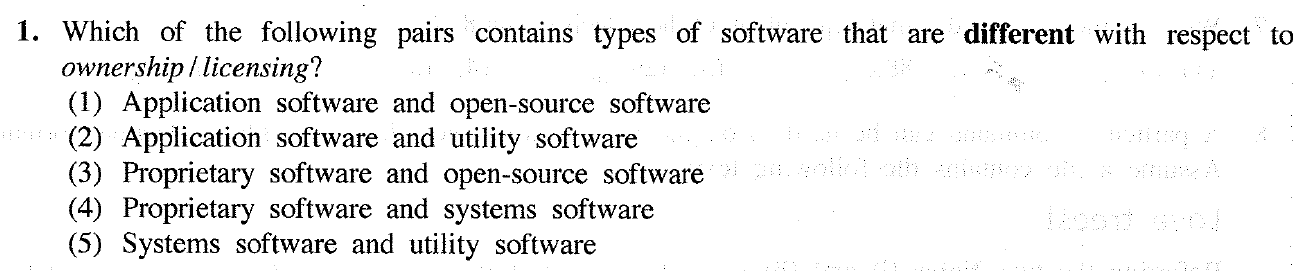
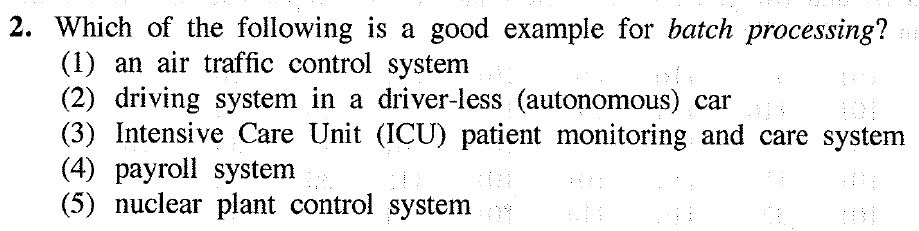
**Advanced Level – Information and Communication Technology**

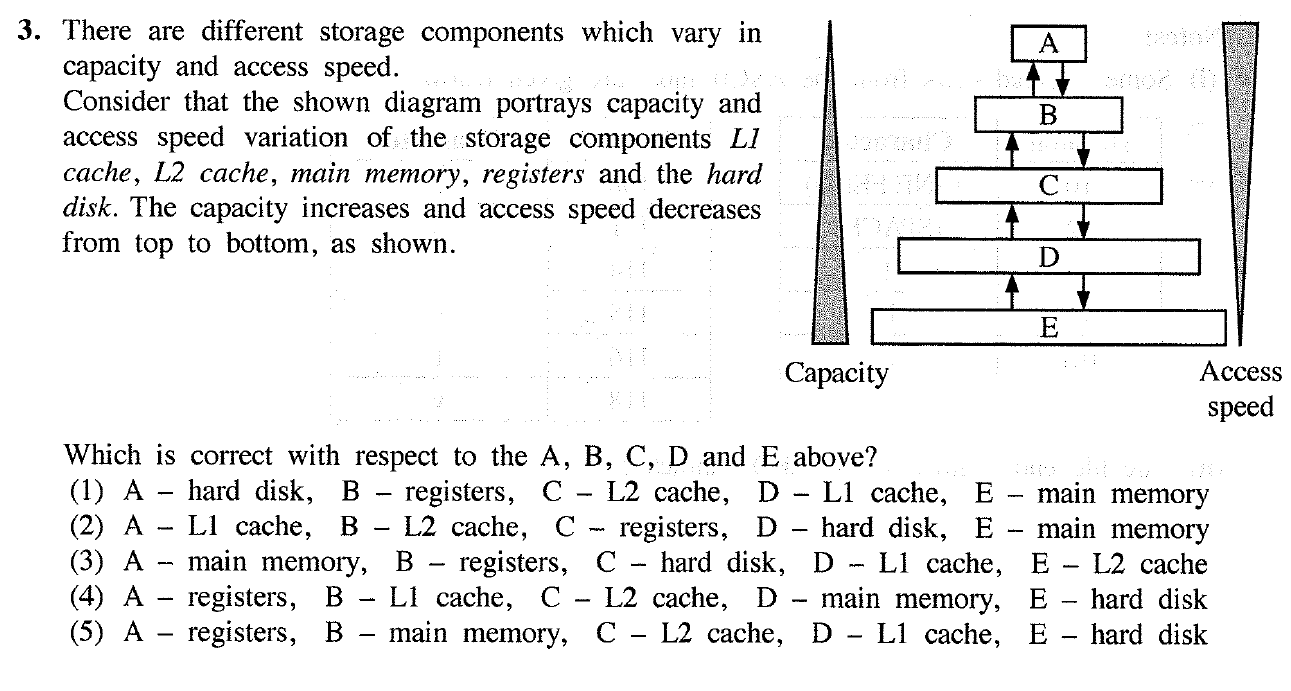
**2021 MCQ Past Paper**

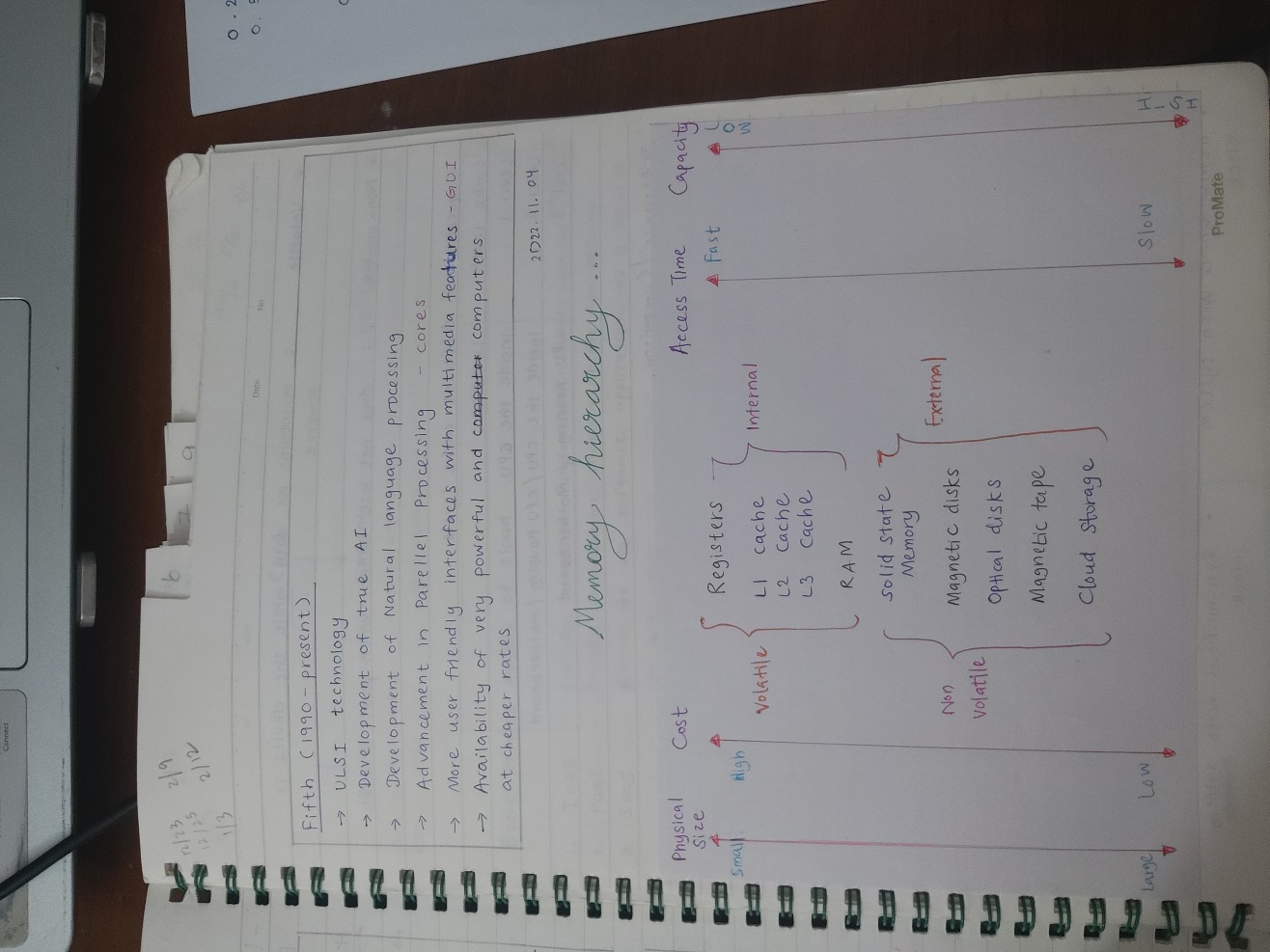


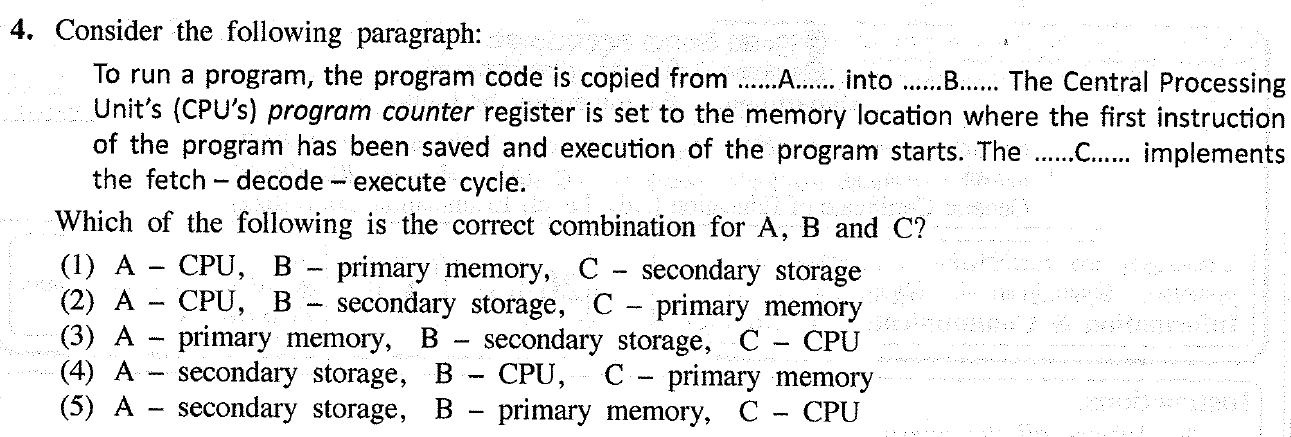
|  |  |
| --- | --- |
| Application Software | A type of computer program that performs a specific personal, educational, and business function. |
| Utility Software | Includes all systems and programs on a computer system that maintain its functionality |
| Open-source software | A software that is distributed with its source code, making it available for use, modification, and distribution with its original rights |
| Proprietary Software | Owned by an organization or an individual |
| System Software | A program designed to run a computer’s hardware and applications and manage its resource, such as its memory, processors and devices |

