1. Briefly explain these terminologies. If they are acronyms, also write what they stand for. (24 points)

(a) Software agent - A software agent is an autonomous process capable of reacting to, and initiating changes in, its environment, possibly in collaboration with users and other agents.

(b) LDAP - Lightweight Directory Access Protocol (LDAP) is a protocol for directory services enabling anyone to locate organizations, individuals, and other resources in a network.

(c) NTP - Network Time Protocol (NTP) is a protocol that is used to synchronize computer clock times in a network of computers.

(d) 2PL - Two-phase Locking (2PL) is a concurrency control method that acquires all necessary locks in growing phase, releases locks in shrinking phase.

(e) PGP - Pretty Good Privacy (PGP) is a popular program used to encrypt and decrypt e-mail over the Internet.

(f) IIOP - IIOP (Internet Inter-ORB Protocol) is a protocol that makes it possible for distributed programs written in different programming languages to communicate over the Internet.

2. Fill out the following table of a comparison between different systems: (20 points)

<table>
<thead>
<tr>
<th>Item</th>
<th>Distributed OS</th>
<th>Network OS</th>
<th>Middleware-based DS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multicomputer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Degree of transparency</td>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Same OS on all nodes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Number of copies of OS</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Basis for communication</td>
<td>Messages</td>
<td>Files</td>
<td>Model specific</td>
</tr>
<tr>
<td>Resource management</td>
<td>Global, distributed</td>
<td>Per node</td>
<td>Per node</td>
</tr>
<tr>
<td>Scalability</td>
<td>Moderately</td>
<td>Yes</td>
<td>Varies</td>
</tr>
<tr>
<td>Openness</td>
<td>Closed</td>
<td>Open</td>
<td>Open</td>
</tr>
</tbody>
</table>

3. Complete the following table listing the seven layers in the OSI 7-Layer Reference Model. Give an example protocol for each layer. (10 points)

<table>
<thead>
<tr>
<th>Layer level</th>
<th>Layer Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 7</td>
<td>Application Layer</td>
<td>HTTP</td>
</tr>
<tr>
<td>Layer 6</td>
<td>Presentation Layer</td>
<td>ASN.1</td>
</tr>
<tr>
<td>Layer 5</td>
<td>Session Layer</td>
<td>WSP</td>
</tr>
<tr>
<td>Layer 4</td>
<td>Transport Layer</td>
<td>TCP/UDP</td>
</tr>
<tr>
<td>Layer 3</td>
<td>Network Layer</td>
<td>IP</td>
</tr>
<tr>
<td>Layer 2</td>
<td>Data Link Layer</td>
<td>PPP</td>
</tr>
<tr>
<td>Layer 1</td>
<td>Physical Layer</td>
<td>RS-232-C</td>
</tr>
</tbody>
</table>
4. Consider the following code:

```c
int f(x,y)
{
    x = x + 1 ;
    y = x * y - 5;
    return(x + y);
}

main()
{
    int a, b;
    a = 2;
    b = f(a, a);
    printf("a = %d, b = %d \n",a,b);
}
```

What will be printed if the parameters are passed using:
(a) call-by-value?
(b) call-by-reference?
(c) call-by-copy/restore using Pascal semantics?

Which of the above parameter passing method(s) cannot be easily implemented using remote procedure calls? Explain. (8 points)

(a)  
    i.  a = 2, b = 4  
    ii. a = 4, b = 8  
    iii. a = 3, b = 4 (Pascal semantics)

(b) Call-by-reference cannot be easily implemented because the reference addresses are different between the local and remote machines.

5. (a) What is an idempotent operation? (3 points)

(b) For each of the following applications, do you think at-least-once semantics or at most once semantics is best? Discuss. (6 points)

    i.  Write a block of data into a file  
    ii. Request the number of users currently logged into a remote system  
    iii. Deposit money into a bank account

(a) An idempotent operation is an operation that can be performed repeatedly with the same effect.
(b) For (i) and (ii), at least once is best. There is no harm trying over and over. For (iii), it is best to give it only one try. If that fails, the user will have to intervene to clean up the mess.
6. (a) What is persistent communication? In which case(s) do we use persistent communication? (6 points)
(b) Suppose you are designing a media transfer protocol over TCP and UDP. Define your phases for this protocol. How do you use TCP and UDP in each phase. (12 points)

(a) i. Persistent communication is a type of communications in which a message is stored at a communication server as long as it takes to deliver it at the receiver.
   ii. Persistent communication can be used in the applications in which the sending or receiving applications need not be executing during the transmission of messages such as e-mail or network news.
(b) i. Connection establishment phase: TCP can be employed for reliable connection establishment.
   ii. Media transfer phase: UDP can be employed for less transfer overhead.

7. (a) What is a logic clock? (3 points)
(b) Why are logic clocks required in distributed systems? (4 points)
(c) How does Lamport synchronize logical clocks? (8 points)
(d) Which events are said to be concurrent in Lamport timestamps? (3 points)

(a) A logic clock is not really a physical clock but is specified by the event timestamps.
(b) In many situations it is the internal consistency of the clocks that matters, not whether they are particular close to the real time. Then logic clocks are required.
(c) Lamport synchronizes logical clocks by happens-before relations:
   i. If A and B are events in the same process, and A executed before B, then A \rightarrow B is true.
   ii. If A represents sending of a message and B is the receipt of this message, then A \rightarrow B.
(d) When nothing can be said about when the events happened or which happened first, these events are said to be concurrent.

