ANSWER SHEET

These problems were generated by TAs and instructors in previous semesters. They may or may not match the actual difficulty of problems on Test5.

Levels of Concurrency

1. Answer the following True/False questions and explain the reasoning behind your answer.
    a. In multitasking, concurrent programs are run on a single CPU
    ANSWER:
    False. In multitasking, the programs are not actually run concurrently, they just appear to be.

    b. Multitasking and multiprocessing are synonyms meaning the same thing.
    ANSWER:
    False. Multitasking involves seemingly concurrent programs on a single CPU. Multiprocessing involves concurrent programs on multiple CPUs.

    c. A CPU has a single logic unit for each basic arithmetic operation (ex: multiplication, addition)
    ANSWER:
    False. Logic units are a set of circuits that can perform basic arithmetic operations, but CPUs have many duplicates of these (possibly hundreds of logic units that all perform the same operation) - this allows for concurrent programs

    d. Concurrency means that programs are running at the exact same time.
    ANSWER:
    True. Given definition of concurrency.

2. If the following algorithm were made concurrent, what is the minimum number of time steps it could take? List the steps (which operation(s) would be done in each time step) and explain. **2*(6+7)-9**

ANSWER:
3 total steps and 3 time steps. Since each operation relies on a previous operation, no steps can be run concurrently.
1. 6+7=13
2. 2*(13)=26
3. 26-9=17

3. What is the last level of concurrency? Briefly explain what this level of concurrency is and what are some applications we can see in daily life.

ANSWER:
The last level of concurrency is distributed computing. Distributed computing is a model in which components of a software system are shared among multiple computers to improve efficiency and performance. Tasks are distributed among several computers. This model is usually used by big Tech companies, such as Google and Amazon, to handle complex actions and thousands of customers at the same time.

4. Which of the following best describes the features of a distributed system? **Select all that apply.**

   - ❏ Has many connected computers.
   - ❏ Has only one computer with many cores.
   - ❏ Can perform multitasking.
   - ❏ Can perform multiprocessing.
   - ❏ Can perform pipelining of tasks.
   - ❏ None of the above.

Parallel Programming

1. What is the definition of deadlock? How can you usually fix deadlock?

ANSWER:
Deadlock is when two or more processes are waiting for the same resource that other processes in the group already hold, resulting in waiting forever and being unable to proceed. This can be avoided if you impose an order that programs always follow when requesting resources.

2. A factory that makes action figures is set up so that every worker does three tasks: collecting the components (5min), gluing them together (5min), and painting the resulting figure (5min). Gluing requires some tidy-up time before switching to a new task (5min), and painting requires both set-up and tidy-up time when switching to a new task (5min each), so the worker's schedule over an hour currently looks like this:

	00:00	00:05	00:10	00:15	00:20	00:25	00:30	00:35	00:40	00:45	00:50	00:55
X	C	G	T	S	P	T	C	G	T	S	P	T
Y	C	G	T	S	P	T	C	G	T	S	P	T
Z	C	G	T	S	P	T	C	G	T	S	P	T

X, Y, and Z are workers. C is collect, G is glue, P is paint, T is tidy, and S is set-up

How many fully-set-up figures can 3 workers currently produce in 1 hour?

ANSWER:
6

Use the concept of **pipelining** to adjust this schedule so that the workers can generate more figures in an hour. Fill in the table below using the same codes we used above. Your new schedule does not need to be ideal; it should just be better than the old one.

	00:00	00:05	00:10	00:15	00:20	00:25	00:30	00:35	00:40	00:45	00:50	00:55
X	C	C	C	C	C	C	C	C	C	C	C	C
Y		G	G	G	G	G	G	G	G	G	G	G
Z		S	P	P	P	P	P	P	P	P	P	P

How many fully-set-up figures can 3 workers produce with your new schedule in 1 hour?

ANSWER:
10

Internet

1. What does it mean to be fault tolerant? At a high level (no specific examples needed) how is the Internet structured to allow it to be fault tolerant?

ANSWER:
To be fault tolerant means to be able to handle and recover from things going wrong, for instance through backing things up and performing checks to ensure data integrity. The Internet achieves fault tolerance through being implemented as a distributed system.

2. What is a packet? Name one thing that could go wrong when sending a packet, and how the Internet deals with it.

ANSWER:
A packet is a package of data sent to a particular IP Address. It includes information about where it's coming and going from and has a message (the data). A packet could get corrupted, so we use a parity bit to check if it has arrived as expected.

3. What is the relationship between an IP Address and a URL?

ANSWER:
A URL acts as a website's nickname. These are easy to read and remember such as google.com or youtube.com. On the other hand, a website's IP Address is a series of numbers that acts as the real name for a site. The IP Address of a specific website is static and can be used to locate where a website is being hosted.

4. For each of the following statements, select whether it is True or False.
    a. Websites split up data into a few large packets based on content before sending them through the internet.
    ANSWER:
    False

    b. Packets all arrive at your browser in the correct order, to support buffering.
    ANSWER:
    False

    c. Packets might all take different paths through routers to reach your computer.
    ANSWER:
    True

Encryption

1. Explain why RSA is secure. What might change that might make RSA NOT secure?

ANSWER:
RSA is secure because it uses extremely large primes to create a number n that is difficult to factorize. If we get better at factoring and can speed it up, then RSA will not be secure.

2. Explain how asymmetric keys can be more secure than symmetric keys

ANSWER:
Shared keys have the flaw that since it is known by both parties it is easier for it to be intercepted upon sending the key to the recipient. Thus, if this key is intercepted and decrypted then this encryption method will no longer be secure. However since asymmetric public keys are already public and only used for encryption then transport of the message is safe and cannot be decrypted during transport since the decryption (private key) is private and known only to the recipient.

3. What is the difference between data privacy and data security? What do the two have in common?

ANSWER:
Data privacy is purely individual, while data security involves the communication between two parties. They both have the same goal that no third party would be able to read data sent across the internet.

4. How can encryption help to stop Man-in-the-Middle Attack?

ANSWER:
Man-in-the-Middle Attack happens because the packets are not encrypted. That's one of the reasons why they often occur in public wifi which are not encrypted.

5. Given the following scenarios, determine what type of security attack it is: DDOS or Man in the Middle.
    a. An attacker sets up a router and allows other people to connect to it. The attacker reads every packet that people send and records usernames and passwords.
    ANSWER:
    Man in the Middle

    b. An attacker wants to prevent people from receiving website content, so they send many packets to the website's server to overwhelm it.
    ANSWER:
    DDOS

6. Given the code below, what string will the variable `msg` hold after the code runs?

```
def encrypt(s, shift):
    result = ""
    for c in s:
        if c != " ":
            result = result + chr(ord(c) + shift)
        else:
            result = result + c
    return result

s = "Hello"
msg = encrypt(s, 3)
```

ANSWER:
"Khoor"

Write a single line of code that will correctly decrypt the message in the variable `msg` and print the result. You may not use the variable `s` in this line of code or hardcode the original message in any way.

Hint: you should call the function `encrypt` again, but what should `shift` be?

ANSWER:
`print(encrypt(msg, -3))`