5.1 Terminology

Short Questions
==========

State two binary operators.
State two binary operators which do **not** perform arithmetic calculations.
An **argument** is another name for a _____.
In the following command, identify four **operands,** one **binary function**, one **unary function**, one **binary operator,** as well as an example of **infix notation** and an example of **prefix notation.**

answer = sqrt(power(A,2) + power(B,3))

Rewrite this expression in **postfix** notation: A*(B+C) / D
What is another name for postfix notation?

Explain why an **infinite recursive loop** causes a **stack overflow.**

Pop and **Push** are standard operations for a _____.
A _____ can be either **linear** or **circular.** If it is _____, it is less likely to suffer an overflow error.
What causes a **stack underflow?**
What would be the consequence of using a **stack** to store print jobs in a server, rather than a **queue?**
Algorithms can be **iterative** or **recursive**. What do the two terms mean?

Long Question
==========

A **compiler** must **parse** (take apart) an expression into pieces. In the first stage, it checks to see whether the **brackets** match correctly. There are 3 types of brackets: ( ) [ ] { }. They can be **nested** inside one another, for example:

if (words[x] <> left(words[x+1] , 5) **then**

The first attempt by the programmer is simply to **count** the left and right brackets – there should be an equal number of each. In the example above, there are 2 (, 2) , 2 [, and 2] . But merely counting is not enough, because the following examples are incorrectly bracketed, although the number of brackets is correct:

) 4*x (, 3*(list[5) + 1] , ([( ]))

The programmer's second attempt uses a stack, with the following algorithm:

```
Scan the string one character at a time, from left to right
If a left bracket (  [  or  {  is encountered then
      Push it onto the stack
If a right bracket )  ]  or  }  is encountered then
      Pop the stack and compare the popped bracket (hopefully left)
        with the current right bracket – they should match
At the end, the stack should be empty.
There should be no underflows in the middle of the loop.
```

(a) Write the program to implement this algorithm.
(b) Explain why a **stack** is the appropriate structure here, rather than a **queue.**
(c) In the language C++, the curly braces { } are used to mark the **begin** and **end** of a program block, such as a loop, or the **then** part of an if..then.., or the beginning and end of a procedure. Name 3 pairs of structural identifiers in PURE which mark the begin and end of blocks of code (not brackets).
(d) Explain why the program above could not be easily modified to scan for correct matching of the **begin** and **end** identifiers identified in part (c).

These are all Stack and Queue algorithms.
Find all the errors and fix the methods.

==========

```
public void car(String data)
{  top = top + 1
   stack = data
}
```

==========

```
public int dad()
{  top = top - 1;
   return stack[top+1];
}
** State a type of error which could occur, and handle it.
```

==========

```
public void en(String data)
{  tail = tail + 1;
   list[tail] = data;
}
** State a type of error which could occur, and handle it.
```

==========

```
public String de()
{  head = head + 1;
   return list[head-1];
}
```

==========

```
public void inys()
{  stack = 0;  }
```

==========

```
public void inyq()
{  head = 0; }
```

==========

```
public void ec(String data)
{  tail = tail + 1;
   if (tail > listMax)
   {  tail = tail - 1; }
   list[tail] = data;
}
```

==========

** Write a procedure to DEQUEUE a value from a **circular** queue, with array size listMax..

Linked-List Algs – assume that HEAD points at the first node, but no pointer to the last node.
Assume each method is called starting with the HEAD node.

```
=============
public Node gt(String item)
{  Node temp = head;
   boolean found = false;

   while ((found == false) && (temp != null))
   {  if (temp.data.equals(item))
      {  found = true; }
      temp = temp.nextNode;
   }
   return temp;
}
==============

void ap(String item, Node start)
{  Node temp;
   if (start == null)
   {  temp = new Node();
      temp.data = item;
      temp.nextNode = null;
      start = temp;
   }
   else
   {  ap(item, start.nextNode); }
}
```

