

APOLLO

Computer Science HL Answers

First assessment 2027

Whole questions arranged by syllabus topic.

Original question and scheme crops preserve source diagrams.

APOLLO

A1 Computer fundamentals

1 questions

1. (a) *Award [2] max*

To store intermediate arithmetic and logic results/hold operand data;
 ...to reduce the number of memory accesses (by holding intermediate results);

Acts as a temporary storage location;
 ...which stores the values and results from arithmetic and logical operations;

Holds data or the results of an operation;
 ...during the processing cycles;

(b) *Award [1] max*

10101100

(c) *Award [2] max*

Award [1] for each correct column (Difference, Borrower)

X	Y	Difference	Borrower
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(d) *Award [2] max*

Tracks resource usage;
 Monitors and logs the consumption of resources like CPU, memory, and disk by various processes and users;

Assesses system performance;
 Collects data to identify system bottlenecks, under-utilized resources, or areas for potential optimization;

Assigns costs based on usage;
 In shared computing environments, users may be billed based on the resources they consume;

Facilitates auditing;
 Helps in detecting abnormal behaviors or potential security breaches;

(e) *Award [2] max*

Award [1] per point up to [2] max

Offers a range of pre-built components and libraries to accelerate development;
 Developers can concentrate on coding rather than managing infrastructure;
 Cloud hosted to allow multiple team members to collaborate on application development;
 PaaS often comes with built-in tools for development, testing, integration, and deployment;
 PaaS providers often have strong security protocols to protect the application and data;
 PaaS solutions are designed to be scalable, allowing for easy expansion;

(f) *Award [4] max*

Award [1] per point up to [4] max

Pre-translation: Compiled code runs faster because it has been pre-translated into machine code, while interpreted code is slower due to the on-the-fly interpretation;

Runtime overhead: Compiled code is pre-compiled, so there's no runtime overhead associated with translating source code to machine code, whereas an interpreter must parse, interpret, and translate each line or block of code in real time;

Code optimization: Compilers can take advantage of the architecture's specific capabilities (e.g., optimizing for loops, tasks), whereas interpreter does not have the full view of the program;

Resource utilization: Compilers are more efficient with CPU and memory resources during execution, whereas interpreters consume more CPU cycles and memory during execution (due to the overhead of interpreting code line by line);

Cache utilization: Compiled code has better cache utilization due to the compilation optimizations, whereas the interpreter itself adds an extra layer of instructions that need to be processed, leading to cache misses and slower execution.

APOLLO

A2 Networks

2 questions

2. (a) *Award [1] max*

Router/Modem/Gateway

(b) *Award [2] max*

Award [1] for point, [1] for expansion

The network interface layer is responsible for the physical transmission of data;
...including the layout of bits into packets, frames, or other formats;

The network interface layer manages interactions between devices/hardware;
...and the network medium itself, such as Ethernet cable or Wi-Fi;

The network interface layer handles error notification;
...Identifying and reporting any issues or faults in the data transmission process (e.g. lost packets or a disconnected network cable);

The network interface layer determines the network topology (e.g., star, bus, ring, hybrid);
...which impacts network operations (e.g., scalability, fault-tolerance, and data transmission efficiency).

The network interface layer manages physical addressing;
...including assignment and identification of devices (e.g., MAC addresses).

(c) *Award [3] max*

Award [1] per point

Transmission speed/higher bandwidth;
Lower latency compared to twisted-pair cables;
Transmits signals over greater distances than twisted-pair cables;
Immune to electromagnetic interference (EMI) and radio-frequency interference (RFI);
Better security against unauthorized access or data tapping;
Better scalability as data transmission needs grow;
Generally more durable than twisted-pair cables;
Longer lifespan and lower maintenance requirements;

– 8 –

(d) *Award [4] max**Award [1] per point*

MAC address learning: A switch learns the device's MAC address (from the incoming packet's source address)/a switch maintains a MAC address table;
...and associates the MAC address with the physical port through which the packet entered;

