UNIVERSITY
OF
JOHANNESBURG

FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	CSC2B10 COMPUTER SCIENCE 2B
CAMPUS	AUCKLAND PARK CAMPUS (APK)
	SSA EXAM

DATE: 2017-01

SESSION: 08:30 - 10:30

ASSESOR(S): DR D.T. VAN DER HAAR
MR M. CILLIERS

MODERATOR: MR A. MAGANLAL

DURATION: 120 MINUTES

MARKS: 100

Please read the following instructions carefully:

1. Answer **all** the questions
2. Write *cleanly* and *legibly*.
3. You may use a non-programmable calculator to answer the questions.
4. This paper consists of 6 pages.

SECTION A - Theory

QUESTION 1

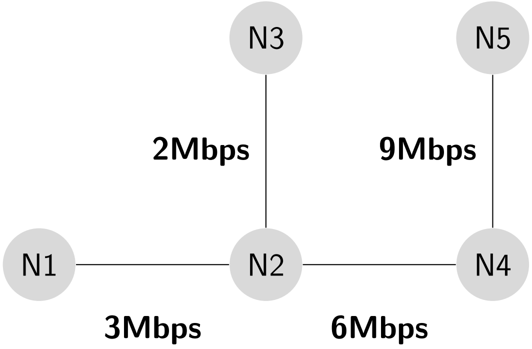
(a) *What* is the **Internet**? [02]

(b) *List* three (3) of the **layers** in the Internet Protocol Stack. [03]

Total: 5

QUESTION 2

Assume there is a copper network with 5 nodes (N1, N2, N3, N4 and N5) and the transmission rates between these nodes are as follows:



It is also determined that the distances between the nodes are as follows:

- N1-N2: 3km
- N2-N3: 80km
- N2-N4: 3km
- N3-N5: 9km

Answer the following questions:

(a) *Determine* the **approximate transmission rate** when communicating between N1 and N5. [01]

(b) Taking this **approximate transmission rate** into account, how *long* (in seconds) will it take to transfer a 360 Megabit **file** from node N1 to N5? [02]

(c) If it is determined that the copper installed in this network **propagates** a signal at a speed of 300 000 km/s. *Calculate* the **propagation delay** for communications between N1 to N5. [03]

(d) Assuming there is no nodal processing delay or queueing delay, *calculate* the **total time** taken to transfer a 360 Megabit file from from N1 to N5? [04]

Total: 10

QUESTION 3

Dicsuss the **peer-to-peer** architecture, along with what makes it **better** than the client-server architecture.

Total: 5

QUESTION 4

The following question is about the **FTP** protocol.

(a) *What* does **FTP** stand for? [01]

(b) What port(s) does the **FTP** protocol run on? [02]

(c) *What* is the purpose of the **FTP** protocol? [02]

Total: 5

QUESTION 5

(a) *Discuss* how **connection-oriented** demultiplexing works within the context of the transport layer. [04]

(b) *Define* **flow control** within the context of the transport layer. [02]

(c) *Describe* for each of the following mechanisms, which problem they address in order to achieve reliability: [04]

1. Timers
2. Pipelining
3. Acknowledgments
4. Sequence numbering

Total: 10

QUESTION 6

The table below represents the payload of a UDP segment. Calculate the **sum** of the following two 16-bit integers, along with their associated 1s complement **checksum**:

Write down just the sum and checksum in your answer sheet

Number 1	1	0	1	0	0	1	0	1	0	1	0	0	1	0	1	1
Number 2	1	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0
Sum																
Checksum																

Total: 5

QUESTION 7

(a) Why is a datagram buffer needed on the output port of a router? [2]

(b) *Explain* why distance vector algorithms are susceptible to interference. [4]

(c) *Discuss* how *ICMP* can be used to find a routing path between two hosts. [4]

Total: 10

QUESTION 8

Given the following **IP address** and **CIDR**, answer the questions that follow:

42.1.220.13/23

(a) Provide this address in **binary** notation. [02]

(b) How many hosts can this network **accommodate**? [02]

(c) Assuming classful addressing was used, what **class** does this address belong to? [02]

