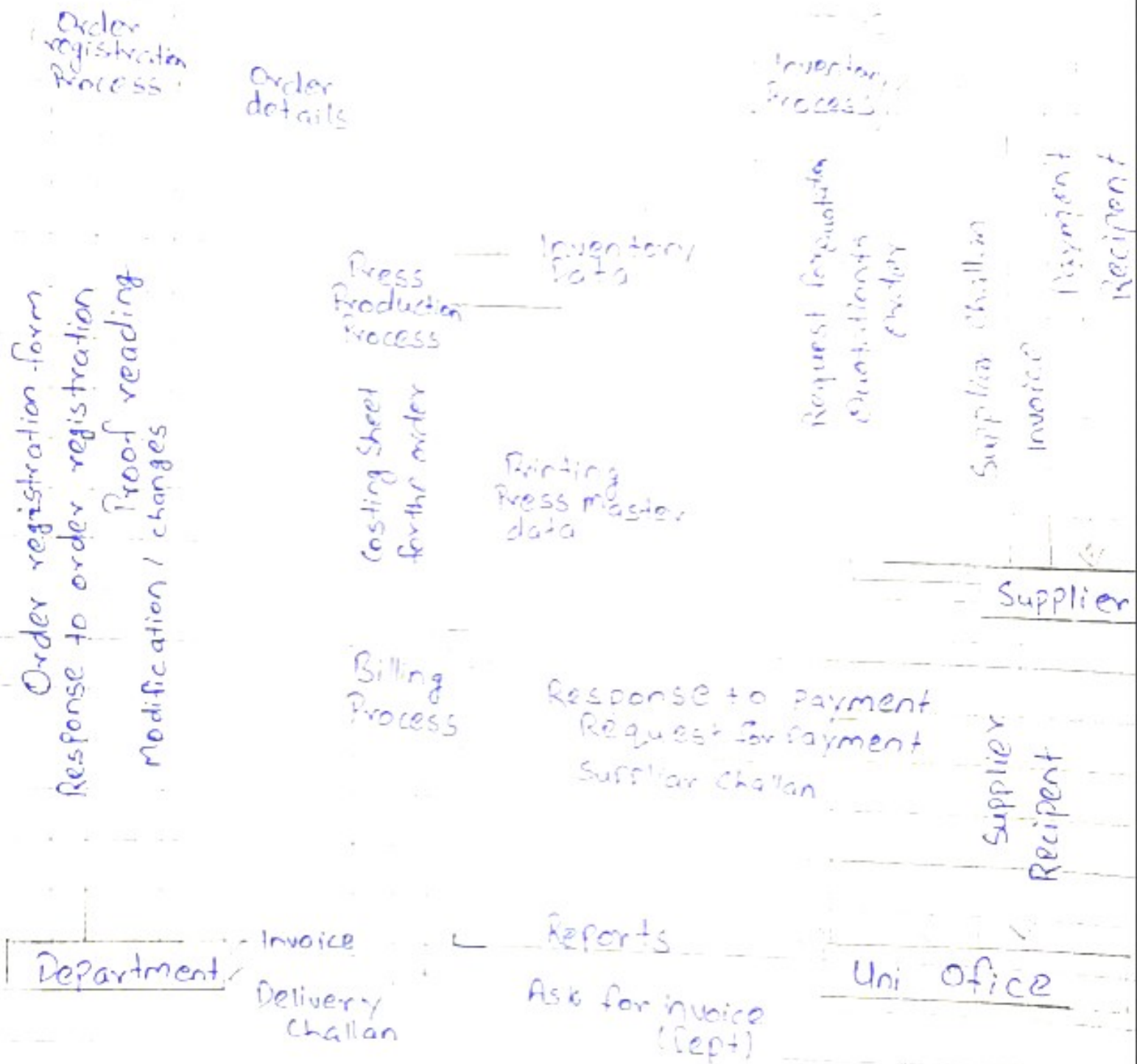
Ink/Paper Sufficient      Paper/Ink refill