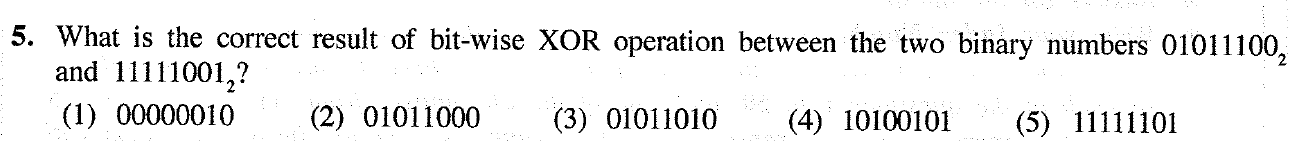


|  |  |
| --- | --- |
| Batch processing | The method computers use to periodically complete high-volume, repetitive data jobs |





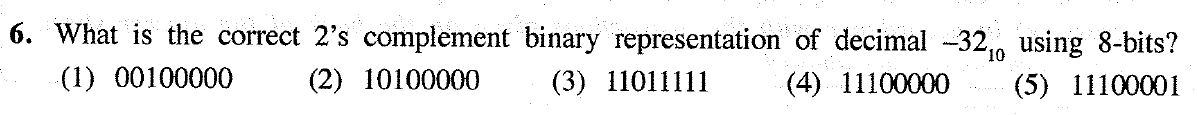




01011100

XOR 11111001

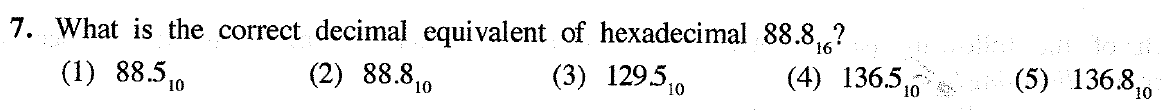
10100101



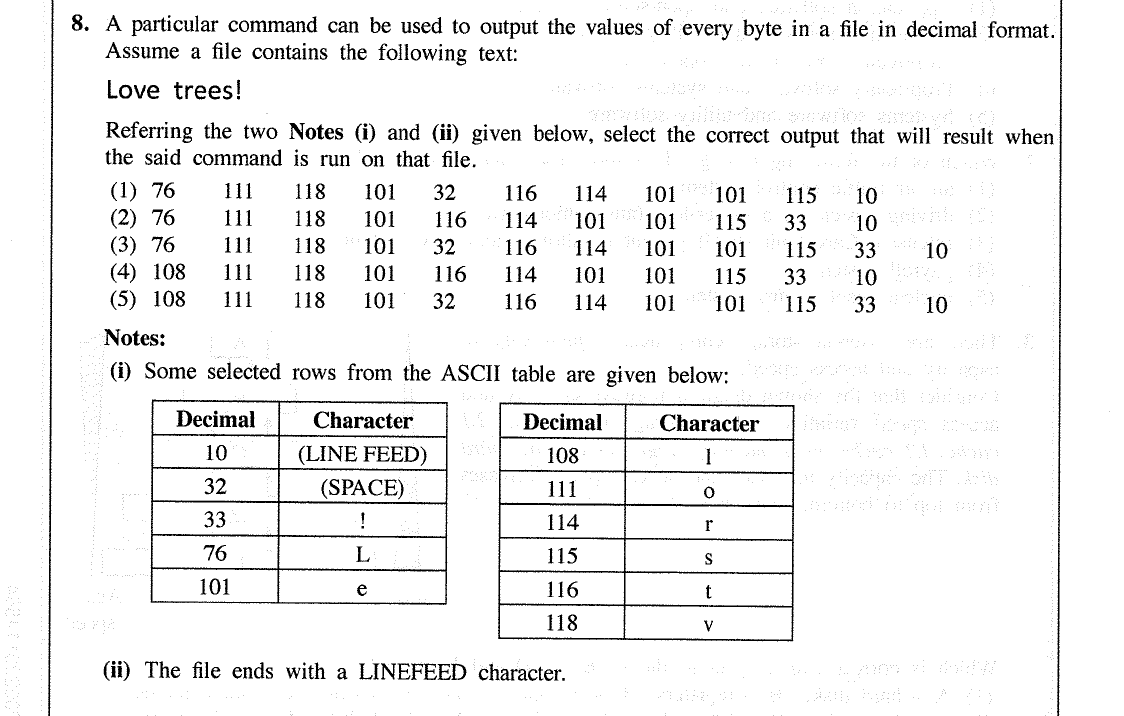
Step 1: Convert positive 32 to binary 🡪 0010 0000

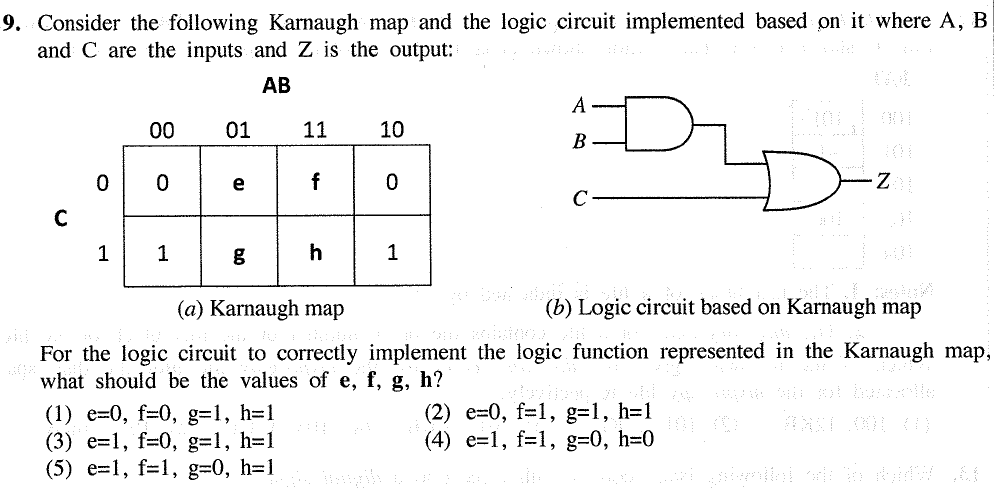
Step 2: Invert all bits and add 1 to the least significant bit

0010 0000 🡪 1101 1111 + 1 🡪 1110 0000

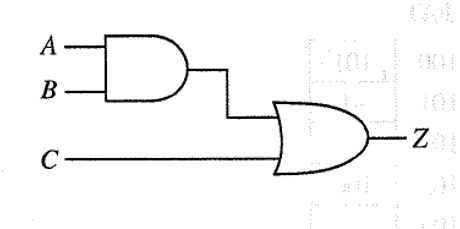


|  |  |  |  |
| --- | --- | --- | --- |
| Power in Hexa | 161 | 160 | 16-1 |
| Equivalent | 16 | 1 | 0.0625 |
| Hexa Number | 8 | 8 | 8 |
| Decimal value | 16\*8  128 | 1\*8  8 | 0.0625\*8  0.5 |
| 128+8  136 | | 0.5 |





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **AB** | **AB+C** | **SOP** |
| 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 1 | 0 | 1 | A’B’C |
| 0 | 1 | 0 | 0 | 0 |  |
| 0 | 1 | 1 | 0 | 1 | A’BC |
| 1 | 0 | 0 | 0 | 0 |  |
| 1 | 0 | 1 | 0 | 1 | AB’C |
| 1 | 1 | 0 | 1 | 1 | ABC’ |
| 1 | 1 | 1 | 1 | 1 | ABC |



AB+C

AB

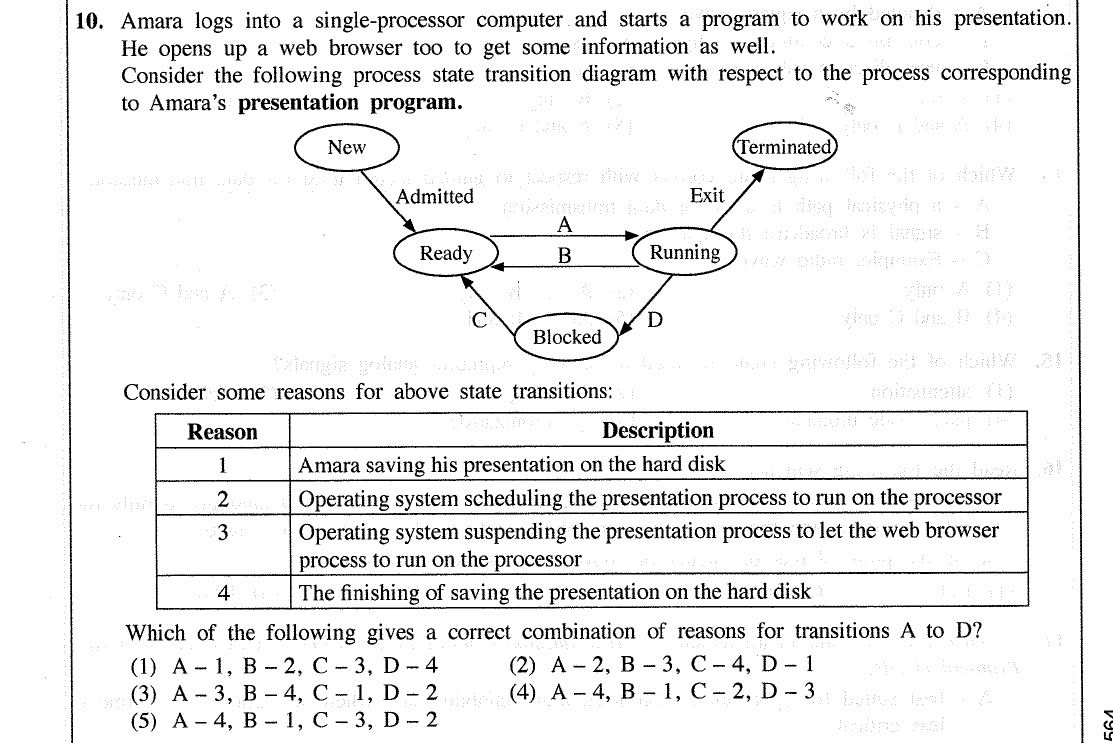
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C  AB | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