** Explain how the use of a TAIL pointer could make this more efficient.

```
==============

void kk(Node start)
{  Node temp ;
   while (start != null)
   {  temp = start;
      start = start.nextNode;
   }
}
** Explain how this changes the value of HEAD.
==============

void rr(Node head)
{  Node temp = null , make = null;

   while (head != null)
   {  temp = head;
      head = head.nextNode;
      temp.nextnode = make;
      make = temp;
   }
   head = make;
}
```

** Write a procedure which removes the HEAD node and re-attaches it at the end of the list.

Tree Algorithms – assume that a binary search tree exists, with ROOT pointing to the root node

```
public void traverse(Node start)
{  output(start.data);
   if (start.leftChild != null)
   {  traverse(start.leftChild); }
   else if (start.rightChild != null)
   {  traverse(start.rightChild; }
}

** This is supposed to print all the names in the tree.
   Will it print the names in alphabetical order?  If not, fix it.
===============

public void adding(String item)
{  Node p = root;
   while (p != null && !p.data.equals(item) )
   {  if (item.compareTo(p.data) > 0)
      {  if (p.rightChild == null)
            p.rightChild = new Node();
         else
           p = p.rightChild;
      }
      else if (item.compareTo(p.data) < 0)
      {  if (p.leftChild == null)
           p = new Node();
         else
           p = p.leftChild;
      }
   }
}
This should put some new data into the tree. How? Where?
Why won't it work correctly? Fix it.

===============
void startHere()
{  ccc(root,1);  }

void ccc(Node start, int count)
{
     if (start.leftChild == null && start.rightChild == null)
     {  output(count); }
     else
     {
        if (start.leftChild != null)
        {  ccc(start.leftChild,count + 1); }

        if (start.rightChild != null)
        {  ccc(start.rightChild,count + 1); }
     }
}
```

** Draw a tree where this procedure would print: 2 , 3 , 3

```
===============
```

Recursive Algorithm

```
void f(int num)
{  int x = 2;
   boolean found = false;

   while (!found && x < num)
   {
     if (num % x == 0)
     {  f(x);
        f(num / x);
        found = true;
     }
     else
     { x = x + 1; }
   }

   if (!found)
   {  System.out.println(num); }
}

** Trace the execution of f(42) and state the output.
```

Big O Efficiency – state the Big O efficiency of each algorithm

(a) Binary search
(b) Selection sort
(c) Quick sort
(d) Sequential search
(e) Calculating the average of a list of real numbers.

(f) For the following algorithm (removeDuplicates), calculate the number of **iterations of the innermost loop** if the LIST array contains the following six numbers at the beginning: { 1 , 5 , 3 , 5 , 3 , 1 }
(g) State the theoretical efficiency in Big-O notation under the **worst-case scenario –** when ALL the numbers are duplicates.

```
public void removeDuplicates()
{  int x , d , t ;
   x = 0;
   while (x < listSize)
   {  d = x+1;
      while (d < listSize)
      {  if (list[d] == list[x])
         {  for (t = d; t < listSize; t = t + 1)
            {  list[t] = list[t+1]; }
            listSize = listSize - 1;
         }
         else
         {  d = d + 1; }
      }
      x = x + 1;
   }
}
```

(h) Describe an O(n) algorithm for reversing the order of the elements in an array.

6.1 + 6.2 Chips

The ALU and CU are two parts of the _____.

The ALU performs _____ and _____ operations.

The _____ controls communications across the data bus.

The data bus contains 32 (or maybe 64) _____ circuits,

each of which can carry one _____ of data.

The CPU uses the data bus to fetch and store data in the _____.

But to speed up operations, the data bus does not connect directly between the CPU and the RAM.

The CPU uses the _____ for temporary storage, and **it** is connected to the RAM.

The data bus runs at a frequency of 100 _____ . If the data bus width is 32 bits, then it manages

a data-transfer-rate of _____ MegaBytes per second.

The RAM can contain both _____ and data.

The BIOS contains small utility programs, which are stored in _____, so they are available

immediately when the computer starts.

The CPU is constantly busy executing instructions. For each instruction, it must go through the cycle of

_____, _____, _____, _____.

This could require many machine cycles. If one instruction requires 20 machine cycles, and the CPU is

running at 1 GigaHerz, the CPU could execute _____ MIPS.

Sometimes the CPU must respond to _____ signals, to take care of a request

from a _____ device, such as the keyboard.