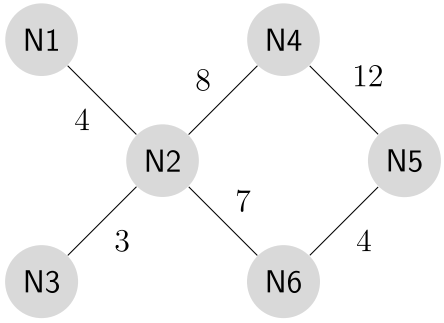
(d) Calculate the **network address** of this block in dotted decimal notation. [02]

(e) Calculate the **broadcast address** of this block in dotted decimal notation. [02]

Total: 10

QUESTION 9

Given the below network **routing graph** (with costs), answer the following questions that follow:



(a) What is the path with the **least cost** when communicating between N1 and N5. Is this the **only** cost effective path? [02]

(b) Given the local datagram **forwarding table** for node N2 below and the destination address is 11100000 01110011 00100000 00010011, which link will this packet be forwarded to? What is this lookup technique called? [03]

Destination Address range	**Output Link Interface**
10100111 01110011 00100000 00011***	N1
10100111 01110011 00100000 000110**	N3
10100111 01110011 00100000 000111**	N4
Otherwise	N6

Total: 5

QUESTION 10

(a) **Describe** the steps that will be taken during reverse ARP process. [4]

(b) **Name** and **describe** two kinds of errors that can occur when transmitting data on the link layer. Your discussion should **include** methods that can be used to detect these errors. [6]

Total: 10

SECTION B - Practical

QUESTION 11

(a) *Discuss* how resource **security** is handled for an android application [03]

(b) Which windows command can be used to check the content of the rooting table. [02]

Total: 5

QUESTION 12

Provide Java source code for a method named `sendImage` which will send an **image** using TCP and ImageIO . The destination address, destination port and a BufferedImage is passed as parameter. You can assume all relevant packages have been imported.

Total: 10

QUESTION 13

The code below illustrates login functionality that was implemented on a server.

Fill in the missing code in your answer booklet.

```java
public class SVRHandler implements Runnable
{
    Socket client=null;
    BufferedReader reader;
    PrintWriter writer;
    BufferedOutputStream os;

    public SVRHandler(Socket socket)
    {
    client = socket;
    try
    {
      reader = (___________(A)_______________[2])
      os = new BufferedOutputStream(client.getOutputStream());
      writer = new PrintWriter(os);

    }
    catch(...)
    {
      ex.printStackTrace();
    }
    }

    public void run()
    {
    boolean running = true;
    try
    {
      String command = (___________(B)_______________[2]);
      String commandSec[] = command.split("\\s");
      if(commandSec.length == 0)
      {
        writer.println("ERROR");
        writer.flush();
        running = false;
      }
      else if(commandSec[0].equals("LOGIN"))
      {
        if(matchUser(commandSec[1], commandSec[2]))
        {
          writer.println("OK");
          (___________(C)_______________[1])
        }
        else
        {
        writer.println("ERROR");
        writer.flush();
        running = false;
        }
      }
      while(running)
      {
        command = ...;
        commandSec = command.split("\\s");
        if(commandSec.length == 0)
        {
          writer.println("ERROR");
          writer.flush();
          running = false;
        }
        ....
        ....
        else if(commandSec[0].equals("EXIT"))
        {
          running = false;
        }
      }
    }
    catch(___________(D)_______________[1])
    {
      ex.printStackTrace();
    }
    finally
    {
      try
      {
        reader.close();
        os.close();
        (___________(E)_______________[2])
      }
      catch(...)
      {
        ex.printStackTrace();
      }
    }
    }

    private boolean matchUser(___________(F)_______________[2])
    {
    boolean found = false;
    File userFile = new File("users.txt");
    try
    {
      Scanner scan = new Scanner(userFile);
      while(scan.hasNextLine()&&!found)
      {
        String line = scan.nextLine();
        String lineSec[] = line.split("\\s");
        if(lineSec.length >= 2)
        {
          if((username.equals(lineSec[0]))&&(password.equals(lineSec[1])))
          {
            found = true;
          }
        }
      }
      scan.close();
    }
    catch(...)
    {
      ex.printStackTrace();
    }

    return found;
    }

}
```

Total: 10

The End!