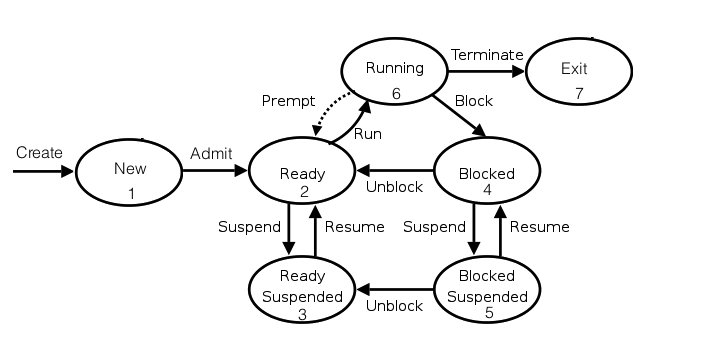
e=0

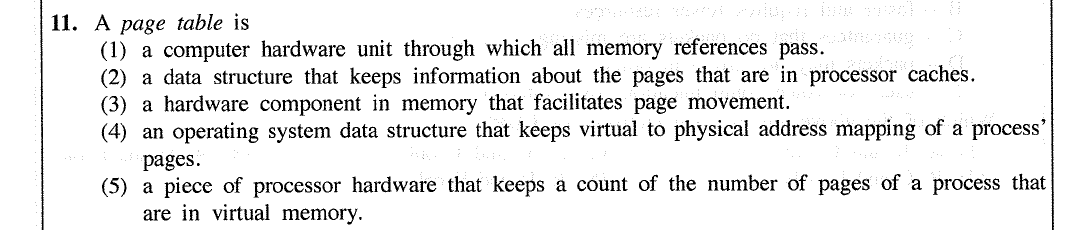
f=1

g=1

h=1

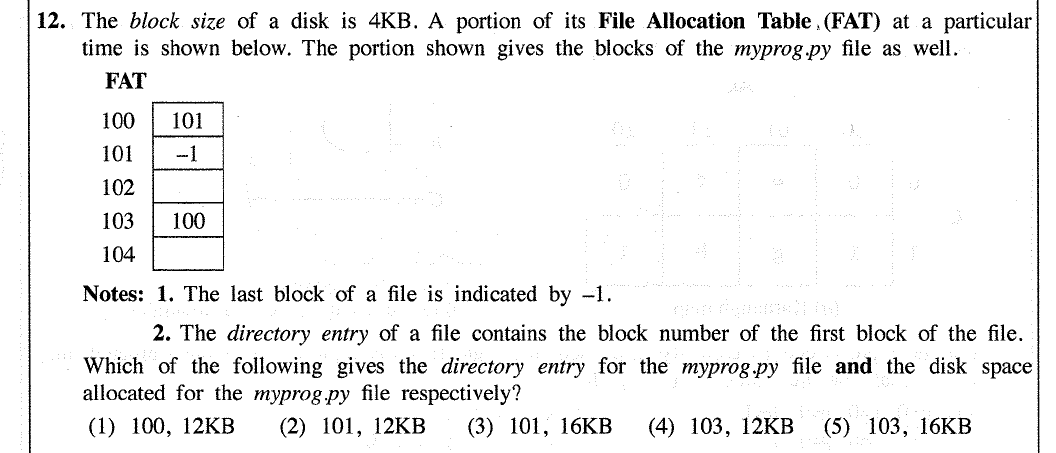






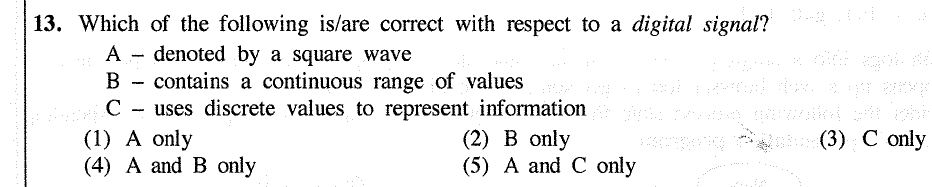
**Paging**

* Divide logical address space (virtual) and physical address space (RAM) into same size smaller blocks
* Is a memory management scheme that eliminates the need for contiguous allocation of physical memory
* Page table is stored in RAM

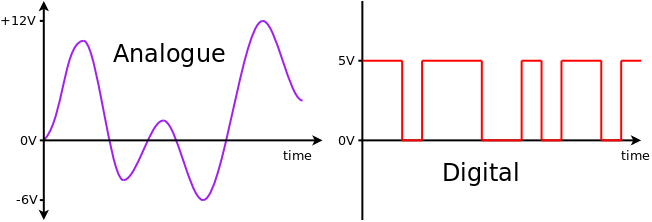


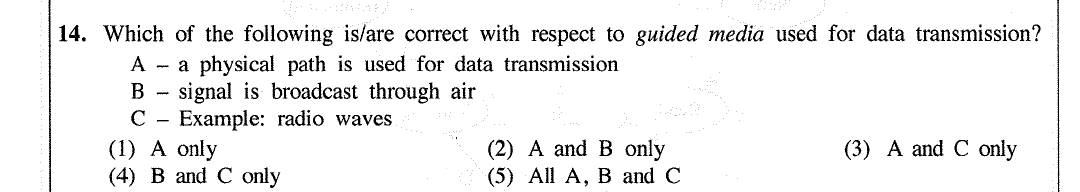
* The directory entry is 103 because the lowest value is in that block.
* As each block is 4KB and there are 3 blocks including the last block, the disk space

allocated for the file will be (4\*3) 12KB.

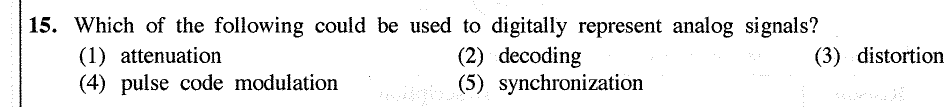


Digital signal is noncontinuous, discrete time signal

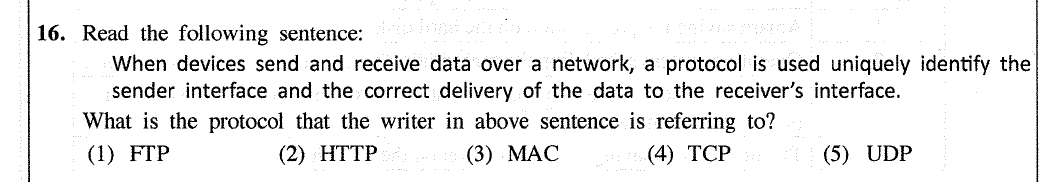




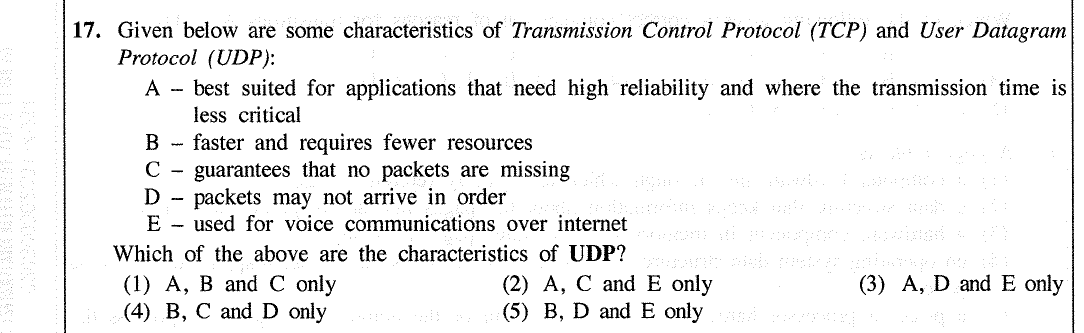
|  |  |
| --- | --- |
| **Guided Media** | **Unguided Media** |
| A medium that sends signals through a solid physical path | A medium that sends signals through free space |
| Comparatively more reliable | Comparatively less reliable |
| Comparatively faster | Comparatively slower |
| * Twisted pair * Coaxial cables * Fiber-optic cable | * Radio waves * Microwave * Satellite |



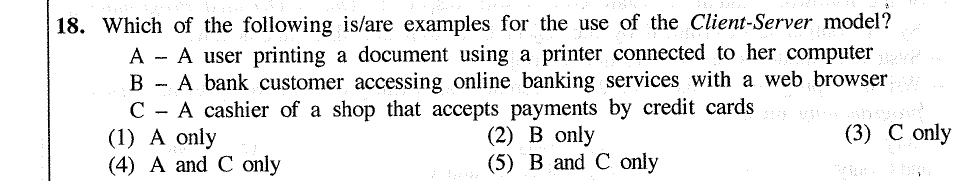
|  |  |
| --- | --- |
| (1) Attenuation | Signal strength (height of the signal) reduces when it travels through the media |
| (2) Decoding | The process by which data transmitted over computer networks is coverted from binary code into its original, often text-based format |
| (3) Distortion | Generally unwanted change in signal |
| (4) Pulse code modulation | A method of digitally representing analog signals by discretizing the amplitude of the signal at regular intervals and quantizing the values to obtain a series of coded pulses. |
| (5) Synchronization | The coordination of events or timing between different components or processes to ensure they operate in a harmonized and orderly manner. |