Every time a key is pressed on the keyboard, a signal is sent to the _____ .

It must stop whatever it is doing and handle the request. To keep track of it's current activities, it

stores the values of all registers in a temporary storage area called the system _____.

After it has finished "handling" the interrupt, it can _____ all the values back out of storage and

continue where it left off.

To handle the keyboard request, the CPU accepts the _____ code from the keyboard,

and places this code in the keyboard buffer. This buffer is in the form of a _____, as it must

function in a FIFO fashion. Otherwise, the order of the keystrokes would get mixed up.

6.3 + 6.4 Disk Storage

A **cylinder** is the set of circular tracks located above each other on separate **platters** in a hard-disk.
Each track is divided into smaller pieces called **sectors**. A **sector** is the smallest quantity of data which can be read or written on a hard-disk. Typically this 512 bytes of data.

- How many sectors does a 20 GigaByte hard-disk have (if each is 512 bytes)?
- How many sectors does a floppy diskette have?

The operating system must keep track of where files are stored on the disk drive. It must **allocate** free sectors when saving a file, or **access** the correct sectors when loading a file. The sectors must be numbered, and the OS keeps track of the status of each sector – whether it is **available** or **used.**

- If a 16-bit number is used to store the sector ID numbers, how many sectors can there be on the disk?
- If a disk has 4 billion sectors, how many bits are needed for each sector ID?

To make storage more efficient, and reduce the size of the sector ID numbers, sectors can be **blocked** together in **clusters**. In MS-DOS, sector IDs were 16-bit numbers. By blocking 16 sectors together, a single ID number could represent 8 KB of data.

- Using a block (cluster) size of 8 KB, how much data can be stored on a disk using 16-bit codes?

When a disk-drive gets full, the OS must use sectors in many different tracks, which may be far apart.
To load a single track, the read-write head may need to jump around between many different tracks.
One measure of disk-drive speed is the **maximum seek time** – the amount of time to move the read-write head from the inner-most track to the outer-most (the farthest distance). This could be something like 10 milliseconds. Once the head is at that track, it may still need to wait for the disk to spin around until the desired sector is found (**rotational delay).** Once the sector is found, it can be copied into RAM. If a large blocking factor is used, then many sectors can be transferred quickly, one after another. However, if the disk is **fragmented,** there may be another **seek** operation to find each new sector. This can slow things down considerably. A **defragmentation utility** can reorganize the files so that they are stored in **contiguous** sectors (next to each other).

- If a disk-drive spins at 5400 RPM, and has 1024 tracks, 4 read-write heads, with a total storage capacity of 20 GB, calculate the maximum possible data-transfer-rate on best circumstances.
- If the disk-controller has a 32-bit data-bus running at 133 MHz, can it carry transfer the data at this theoretical maximum rate?

Other questions:
- Can a **virus** attack cause files to become fragmented? Justify your answer.
- When a file is deleted, the **data** in the **sectors** is **not changed.** What does change?
- Explain why it's **not possible** to change **blocking size** on a hard-disk to improve efficiency.
- When a file is **compressed**, does the **storage structure** change, or does the **data** change?
- Explain why a compressed file might load more **slowly** than a normal file.
- Describe one other **file maintenance utility** other than defragmenter, compression, and virus scanner.

A **linker** can combine several **object modules** into a single **executable program.**
- What would **produce** the original object modules?

A **loader** is the part of the OS that copies a program into the memory and starts it running. The loader must input the file name, find the file on the hard-disk, load it into memory, and start the execution.
- If the loader is part of the OS, then what loads and executes the OS?

A **library manager** maintains a database of reusable modules, for example .dll files in Windows.
- What does "DLL" stand for, and how does Windows manage the library modules?

6.5 Communication

Port
The place on the back of the computer where you attach cables to connect to another deviece.

Different connections use different cables. A _____ connections has only one single data wire. This transmits one bit of data after the next.

A _____ interface uses 8 (or more) wires to send many bits of data simultaneously.

- Which type of interface is generally faster?
- State one disadvantage of the faster method.

