1.1
You are to write an IAS program to compute the results of the following equation.

$$Y = \sum_{X=1}^{N} X$$

Assume that the computation does not result in an arithmetic overflow and that X, Y, and N are positive integers with N ≥ 1. Note: The IAS did not have assembly language, only machine language.

a. Use the equation Sum(Y) = (N(N + 1)) / 2 when writing the IAS program.

```
N = M[0]
Y = M[1]
1 = M[2]
2 = M[3]

LOAD M[0]
ADD M[2]
MUL M[0]
DIV M[3]
STOR M[1]
JUMP M[6,20:39]
```

b. Do it the “hard way,” without using the equation from part (a).

```
N = M[0]
1 = M[1]
i = M[2]
Y = M[3]

LOAD M[0]
SUB M[2]
JUMP + M[6,0:19]
JUMP + M[5,20:39]
LOAD M[2]
ADD M[1]
STOR M[2]
ADD M[3]
STOR M[3]
JUMP M[4,0:19]
```

1.2

a. On the IAS, what would the machine code instruction look like to store the contents of an accumulator to memory address 8?

STOR M[8]

b. On the IAS, what would the machine code instruction look like to add the contents of memory address 16 to the accumulator?

ADD M[16]

1.3
The IAS operates by repetitively performing an instruction cycle, which consists of two sub cycles: a fetch cycle and an execute cycle. On the IAS, describe in English the tasks accomplished during the fetch cycle and those accomplished during the execute Cycle.

Fetch: The address in the program counter is moved to the memory address register(MAR), Then this address is placed in the address bus. The control unit then uses the READ command on the control bus, which then copies the result to the MBR from the data bus. The MBR is then moved to the instruction register (IR)

Execute: Control circuitry interprets the opcode in the IR and executes the instruction by sending out the appropriate control signals to cause data to be moved or an operation to be performed by the ALU.

1.4
The instruction format of the IAS computer includes two instructions per word. Considering the design of the IAS computer, what benefits does this format provide?

Because the bits are split into 40 parts per instruction, and 20 parts per word. This allowed the computer to store the instructions compactly and have enough for the opcode and the address.

1.5
During the fetch cycle in Figure 1.7, why is an instruction always taken from the IBR?

Because the IBR holds temporary right-hand instructions. The IBR could also be updated without messing up the IR by constantly changing itself.

1.6
Under what conditions could the instruction format of the IAS computer be Inefficient?

Some things cannot be encoded in one string, an example might be a 32-bit immediate Operands.

1.7
The relative performance of the IBM 360 Model 75 is 50 times that of the 360 Model 30, yet the instruction cycle time is only 5 times as fast. How do you account for this discrepancy?

The discrepancy can be explained by memory systems and advances in I/O processing contribute to the performance ratio. A system is only as fast as its slowest link, so while some of the computer has sped up, another part of it hasn't been able to keep up.

1.8
While browsing at Billy Bob's computer store, you overhear a customer asking Billy Bob what the fastest computer in the store is that he can buy. Billy Bob replies, "The fastest computer in the store is a Pentium 4. It has a clock speed of up to 1.8 GHz. The second best is the Core 2 Duo, but it only has a clock speed of 1.2 GHz." Is Billy Bob correct? What would you say to help this customer?

No. Billy bob is not correct because clock speed is not the only factor. Even though the Pentium 4 chip may have a faster clock speed at 1.8 gigahertz, it does not mean the system will perform faster. There are a lot of factors such as system components like the number of cores to speed up activities that require two calculations at once.

1.9
For each of the following examples, label them as either general-purpose systems, embedded systems, or deeply embedded systems.

a. university mainframe server

Embedded system

b. smart TV

Embedded system

c. smart fridge sensors

Deeply embedded system

d. personal laptop

General-purpose system

e. hearing aid

Deeply embedded system

f. GPS Navigation Unit

Deeply embedded system

1.10
The Cortex-R is an ARM CPU specifically designed for real-time or safety-critical systems. For each example in the list below, explain why the Cortex-R may or may not be appropriate.

a. self-driving car

Yes, a self-driving car would need to be able to auto-correct itself constantly and quickly in real time.

b. environmental control system in an aircraft

Yes, as altitude changes quickly while one flies, they most likely will need a chip that can respond quickly to changing conditions in the atmosphere

c. blood gas analyzer

Yes. The blood gas analyzer wants to analyze things as quickly as possible, and the Cortex-R can do calculations to track changes in the blood or even save a patient's life.

d. automated telephone communication switchboard

Yes. It is a real time system that wants people to be connected as fast as possible to the internet.

e. electrocardiogram machine

Yes. The quicker you can tell a person's heart rate, the faster you can help save them if it dips.

f. programmable keyboard

No. It is most likely going to have the Cortex-M, as it is not needed for quick processing.

g. standard monitor for a desktop computer

No. The monitor does not need any programming, it only needs to display the desktop as fast as possible so probably a Cortex-M.

h. video game console

No. Game consoles do not need to load many things, and a Cortex-R chip simply does not have the power to load a game so probably a Cortex-A.

i. electronic proximity key finder

No. They most likely only need a Cortex-M processor for such a simple task.