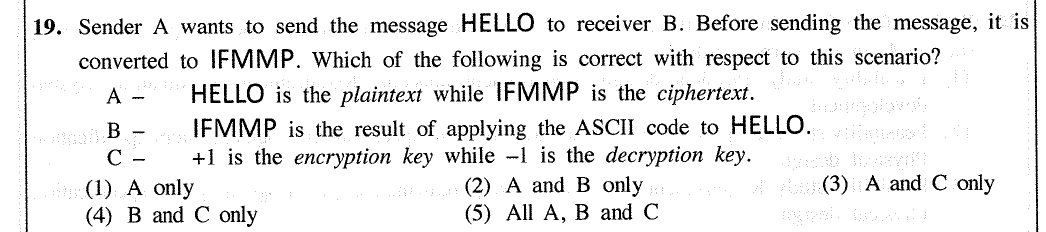


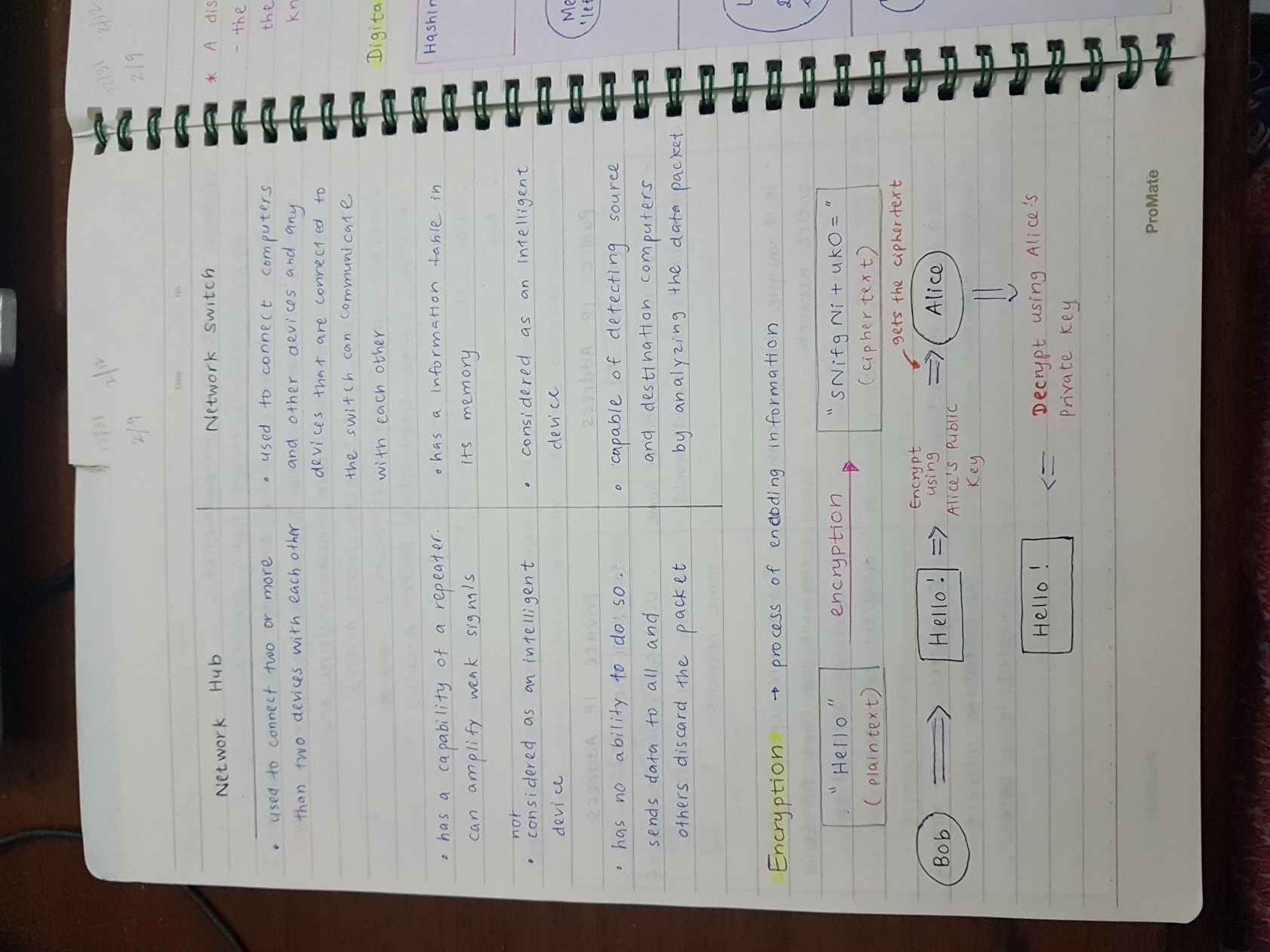
|  |  |
| --- | --- |
| (1) FTP | A way to download, upload and transfer files from one location to another on the Internet and between computer systems |
| (2) HTTP | An application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack |
| (3) MAC | Is a unique identifier assigned to a network interface controller (NIC) for use as a network address in communications within a network segment |
| (4) TCP | A communications standard that enables application programs and computing devices to exchange messages over a network |
| (5) UDP | A communication protocol for time-sensitive applications like gaming, playing videos or Domain Name System (DNS) lookups |

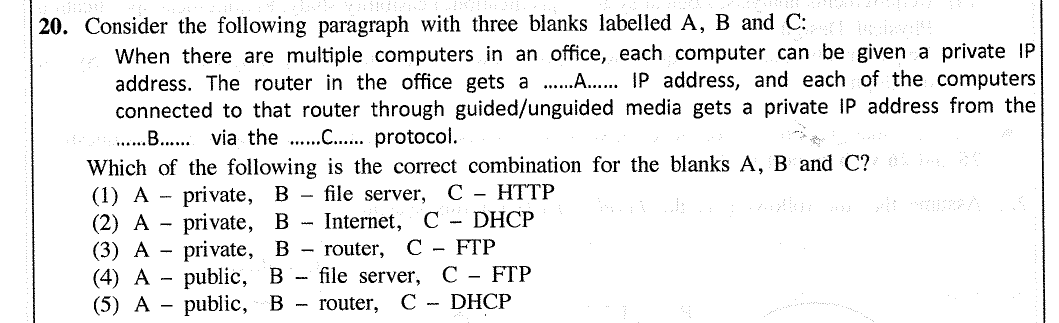


|  |  |
| --- | --- |
| **Transport layer protocols** | |
| **TCP** | **UDP** |
| 3 way handshake  1. SYN 2. SYN-ACK 3.ACK | Request 🡪 Response |
| Connection-oriented  (Establish🡪Maintain🡪Close) | Connectionless |
| Guarantee delivery of data packets if possible | Do not guarantee |
| Retransmission of lost data packets is possible | Not possible |
| Reliable | Unreliable |
| Data packets arrive in order | Application layer manage order if needed to be ordered |
| Comparatively slower | Much faster, simpler, efficient |
| Checks the readiness of the receiver | Do not check |
| Extensive error checking (parity) | Basic error checking mechanisms |
| Does not support broadcasting | Does support broadcasting |
| E.g.  Email, Web pages (HTTP); FTP | E.g.  Live video streaming  DNS, DHCP |



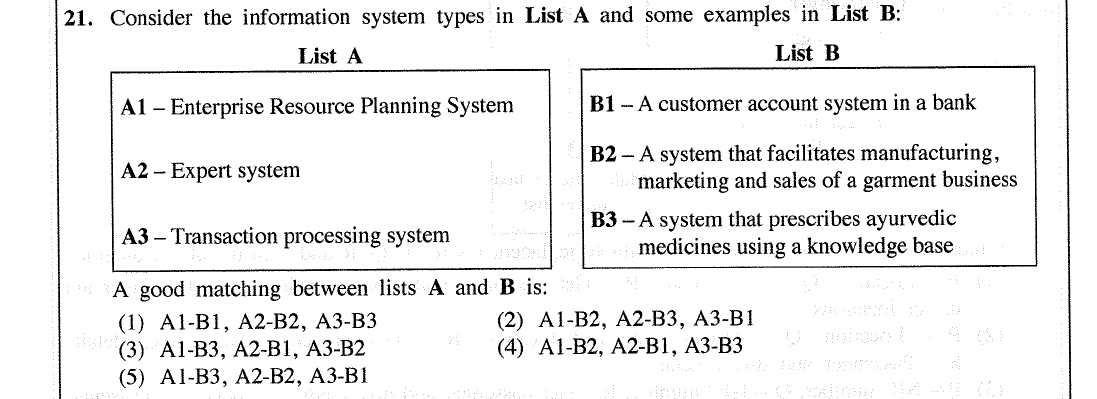




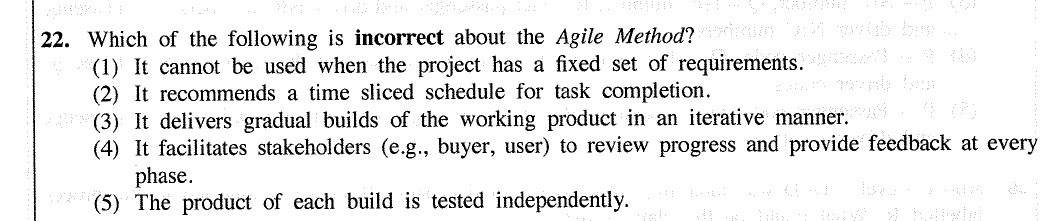


In a network, the router which connects to the internet obtains a public IP address. Each device which connects to the router gets private IP addresses. Each address is unique in a particular network.

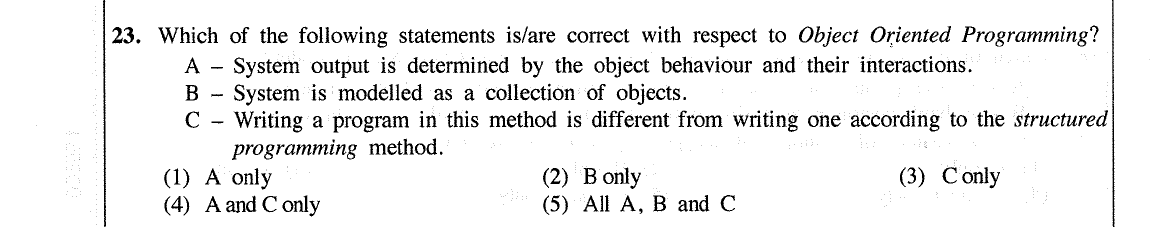
|  |  |
| --- | --- |
| HTTP protocol | An application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack |
| DHCP protocol | A client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway |
| FTP protocol | A way to download, upload and transfer files from one location to another on the Internet and between computer systems |



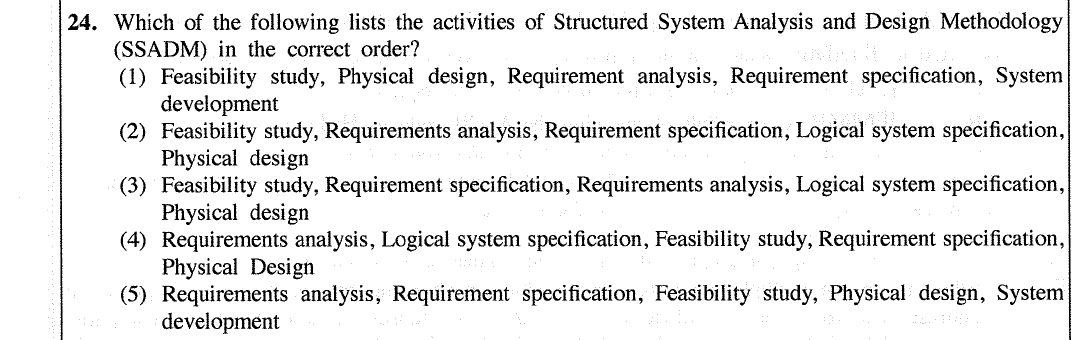
|  |  |
| --- | --- |
| Enterprise Resource Planning System | A software system that helps to run the entire business, supporting automation and processes in finance, human resources, manufacturing, supply chain, services, procurement and more |
| Expert System | A computer program that uses artificial intelligence (AI) technologies to simulate the judgement and behavior of a human or an organization that has expertise and experience in a particular field |
| Transaction processing system | An information processing system for business transactions involving the collection, modification and retrieval of all transaction data |



|  |
| --- |
| Waterfall model   * First process model to be introduced which is simple to understand and use * Requirements have to be well-known, clear and fixed * Project is short |
| Spiral model   * Has four phases 🡪 Planning, Risk analysis, Engineering and Evaluation * Complex and unsure, unfixed user requirements (Change according to the user time to time) * Long term projects |
| RAD model (Rapid Application Development)   * For systems which are needed in in a short span of time * User will be involved all through the life cycle * Requirements are known but might change |
| Agile model   * Prioritizes features, continuously gathers customer feedback and adjusts and remains flexible throughout the process |



|  |
| --- |
| **Structured software development methods** are process-oriented, focusing primarily on modeling the processes, or actions that capture, store, manipulate, and distribute data as the data flow through a system. These methods *separate* data from processes. |
| **Object oriented software development** is a programming paradigm that uses objects, which are instances of classes, for organizing code. |