Packet examination: destination MAC address is shown in the frame/packet;
...and compared with the switch's MAC address table to find the port;

Selective forwarding: The switch determines the appropriate port to forward the packet/frame based on its destination MAC address;
If the destination MAC address is unknown the switch broadcasts the packet (frame) to all ports;

Possible candidate response worth full marks

The switch saves the arrival port/sender mac address extracted from packet;
The destination MAC address is extracted from the packet;
And the port is looked up in a table based on the MAC address;
The switch then plots the path to the destination port.

Note: either frame or packet is acceptable for candidates' answers.(e) *Award [4] max**Award [1] for each point from the same section*

NAT translates private IP addresses of devices within a local network to a public IP address;
External entities see only the public IP address of the network;
NAT makes it more difficult for attackers to directly reach internal systems;
The extra layer of obscurity enhances the security of the network;

NAT and firewalls often work together;
Firewall inspects and filters incoming and outgoing traffic based on security rules;
NAT allows network administrators to control how traffic is routed;
NAT can be configured to forward specific types of traffic to specific internal devices;

Not only does NAT add an extra layer of security;
NAT enables multiple devices on a local network to share a single public IP address.
Useful given the limited availability of IPv4 addresses;
Enables the network to have more devices than they have IPv4 IP addresses;

5. (a) (i) *Award [1] per line up to [2] max*

Adware
Backdoor
Bots/botnet
Crypto-jacking
Drive-by download
Exploit kit
Fileless malware
Keylogger
Polymorphic malware
Spyware
Trojan horse
Virus
Worm
Zero-day exploit

Accept other examples of legitimate malware

(ii) *Award [2] max*

Award [1] for identifying issue and [1] for expansion

Reputational damage;
Clients may lose confidence in a company that has a breach of security;
For example, parents may withdraw their students from a school;

Financial fraud for patients;
Patients' credit card numbers/banking data/passwords may result in funds being removed;

Theft of data;
Patients' data is used unethically (e.g., insurance company refusing to insure patients with potential pre-existing medical conditions);

Identity theft;
Having your identity stolen can cause psychological damage; e.g., a person may steal a person's online profile and insult others, causing emotional distress;

Data loss causing disruption to hospital services;
Phishing attacks are often designed to steal sensitive data; e.g., doctors lose access to patients' records and can't treat them;

Legal consequences;
Company may be sued for leaking sensitive information by patients/companies/government/ombudsman and lose their license to practice medicine;

Additional malware installed;
Creates vulnerability for the additional malware to cause further disruption.

– 15 –

(b) *Award [4] max*

Award [1] for identifying strategy and [1] for expansion

Implement **device authentication** protocols;
Registering devices, require complex passwords, two-factor authentication, certificate authentication, or biometric data to access the network;

Implement **access control** protocols;
Restricting the user access based on their role;

Use **device management software**/mobile device management (MDM);
Allows IT to manage, monitor, and secure employees' mobile devices;

Require **security updates**/patches be installed;
Devices without the updates/patches will trigger an alert to IT admin;

Install **anti-malware software** on all devices;
Ensure it is kept up to date/push updates to the users' devices;

Remote wipe software;
Install software to delete company data on lost devices remotely;
End-user technical agreement agreeing to device use;
Employee signs as part of their contract;

User training on cybersecurity;
To reduce the risk of cybersecurity breaches;

Mark as [2] + [2]

(c) *Award [4] max*

Award [1] for advantage, and [1] for expansion

Cost efficiency: Reduces initial costs on purchasing hardware;
Maintenance, repair, and replacement costs shifts to employ;

Familiarity: Users are more familiar with their own devices;
Leading to increased productivity/satisfaction and reduced training needs;

Personal choice: Employees are empowered to choose their own devices;
Gives them autonomy/sense of ownership/enables remote working;

Mark as [2] + [2]

(d) **Award [12] max**

Candidates should discuss prevention and response, expressing their opinions on the effectiveness of the strategies. The best responses will demonstrate research and cite real-world related events and/or experts in the field.