Memory Mapped IO
The CPU treats this connection like a memory location, by assigning a memory address to it.
To send data to the other device, the CPU "stores" the values to this memory location.

DMA and Buffers
A disk-drive can read and write directly to the memory (RAM). This is called Direct Memory Access. Some disk-drives and many other devices are not allowed to use DMA. Instead, transferred data is

collected in a _____ until it is full, or until the CPU has time to process it, store it, or send it.

To speed things up, the system can use _____, where one buffer is being filled at the same time as the other as being emptied.

Interrupts and Polling
Some devices send an **interrupt signal** to tell the CPU they need attention. Other devices wait for the CPU to ask – this is called **polling.** Polling usually happens on a scheduled basis (e.g. 10 times per second).

- Why are **interrupts** better for a keyboard rather than **polling?**
- Why is **polling** a suitable system for sensors (e.g. temperature sensor) rather than **interrupts?**

Controller
A separate **interface card** which sends **control signals** to the device to start or stop processes, or to monitor communication. A disk-drive controller controls the **buffering** and/or **DMA** activities.
- What does IDE mean? How can an IDE disk-drive work without needing a controller card?
- Explain why allowing the disk-drive to use DMA speeds up the entire system./
- State one **control signal** that a printer might send to the CPU.
- State one **control signal** that might be sent from the CPU to a printer.
- State two different methods allowing a printer to differentiate between **control signals** and **data signals.**

Analog/Digital
Which of the following involve DAC or ADC? Which do not?

- Storing data on a disk drive
- Playing a phonograph record on a stereo
- Using a phone-card in a public telephone
- Printing from a PC to a laser printer
- Photography with a digital camera
- Playing CD music on a PC speaker
- Measuring temperatures and storing them in a data file
- Controlling an industrial robot
- Audio sampling (what's that?)
- Scanning and OCR

Which is more "realistic" – analog or digital data?
Which can be copied more perfectly – analog or digital data?
Which requires more storage space – analog or digital data?
Which can be transmitted more rapidly across the Internet – analog or digital data?

7 System Life Cycle

Why is it a life **cycle?** What about it is cyclic?

Match each **stage** with a description:

Stage	Description
Analysis	Deciding what data needs to be stored and what features need to be implemented
Design	Installing and debugging the finished system
Implementation	Investigating the existing system, talking to the intended users
Operation	Writing programs and/or building hardware
Maintenance	Making minor changes to adjust to new requirements, fixing problems

Explain the importance of collecting sample data during the analysis stage, and outline an example of something which could "go wrong" if this is not done.
What is a **requirements specification?** In which stage would this be produced?
What is a **feasibility study?** Which should be done first – **feasibility study** or **requirements specification?**
Explain the difference between **module level testing** and **system testing.**
Explain the difference between **test data** and **test-cases.**
Describe a social implication of poor testing.
Describe one positive social implication and one negative social implication of converting a factory to depend on robots rather than human labor.

Long Question
==========
A **video rental** shop wants to computerize their business. They want to keep track of customer names and phone numbers, so they can call a customer who keeps a video too long. They want to keep track of which videos are loaned out, so if someone calls they can easily tell whether the video is available. They want the system to calculate the cost of a rental when the video is returned. They have a variety of price categories with different rules for each price category. For example:

> **Brand New Videos :** first day 3 Euros, each day after that 2 Euros
> **Very Old Videos :** 2 Euros for the 2 days (minimum charge), 1 Euro for every 2 days after that

There are also other categories with other rules.

(a) Describe 3 types of sample data which should be collected.
(b) Explain why the **intended user** should be involved in the **selection of hardware**.
(c) Explain why the intended user should **not** be involved in the choice of a programming language.
(d) Explain why the **programmer** would **not** be involved during the analysis stage.
(e) Explain two reasons why the introduction of the computer system might affect employment in the shop.
(f) Describe the need for an alternative input device other than keyboard and mouse.
(g) Describe the need for an alternative output device other than a monitor.
(h) Which is probably more appropriate for this system – **on-line** or **batch** processing?
(i) Once the system is up and running, describe the worse conceivable consequence of computer failure.
(j) Describe an appropriate **backup strategy**, including **scheduling** the backups.