**Software Development Life Cycle**

1. System Investigation

- Preliminary Analysis

- Feasibility Study

2. System Analysis

- Functional requirements

- Non-functional requirements

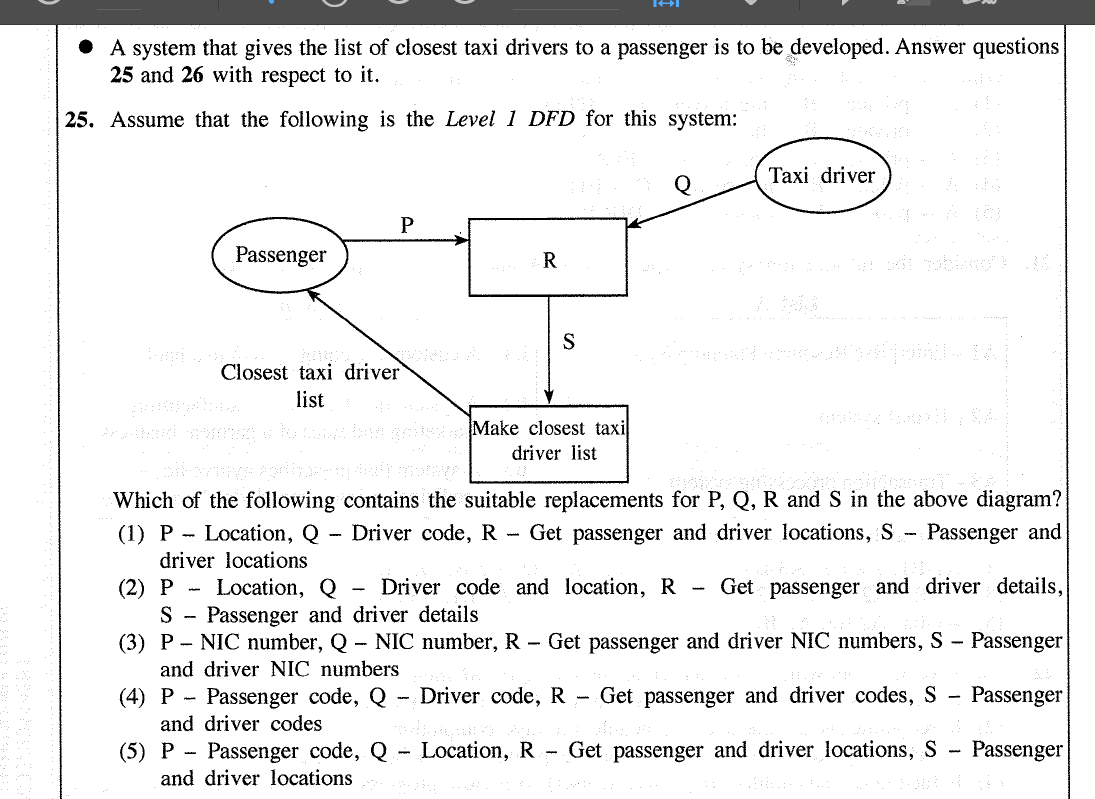
3. System Design

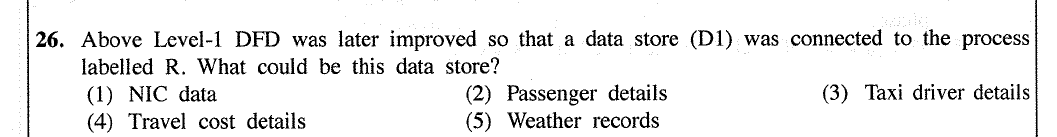
4. System Implementation/ Coding

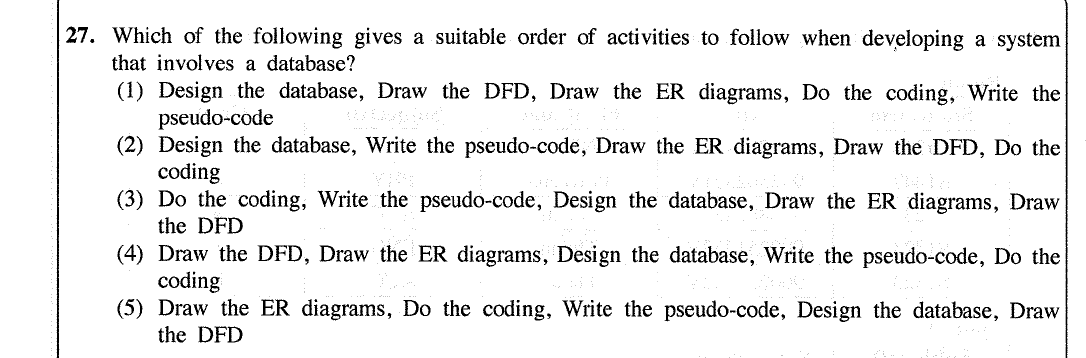
5. Testing

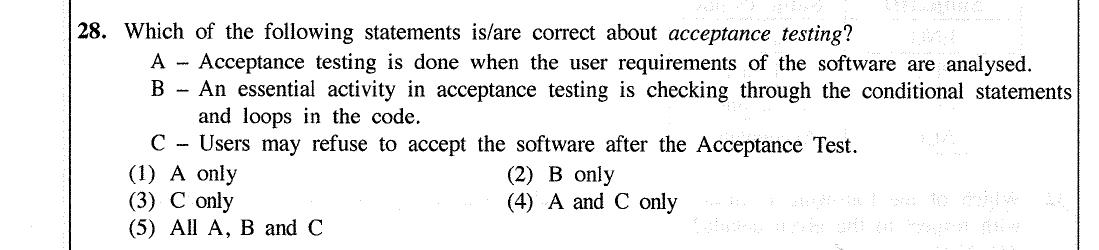
6. Deployment of the developed system

7. Maintenance phase









**Software Testing Types**

1. Unit Testing (Individual units are tested)

* Using white or black box testing
* Before integration testing
* Done by the developers

2. Integrated Testing

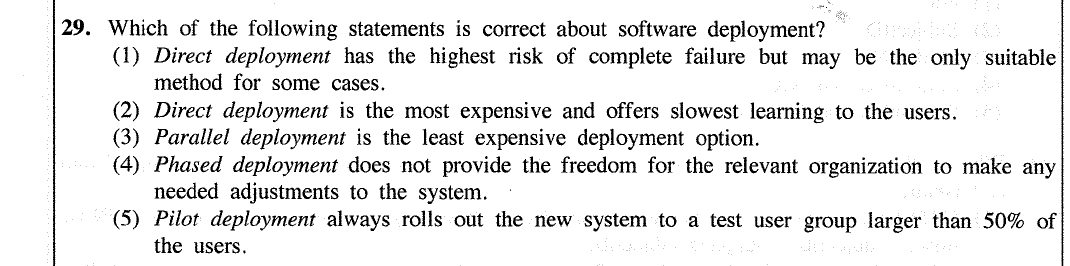
* Individual units are combined and tested as a group
* Using white or black testing
* Done by a specific tester or test team

3. System Testing

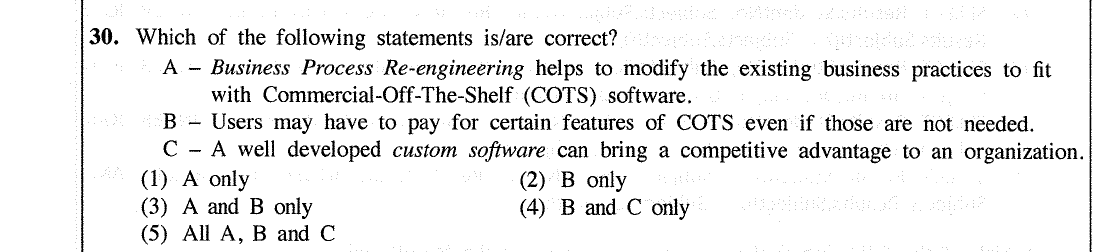
* Using black box testing
* Done by a team that is independent of the development team in order to measure the quality of the system unbiased

4. Acceptance Testing

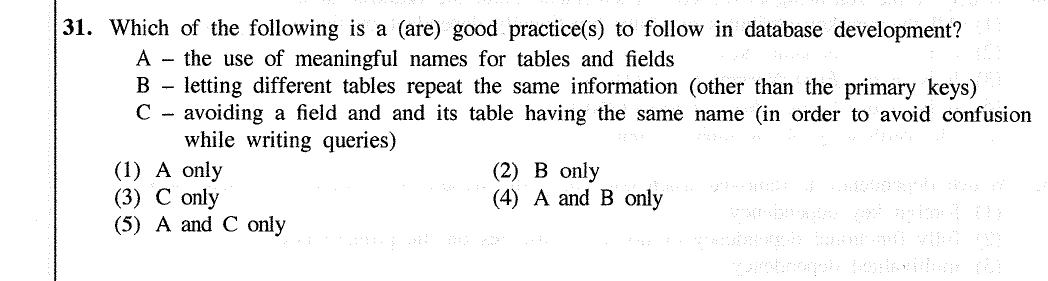
* Performed by the end user or the client to verify/ accept the system