8. Briefly explain crash, omission, and arbitrary failures and give an example for each of them. (18 points)

<table>
<thead>
<tr>
<th>Type of failure</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crash failure</td>
<td>A server halts, but is working correctly until it halts</td>
<td>An Operating System comes to a grinding halt.</td>
</tr>
<tr>
<td>Omission failure</td>
<td>A server fails to respond to incoming requests</td>
<td>A send buffer overflows while the server was not prepared.</td>
</tr>
<tr>
<td>Arbitrary failure</td>
<td>A server may produce arbitrary responses at arbitrary times</td>
<td>A printer functions but prints incorrectly.</td>
</tr>
</tbody>
</table>

9. (a) Describe reasons for replication. What is main problem of replication? (9 points)
(b) Explain how replication in DNS takes place, and why it actually works so well. (6 points)

(a) The reasons for replication are it creases the reliability of a system and improves the performance of a system. The main problem is the need to maintain consistency of replicated data.
(b) The basic idea is that name servers cache previously looked up results. These results can be kept in a cache for a long time, because DNS makes the assumption that name-to-address mappings do not change often.

10. (a) Encrypt the following plaintext using a transposition cipher based on the key COUGAR. (8 points)

plaintext = E N J O Y Y O U R S U M M E R V A C A T I O N S

C O U G A R
2 4 6 3 1 5
E N J O Y Y
O U R S U M
M E R V A C
A T I O N S

ciphertext = YUAN EOMA OSVO NUET YMCS JRRI
(b) Using the RSA public key cryptography algorithm. (8 points)
   i. If \( p = 5 \) and \( q = 11 \), list four legal values for \( d \).
   ii. If \( p = 7 \), \( q = 13 \), and \( d = 7 \), find \( e \).
      i. \( z = (p - 1) \times (q - 1) = 4 \times 10 = 40 \), \( d = 7, 11, 13, 17, 19, 23 \)
      ii. \( z = (p - 1) \times (q - 1) = 6 \times 12 = 72 \), \( 7 \times e \mod 72 = 1, 73, 145, 217, 7 \times e = 217, e = 31 \).

(c) Why are asymmetric encryption algorithms usually used for authentication and symmetric encryption algorithms for document secrecy? (4 points)

Asymmetric encryption algorithms provide the public-key which facilitates authentication. In addition, they are usually too slow for document encryption. Symmetric encryption is usually fast enough to use secret key to encrypt document.

11. (a) What is the topic of your research project? (4 points)
    (b) What are the major goals of your research project? (6 points)
    (c) What is the scope of your research project? (6 points)

12. Consider an application that implements a remote phone book. The phone book offers 4 operations: count to return the total count of entries (personal or business name), insert to insert an entry and its phone number, delete to delete an entry, and lookup to search for an entry and return its phone number.

   (a) Use Sun RPC to specify this application. (10 points)
   (b) Specify this application in a Java RMI interface file. (10 points)

   (a) Sun RPC

   struct directory_entry {
      string name;
      int phoneno;
   };

   program RDIRPROG {
      version RDICTVERS {
         int COUNT(void) = 1;
         int INSERT(directory_entry) = 2;
         int DELETE(string) = 3;
         int LOOKUP(string) = 4;
      } = 1;
   } = 0x30090949;

   (b) Java RMI interface file

   import java.rmi.*;
   public interface DirectoryInterface extends Remote {
      public long count() throws RemoteException;
      public long insert(String name, int phoneno) throws RemoteException;
      public long delete(String name) throws RemoteException;
      public int lookup(String name) throws RemoteException;
   }
Consider the following header file (day.h) and C code. (24 points)

```c
#define NAMESIZE 40

typedef struct {
    int day;
    int month;
    int year;
} date;

struct person {
    char name[NAMESIZE];
    date birthdate;
};

#include <day.h>
extern date birthday_dblookup(char *); /* database lookup by name */
extern char **list_dblookup(date); /* database lookup by date */

date birthday_lookup (char *name) {
    return(birthday_dblookup(name));
}

char **list_lookup(date birthday) {
    return(list_dblookup(birthday));
}

main()
{
    char name[NAMESIZE];
    date birthday;
    char **list;
    int i = 0;

    printf("Name: ");
    scanf("%s", name);
    birthday = birthday_lookup(name);
    printf("Hello! %s, your birthday is %d-%d-%d.\n", 
           name, birthday.day, birthday.month, birthday.year);
    list = list_lookup(birthday);
    printf("Following persons have the same birthday as you:\n");
    while (list[i] != '\0') printf("%s\n", list[i++]);
}
```

(a) Use CORBA IDL to specify this application (The main function is not required).

(b) Suppose BirthdayDBLookup and ListDBLookup are two available Java methods corresponding to C
    birthday_dblookup and list_dblookup functions respectively. Write the Java servant class that im-
    plements the interfaces and methods defined in the above COBRA IDL file.
(a) CORBA IDL (Birthday.idl)

```idl
module Birthday_App {

    struct date {
        int day;
        int month;
        int year;
    };

    struct person {
        string name;
        date birthdate;
    };

    typedef sequence<string> StringSeq;

    interface Birthday {
        date BirthdayLookup (in string name);
        StringSeq ListLookup(in date birthday);
    }
}
```

(b) Servant class

```java
class BirthdayImpl extends BirthdayPOA {

    private ORB orb;

    public void setORB(ORB orb_val) {
        orb = orb_val;
    }

    // implement BirthdayLookup() method
    public date BirthdayLookup(String name) {
        return BirthdayDBLookup(name);
    }

    // implement ListLookup() method
    public String[] ListLookup(date Birthday) {
        return ListDBLookup(Birthday);
    }
}
```