Prevention

Implementing robust security protocols: TaylorMade Healthcare can establish stringent security protocols, including firewalls, encryption, email filtering, and regular software updates. These measures can prevent many common attacks, but candidates should be aware that security measures quickly become outdated.

Access control and authentication: Implement strict access control policies, ensuring employees have access only to the data necessary for their role. Use strong authentication methods like two-factor authentication.

User awareness training: Educating staff about potential threats, such as phishing scams and ransomware attacks, can prevent many attacks. Candidates should be aware that human error remains a significant challenge. TaylorMade Healthcare should have clearly defined procedures and protocols for new staff and departing staff. No matter how much training is provided, it is virtually impossible to eliminate all human errors that might lead to a security breach.

Network segmentation and access control: This strategy can limit the damage if a breach does occur by preventing hackers from accessing the entire network. Candidates should be aware that this strategy may not prevent the initial breach and can be complex to implement and manage.

Update and patch management: Ensure all software, including operating systems and applications, are kept up to date with the latest security patches.

Regular security audits and assessments: Conduct regular security audits and risk assessments to identify and address vulnerabilities.

Monitoring and logging: Continuously monitor network activity and maintain logs to detect and investigate suspicious activities.

Physical security: Enhance physical security measures to prevent unauthorized access to servers and other critical IT infrastructure.

Vendor risk management: Assess and manage the cybersecurity risks associated with third-party vendors and service providers.

Cybersecurity insurance: Consider purchasing cybersecurity insurance to mitigate financial risks associated with data breaches.

Collaboration with law enforcement and cybersecurity agencies: Work with law enforcement and cybersecurity agencies for threat intelligence sharing and guidance on emerging threats and best practices.

Response

Data backup and disaster recovery plans to limit the damage caused by an attack and ensure that services can quickly resume after a breach. Candidates should be aware that implementing these measures requires significant resources and that recovering from a breach can still be a lengthy and expensive process.

Third-party services: Cybersecurity consultants and managed security service providers can help respond to breaches quickly and effectively. Candidates may consider whether TaylorMade Healthcare can afford this approach because these services can be expensive.

Failover systems: These systems ensure that critical services remain available even if the primary system fails. However, failover systems require significant investment and ongoing maintenance.

Conclusion

A reasoned conclusion that expresses an opinion supported by evidence.

Marks	Level descriptor
No marks	- No knowledge or understanding of the relevant issues and concepts. - No use of appropriate terminology.
Basic 1–3 marks	- Minimal knowledge and understanding of the relevant issues or concepts. - Minimal use of appropriate terminology. - The answer may be little more than a list. - No reference is made to the information in the case study or independent research.
Adequate 4–6 marks	- A descriptive response with limited knowledge and/or understanding of the relevant issues or concepts. - A limited use of appropriate terminology. - There is limited evidence of analysis. - There is evidence that limited research has been undertaken.
Competent 7–9 marks	- A response with knowledge and understanding of the related issues and/or concepts. - A response that uses terminology appropriately in places. - There is some evidence of analysis. - There is evidence that research has been undertaken.
Proficient 10–12 marks	- A response with a detailed knowledge and clear understanding of computer science. - A response that uses terminology appropriately throughout. - There is competent and balanced analysis. - Conclusions are drawn that are linked to the analysis. - There is clear evidence that extensive research has been undertaken.