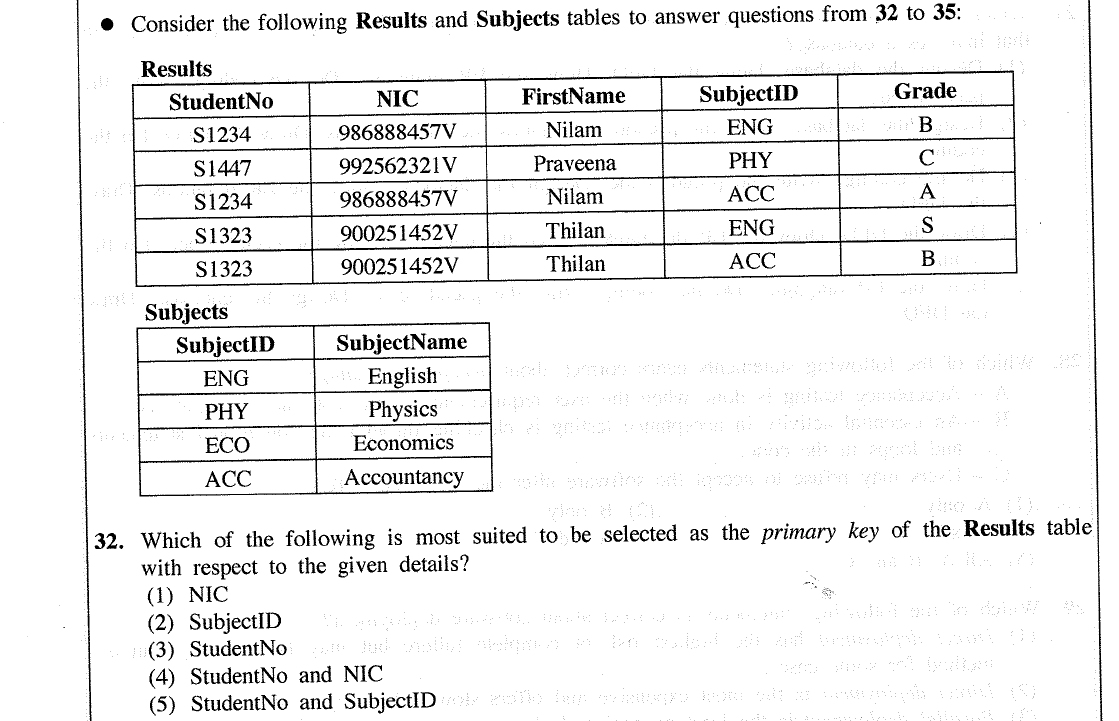


|  |  |
| --- | --- |
| Direct | Old system is discontinued and the new system will be used from that point onwards |
| Phase | Parts of the new system are implemented one by one over the time |
| Parallel | Old and new systems are used at the same time |
| Pilot | Entire system is used in one location |

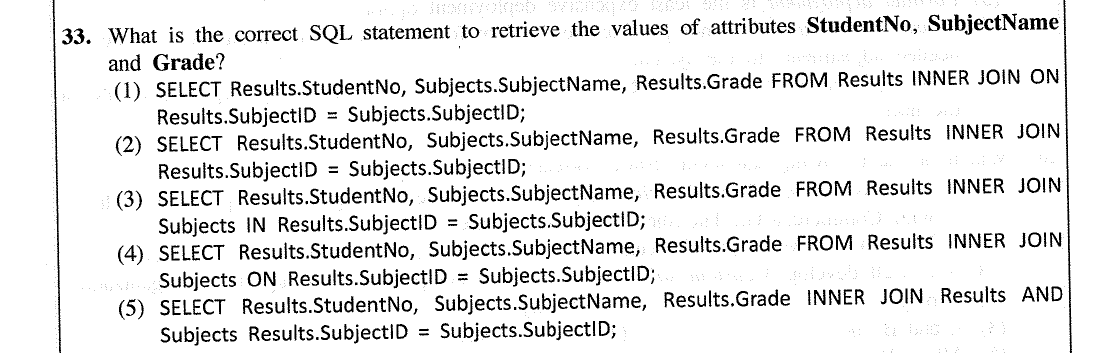


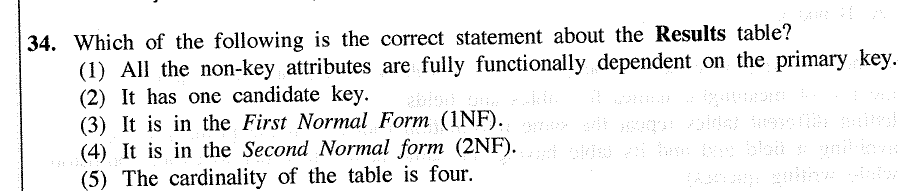
|  |  |
| --- | --- |
| **Commercial-Off-The-Shelf (COTS) software** | **Custom software** |
| Ready-made softwares | Made specifically for a individual or company |
| Initial cost of the software is comparatively lesser | Cost is higher |
| Easy to install and use | As this a brand new software, it won’t be easy to solve issues |
| All the features may not be relevant |
| Availability of supporting documentations |
| Availability of updates |



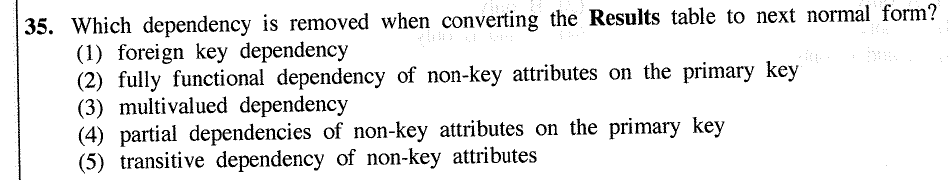


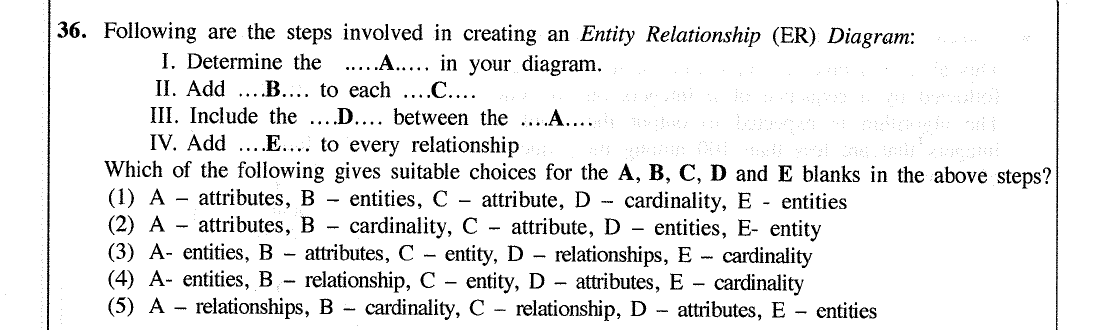
As one student can learn more than one subject, each record can uniquely identify if both StudentNo and SubjectID are the composite primary key.

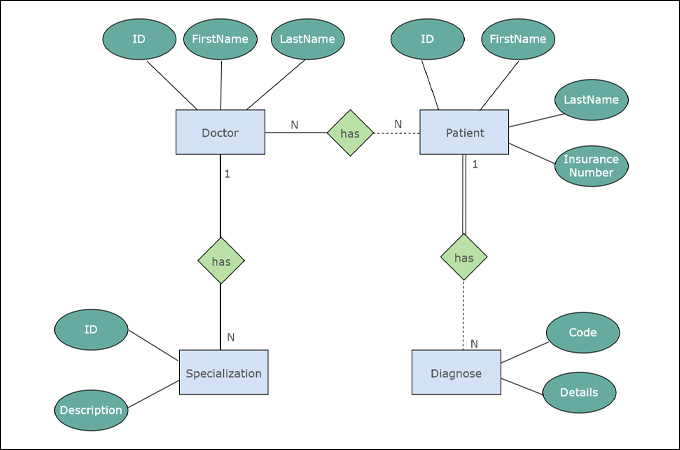




|  |  |
| --- | --- |
| Zero Normal Form | Not yet normalized. Have repeating attributes (such as author1, author2, author3) |
| First Normal Form | Primary key is repeated but there is a composite primary key 🡺 partial dependency |
| Second Normal Form | Np partial dependencies. Has transitive dependencies 🡺 non-prime attribute/ column doesn’t depend on the primary key but depends on another non-prime attribute |
| Third Normal Form | Fully functional dependencies |





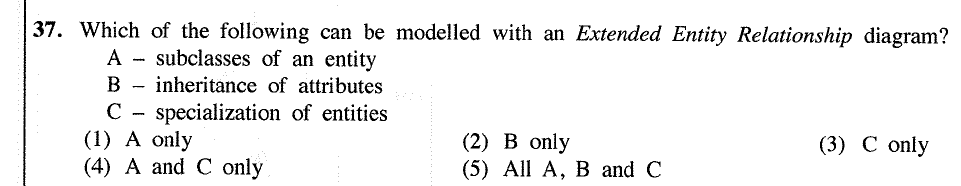


Represent the relationship

Each rectangle box represent an entity.

The ovals connected to each entity represent the attributes of each entity.

