Q1) Major software functions identified

1. **User interface**
2. **Database management**
3. **Report generation**

For user interface :

Sopt: 2800
Sm : 3000
Spess : 5000

For dababase managemant

Sopt : 5600
Sm : 7900
Spess : 7600

For report generation
Sopt : 1200
Sm : 1600
Spess : 3200

Suppose,
102 productivity = 2000 LOC / person-month
103 Development Cost = $9000 person-month

** Now calculate the Person-month, Cost of software, and Cost.

Q2)

Information Domain Values	Simple	Average	Complex	Count
Number of inputs	**5**	6	6	**11**
Number of outputs	4	**4**	8	**9**
Number of inquiries	7	8	**7**	**8**
Number of files	**3**	4	6	**13**
Number of external	6	**5**	3	**15**

interfaces				

Total factors = 49.

For historical data, productivity is 65.5 FP person-month, and development cost is $9000 per month.

****Now, calculate the Productivity, Total Cost, and Cost per FP.**

Q3)

Project Class	a_b	b_b	C_b	d_b
Organic	5.12	6.25	6.33	11.38
Semidetached	4.17	4.63	8.45	9.23
Embedded	7.45	8.23	7.23	8.33

Assume that the expected size of a **Semi-detached** software project is 35.4 KLOC. Using the coefficients in the above table and the **BASIC COCOMO MODEL**, calculate the recommended number of people needs to be engaged to complete the project.

Q4)

Project Class	a_i	b_i	d_i
Organic	5.12	6.25	11.38
Semidetached	4.17	4.63	9.23
Embedded	7.45	8.23	8.33

Cost Drivers	Level	Effort Adjustment Factor

Complexity	**High**	**1.25**
Storage	**Low**	**2.25**
Experience	**Low**	**1.68**
Programming Capabilities	**High**	**2.56**
Analyst capability	**High**	**1.29**
Memory Constraints	**Low**	**1.78**

** Calculate the effort applied in person-month using the Intermediate COCOMO Model. What will be the total development time, and how many people should be hired?

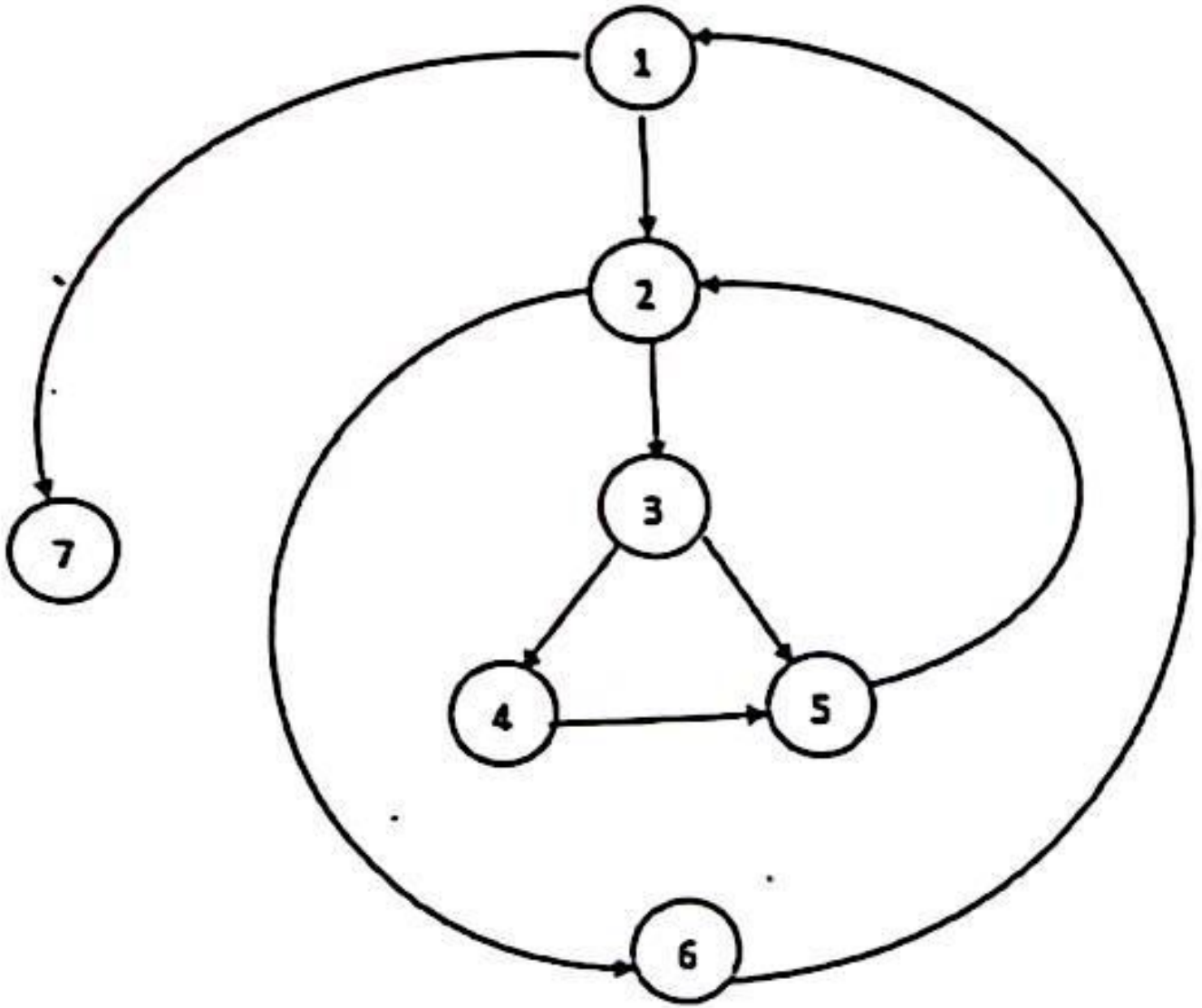
Q5

Consider the following project schedule with activity labels, durations (in weeks), and their respective precedents:

Activity	Duration (weeks)	Precedents
A	7	-
B	5	-
C	3	A
D	6	B
E	4	C, D
F	9	E
G	4	E
H	3	F, G
I	1	H

Convert the given table into a Network Diagram and find the critical path showing the earliest start time (ES), earliest finish time (EF), latest start time (LS), latest finish time (LF).

Estimate the Cyclomatic complexity of the following flow chart. Verify your answer using Graph matrix theory.



Q6.

Ultimate suggestions for Introduction to Software Engineering:

1. Use Case Diagram and Description

- Identify actors & use cases
- Show relations
- Describe use case

(See the lecture slide for more practise example)

2. Risk Analysis (Theory)

- Classify

  Requirement, Cost, Scheduling, Quality
- Scenario based questions

(See the lecture slide)

3. Software Scheduling & Management

- Planning, lifetime and principle (Theory)
- CPM
- Paths
- ES/EF
- LS/LF

(See the lecture slide for theory)

4. Cost Estimation

- COCOMO (Basic, Intermediate)
- LoC

- Effort
- EV (Function Point)

(Memorise the theories well!)

5. Risk Analysis

- Classify risks,
- Identify from scenario

6. Software Testing

- Cyclomatic Complexity (math)
- Types (Unit, Integration, System, Acceptance)
- Black-box vs White-box Testing
- Functional vs Non-functional

7. Project management (Theory)

(See the lecture slide)