APOLLO

A3 Databases

1 questions

3. (a) Award [3] max*Award [1] for SELECT DISTINCT or GROUP BY**Award [1] for using LIKE**Award [1] for using the wildcard (any wildcard accepted) and therapy*

```
SELECT DISTINCT PatientName FROM VISITS
WHERE Treatment LIKE '%therapy%';
```

Alternative solution using group by:

```
SELECT PatientName FROM VISITS
WHERE Treatment LIKE '%therapy%'
GROUP BY PatientName
```

Alternative solution using a subquery and group by:

```
SELECT PatientName
FROM (
    SELECT PatientName
    FROM VISITS
    WHERE Treatment LIKE '%therapy%'
) AS subquery
GROUP BY PatientName;
```

(b) Award [5] max*Award [1] for the PATIENT table with correct Primary Key**Award [1] for the DOCTOR table with correct Primary Key**Award [1] for PatID foreign key in the VISIT table**Award [1] for DocID foreign key in the VISIT table**Award [1] for VisitID or PatID/DocID/VisitDate composite key in the VISIT table*

```
PATIENT(PatID, PatientName, PatientDOB)
DOCTOR(DocID, Doctor)
VISIT(VisitID, PatID, DocID, VistDate, Treatment)
```

– 10 –

(c) *Award [3] max*

Award [1] per point

Atomicity assures that a transaction is treated as a single, indivisible unit;

A transaction must either be fully completed on all nodes or not at all;

Transaction failure on one node triggers a rollback of the entire transaction across all nodes;

A commit will make sure that the updates which happened on one node because of the transaction are persisted in all nodes.

(d) *Award [3] max*

Award [1] per point

Spatial DBs are designed to store geographic data such as addresses;

Spatial DBs can be queried to identify clusters of virus patients;

Spatial DBs can create visual representations (maps) to see patterns;

Spatial DBs can support advanced spatial analysis, calculating distances between patient addresses and hospitals or identifying the cause e.g. nearby research lab;

Enables resource allocation (medical teams can be dispatched to the region);

Stored address data enables location-based contact tracing/highlights the area to quarantine/notify neighbours of the outbreak;

APOLLO

A4 Machine learning

1 questions

4. (a) *Award [1] max*

Input layer;
Output layer;
Convolution layer/conv layer;
Pooling layer/subsampling layer/down-sampling layer;
Fully-connected layer/FC layer;
Normalization layer;
Activation layer.

(b) *Award [1] max*

To improve model accuracy/ensure the model is not trained on poor quality data;
To ensure inaccuracies/errors/inconsistencies in the data doesn't lead to false assumptions;
To removing duplicate data/irrelevant data points;
Reduce the training time/computational resources required;
To handle missing values by using imputation (e.g. inserting mean) or removing;
Ensures the CNN doesn't make predictions on incomplete data/have gaps in its knowledge;
To provide uniformity of data/ensure all images are the same size and format;
To ensure the model isn't giving undue importance to some features over others;
To balance the dataset (undersampling or oversampling);
To ensure the dataset doesn't have an uneven distribution of classes/become biased towards the majority class;
To avoid overfitting;
To remove outliers or extreme values in the training data that can cause overfitting.

(c) *Award [2] max*

Award [1] per point from one section

Deep learning involves many matrix operations (multiplications and convolutions);
GPUs have many smaller cores designed explicitly for parallel operations;

Large datasets increase the computational demand;
Training on a CPU only would be too time-consuming/GPUs speed up the training time;

Converges by minimizing the cost function (e.g., batch gradient descent) lending itself to simultaneous operations;
GPUs provide parallel processing, leading to faster convergence;

Accept any reasonable machine learning process that requires parallelization

– 12 –

(d) *Award [3] max**Award [1] per point*

Calculate the F1 score as the harmonic mean of precision and recall;
 A low F1 score implies inadequate performance of the CNN model;
 This may require training adjustments or data rebalancing;
 It is an iterative process of adjusting and testing to improve the F1 score.

Alternative answer

Produce a confusion matrix (TP, TN, FP, FN);
 Calculate precision using $TP / (TP + FP)$;
 Calculate recall/sensitivity using $TP / (TP + FN)$;
 Calculate F1 score = $2 * (Precision * Recall) / (Precision + Recall)$.