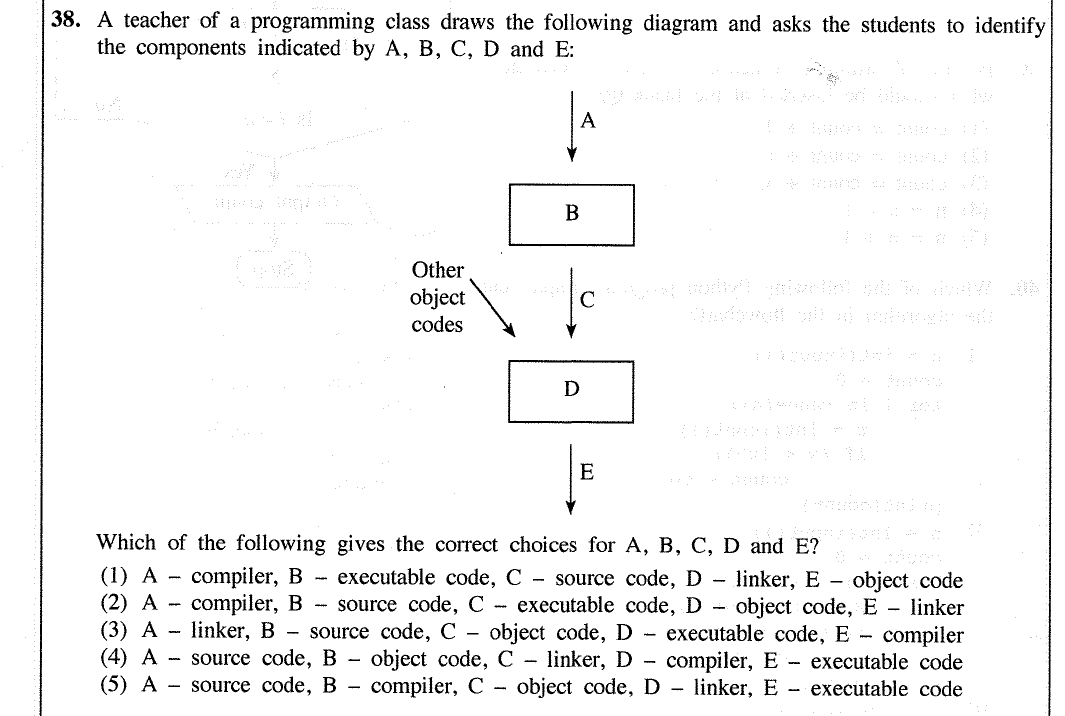
1 and N represent the cardinality



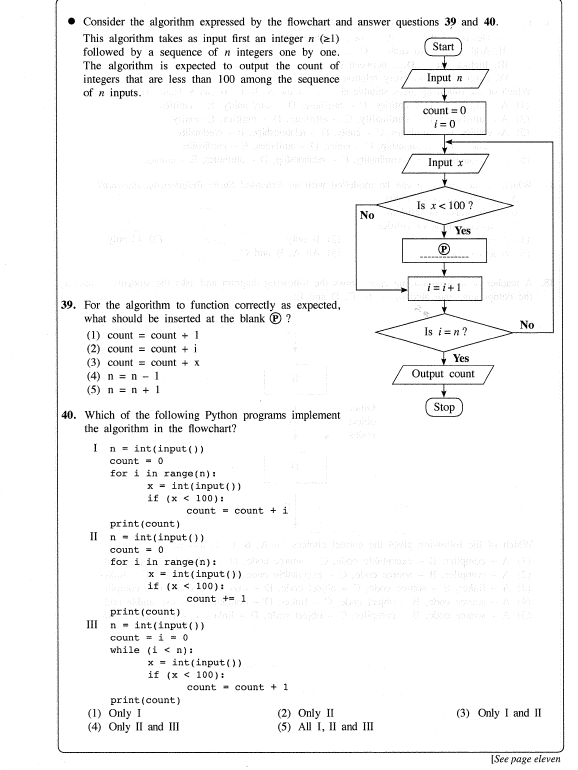
EER 🡪 Expansive version of ER diagrams

An EER diagram provides you *with all the elements of an ER diagram* while adding:

* Attribute or relationship inheritances
* Category or union types
* Specialization and generalization
* Subclasses and superclasses



|  |  |
| --- | --- |
| source code | The programs we (programmers) write which is not yet been compiled into an executable file |
| compiler | Converts the whole source code in one session and reports all the errors afterwards |
| object code | Compiled code that can be run on any computer with the same CPU architecture |
| linker | A program in a system which helps to link object modules of a program into a single object file |
| executable code | Machine code which can be executed by the computer |



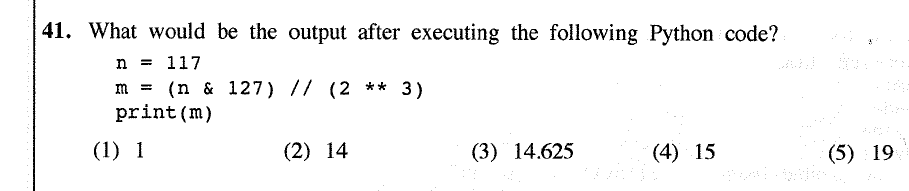
39. As the count have to increase after each relevant iteration, ‘P’ will be count = count + 1

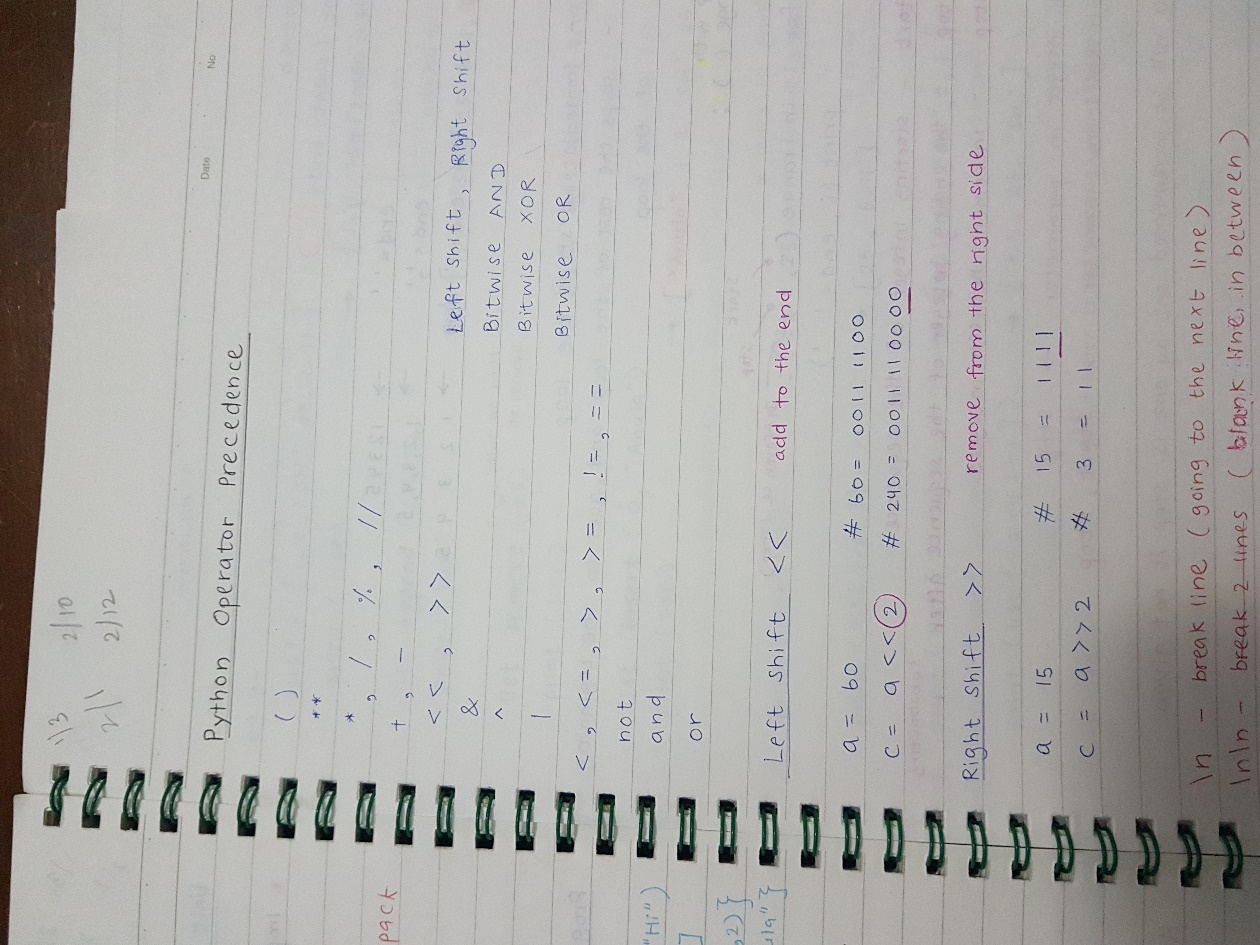
40.

The first code is incorrect as the count statement is incorrect. As it is count = count + i, the count will be counted incorrectly

The second code is correct

The third code is incorrect because the while statement is incorrect

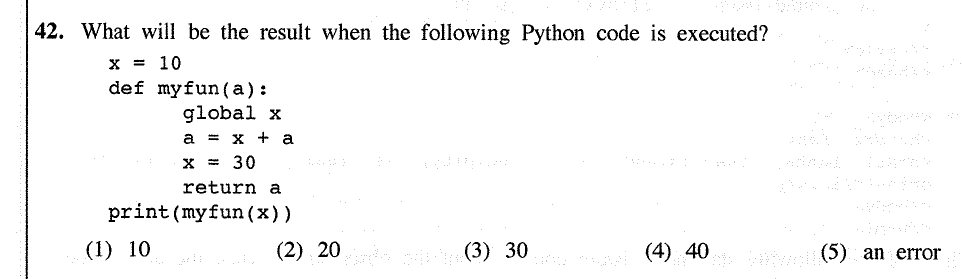


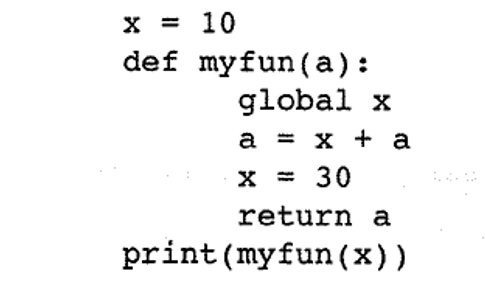


( 117 & 127 ) / / ( 2 \* \* 3 )