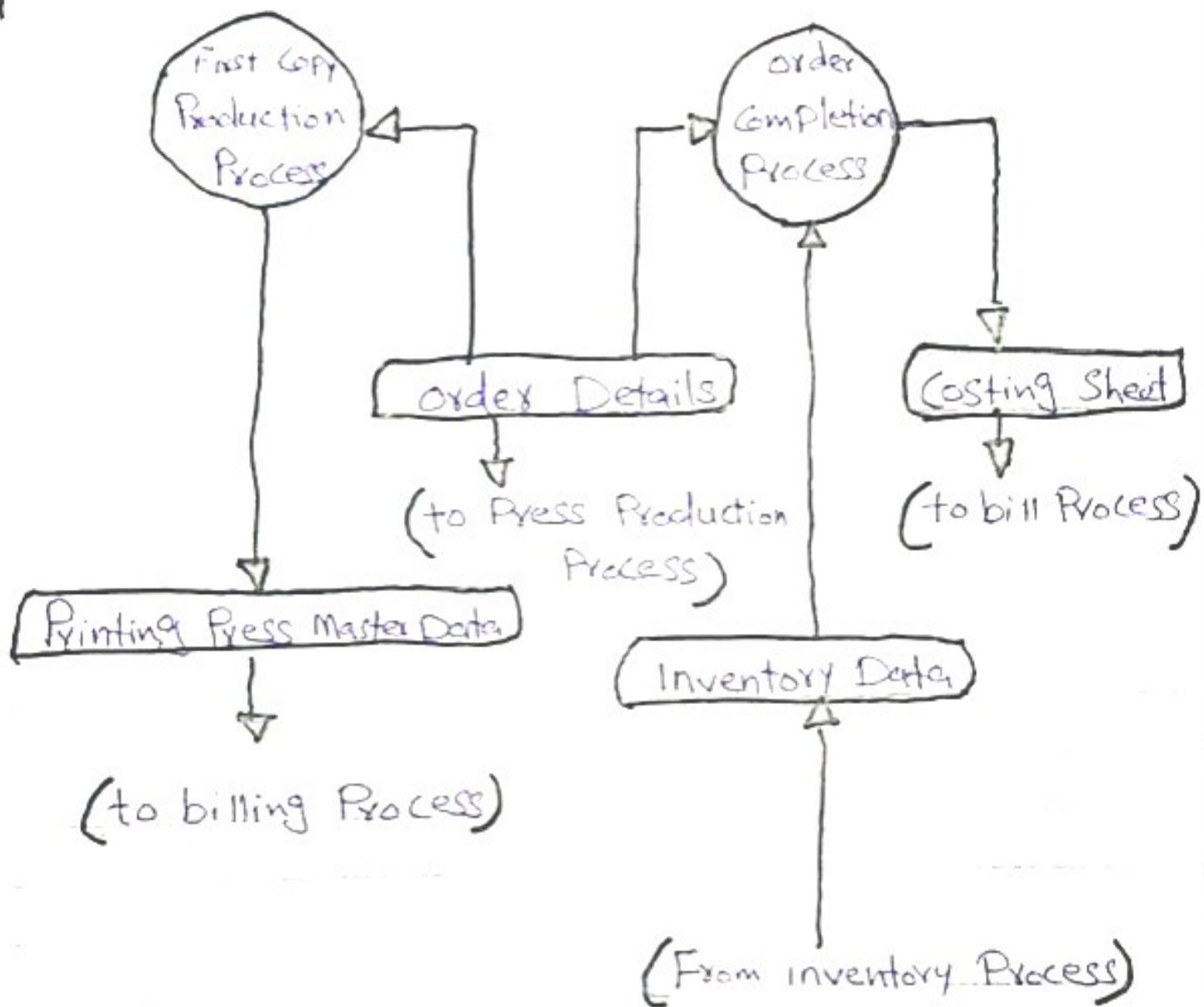
Printing Configuration files to be print confirms Sufficient ink and Paper



# Question 1 (ii)



# Question 1 (iii):-



## Question 2(i):-

Answer:-

Testing can detect only the presence of errors, not their absence because the main goal of the testing is to observe the behavior of the particular software and to check whether it meets its requirements expectation or not.

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 can not demonstrate the faults other than specified in every circumstance it is always possible that a test have overlooked could discover further problem with the system.

## Question 2(ii):-

### i) Unit Testing:-

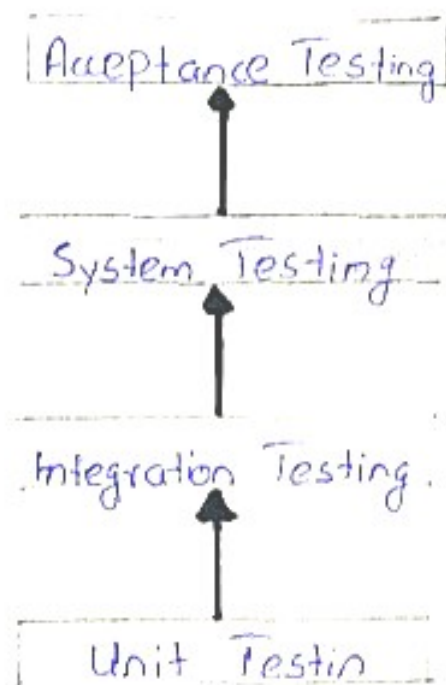
Unit testing is defined as a type of software testing where individual components of a software are tested. Unit testing of software product is carried out during the development of an application. An individual component may be either individual function or a procedure.

### ii) System Testing:-

System Testing is a level of software



Testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.



### 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.

### White Box Testing:-

White box testing (also known as clear box testing, glass box testing, transparent box testing, code based testing or structural testing) is a software testing method in which the internal structure / design / implementation of the item being tested is known to the tester.

## Question n.o 3(i):-

### i) 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 to repair because of the expensive system redesign which be necessary.

### ii) Environmental Adaptation: →

This type of maintenance is required when some aspects of the system's 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 the environmental changes.

### iii) 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.

Why is it difficult to distinguish b/w the types of maintenance:-

In practice, there is not a clear cut distinction b/w these types of maintenance, when the system adapt to new environment, then add



functionality to take advantage of new environmental changes. Software faults are often exposed because users use the system in unanticipated ways. These types of maintenance are recognised but a different person sometimes gives them different names.

"Corrective maintenance" is universally used to refer to maintenance for "Fault Repairs".

"Adaptive maintenance" sometime means adapting to new environment and sometimes means adapting the software to new requirements.

"Perfective maintenance" sometime 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.

### Question 3 (ii):-

Answer:-

#### System Re-Engineering:-

Software re-engineering is the examination of a system to reconstitute it in a new form. The principles of Re-Engineering when applied to the software development process is called software re-engineering or system re-engineering. It affects positively at software cost, quality, service to the customer and speed of delivery. In System Re-Engineering we are improving the software to make it more efficient and effective.

#### Re Engineering Cost factors:-

- i) The quality of the software to be re-engineered.
- ii) The tool support availability for engineering.
- iii) Extent of the data conversion which is required.
- iv) The availability of expert staff for Re-Engineering.