Note to examiners:

TP is true positive (correctly identified cases of pneumonia);
TN is true negative (correctly identified cases without pneumonia);
FP is false positive (wrongly identified cases of having pneumonia);
FN is false negative (cases of pneumonia that were not identified).

(e) *Award [3] max**Award [1] for identifying an **issue****Award [1] for expanding on the issue/stating the **context****Award [1] for a further **evidence** highlighting the choice of RL over SL*

RL works well in **dynamic environments** where layouts change/hospitals are dynamic environments (beds move, machinery is moved);
 RL involves the cleaner (agent) interacting with the environment (hospital), receiving continuous feedback based on its actions (movement decisions);
 RL allows the robot cleaner to update its learned policy through rewards and punishments;
 SL is trained in advance on a static dataset so would struggle to adapt to untrained layouts/changing environments;
 SL requires accurate labelled datasets, which is impractical for every possible situation or condition in a hospital;

Alternative answer:

RL allows for **real-time autonomous learning** (without interventions);
 The robot (agent) uses RL to interact with the environment making decisions;
 The feedback loop in RL (perform action, get reward, update policy) ensures the robot continuously improves its strategy based on real-time outcomes;
 SL is trained before use and does not learn in real-time;
 Changes in the environment would require external intervention (retraining);
 A dataset with all possible layouts and actions is implausible;

Alternative answer:

RL is goal driven/has an **ultimate goal** of cleanliness;
 RL is designed to allow the cleaner to discover an optimal sequence of actions using trial-and-error;
 RL, unlike SL, can handle the situation where there are delayed rewards;
 SL is not designed to find the best sequence of actions;
 SL is designed to predict outcomes based on provided input;
 SL is suited to classification and regression tasks, not optimization tasks.

(f) *Award [5] max*

The following points might be considered:

- Definition of online harassment - any form of bullying or intimidation that takes place through digital mediums
- Consideration of the context because much depends on the comments
- The sensitivity shown by hospital staff – only online harassment when they harm, threaten, or are offensive. They may only be asking for retractions in extreme circumstances
- The method of contacting public members (online, email, phone)
- How the hospital obtained members contact details (right to privacy)
- Freedom of speech and protection from unsubstantiated comments
- Potential impact on the public's trust in the hospital
- Alternative suggestions to the current practice to alleviate ethical concerns
- Legal implications such as GDPR
- Impacts of online harassment can be severe, leading to emotional distress, anxiety, depression, and even self-harm or suicide in extreme cases
- Is it ethical to use machine learning in this situation
- Consideration of the training data used for sentiment analysis
- The accuracy and potential bias (e.g. ethnicity, socio-economic status, gender) of the model

Marks	Level descriptor
No marks	No knowledge or understanding of the relevant issues and concepts.
1–2 marks	Limited knowledge and understanding of harassment. The answer may be little more than a list of points. (General)
3–4 marks	Detailed knowledge that links harassment to the scenario; Fairly balanced analysis. (In context)
5 marks	An appropriate conclusion, linked to the analysis.

Section B

APOLLO

B1 Computational thinking

4 questions

1. (a) *Award [1 max]*

because it is in diagrammatic form;
 it is program agnostic;
 no complicated code / syntax to follow;

(b) *Award [3]*

correct values of i;
correct values of j;
correct output;

i	i <= N ?	j	j <= N ?	output
1	TRUE			1
		2	TRUE	2
		3	TRUE	3
		4	FALSE	
2	TRUE			2
		3	TRUE	3
		4	FALSE	
3	TRUE	4	FALSE	3
4	FALSE			

(c) *Award [1]. Note: the algorithm shows a nested loop.*

$O(n^2)$;

– 6 –

(d) Award **[5 max]** as follows

opening *employees* file for reading / default mode;
open *surnames* file for writing;
looping through *employees* file;
use of a counter or equivalent;
writing surnames only to *surnames* file;
closing both files;