117 / / 8

14



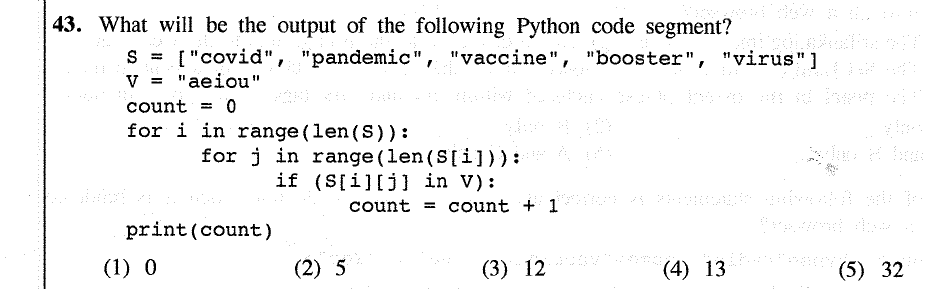


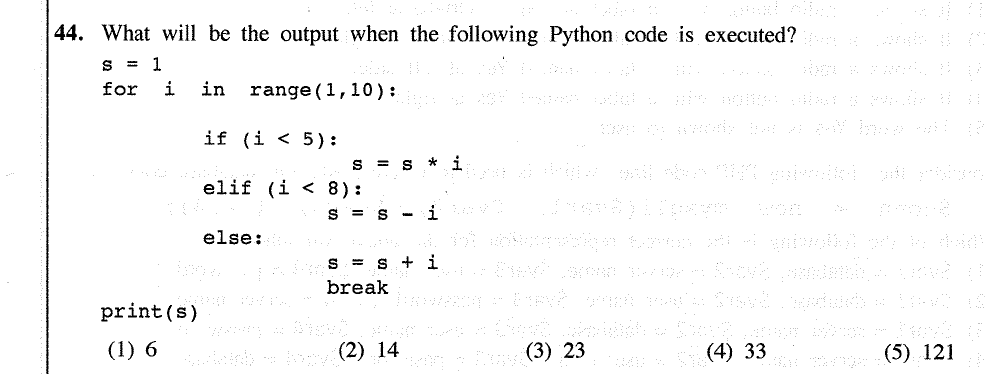
10

10

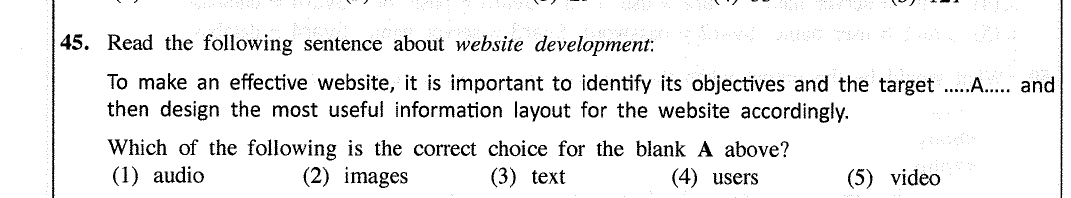
a = 10 + 10 🡪 a = 20

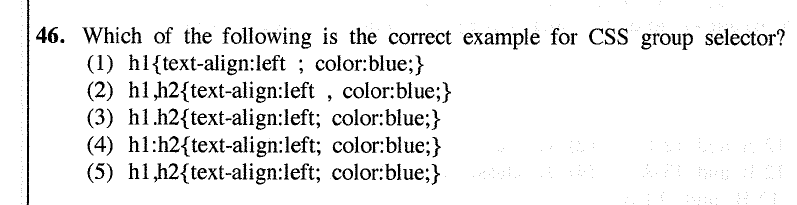
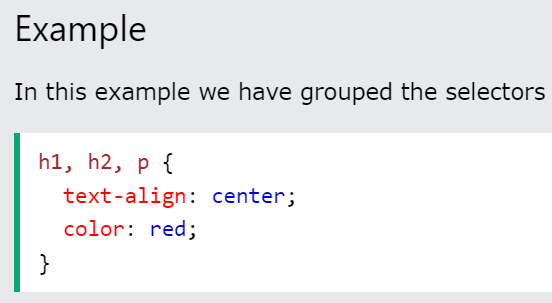
20

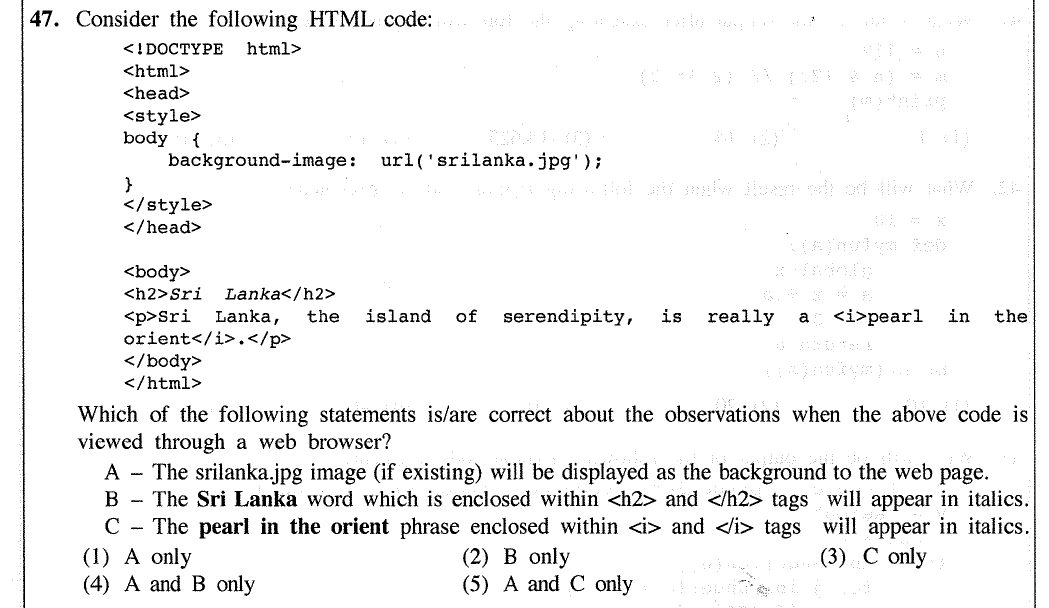


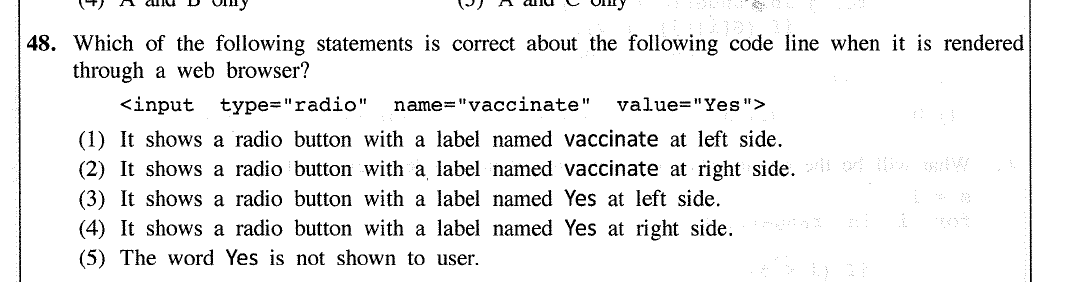


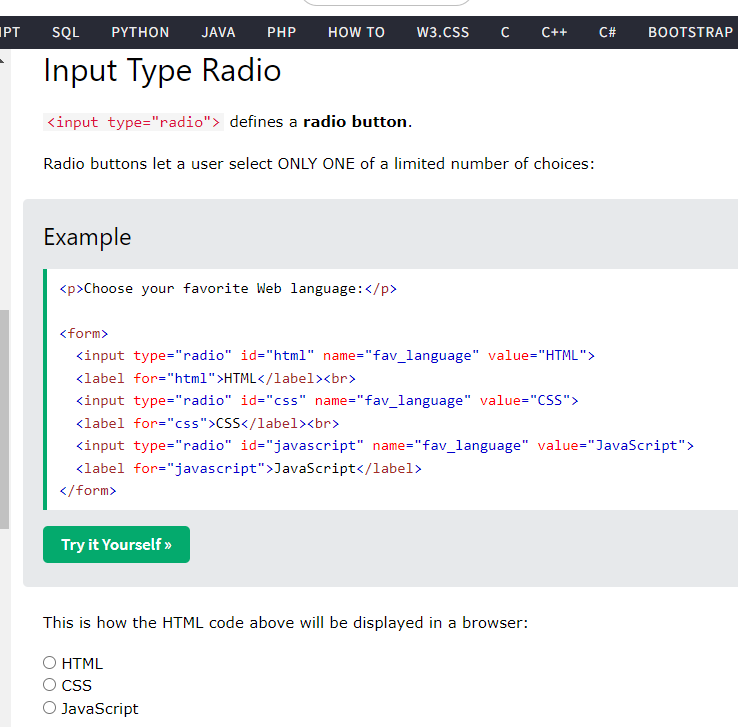
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **s** | **i in range (1,10)** | **if (i<5)** | **s = s \* i** | **elif (i<8)** | **s = s - i** | **else**  **s = s + i** | **break** | **print (s)** |
| 1 | 1 | 1 < 5 (Yes) | 1 = 1 \* 1 |  |  |  |  |  |
|  | 2 | 2< 5 (Yes) | 2 = 1 \* 2 |  |  |  |  |  |
|  | 3 | 3 < 5 (Yes) | 6 = 2 \* 3 |  |  |  |  |  |
|  | 4 | 4 < 5 (Yes) | 24 = 6 \* 4 |  |  |  |  |  |
|  | 5 | 5 < 5 (No) | - | 5 < 8 (Yes) | 19 = 24 - 5 |  |  |  |
|  | 6 | 6 < 5 (No) | - | 6 < 8 (Yes) | 13 = 19 - 6 |  |  |  |
|  | 7 | 7 < 5 (No) | - | 7 < 8 (Yes) | 6 = 13 - 7 |  |  |  |
|  | 8 | 8 < 5 (No) | - | 8 < 8 (No) | - | 14 = 6 + 8 | break | 14 |
|  | 9 |  |  |  |  |  |  |  |

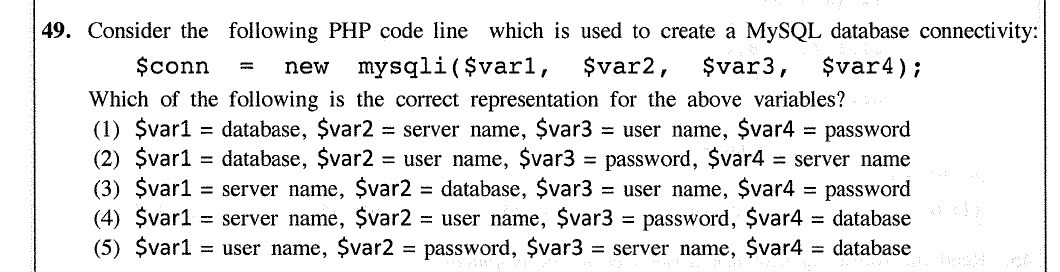




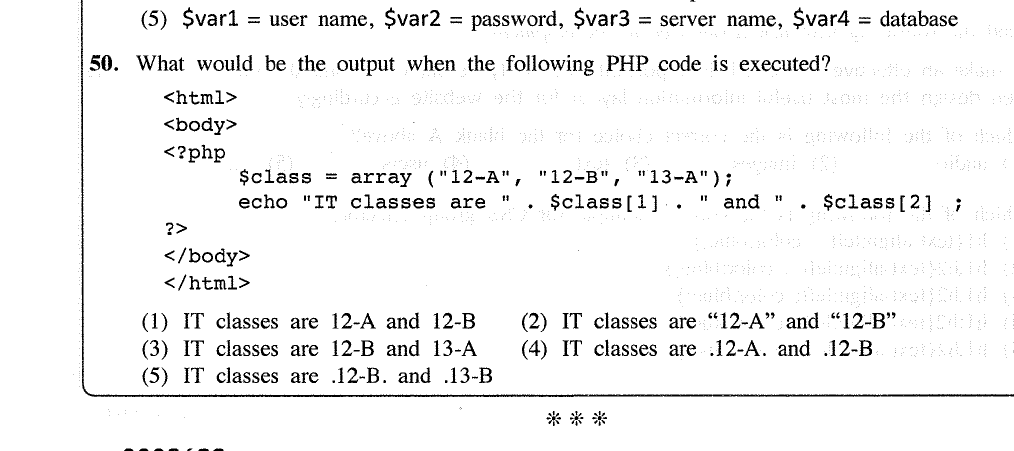








To create a MySQL database connectivity, we have to access the server first. Therefore, the server name is mentioned. Then, we have to enter the user name and password of the account. Afterwards, we can access the database.



$class[0] = 12-A

$class[1] = 12-B

$class[2] = 13-A