*Note: Award **[1]** for first 2 marking points if files are opened without indicating access modes.*

Algorithm is:

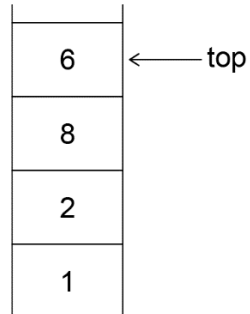
```
open employees file for reading / in default mode
open surnames file for writing
set counter to 0
repeat until end of employees file
    read next line from employees file
    if counter%3 = 0 *
        write line to surnames file
    increment counter
close both files
```

* there are alternatives ways of using counter

– 7 –

(e) *Award [2].*

correct elements in stack **[1]**;
correct indication of top **[1]**;



(f) *Award [3] as follows:*

Award [1] for each correct line (or equivalent).

repeat until isEmpty is true / while isEmpty is false;

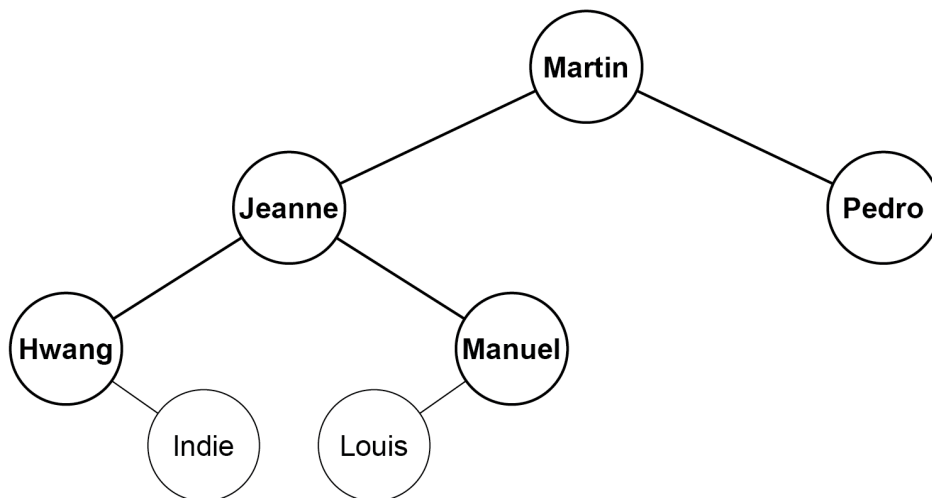
pop an element and save its value/save in a variable/push it onto a new stack (stack2)

print the last value saved/print the variable/pop stack2 and print value

Note: Answer must demonstrate an algorithmic approach.

5. (a) **Award [2 max]**

award [1] for each correct entry;

(b) **Award [3 max]**

The insert method is initially passed the root of the tree as a parameter (where a comparison is made);

Then either the left or right subtree is selected (depending upon the result of the comparison);

The method is called again, using the root of this subtree as the parameter;

This is repeated until the insertion point is found (base case) / subtree is null;

(c) (i) **Award [1 max]**

21;

(ii) **Award [4 max]**

Award [1] for including a recursive call that will lead to the base case;

Award [1] for having a base case;

Award [1] for return + int;

Award [1] for a completely correct algorithm;

```

public int recur(int x)
{
    if (x != 0)
        return x + recur(x-1);
    else
        return x;
}
  
```

(iii) **Award [3 max]**

data is stored in the memory / on a stack every time a recursive call is made;

if the value of x is very large ...;

... the stack may overflow / system may run out of memory;

7. (a) *Award [1 max]*

because it is in diagrammatic form;
 it is program agnostic;
 no complicated code / syntax to follow;

(b) *Award [3]*

correct values of i;
correct values j;
correct output;

i	i < = N ?	j	j < = N ?	output
1	TRUE			1
		2	TRUE	2
		3	TRUE	3
		4	FALSE	
2	TRUE			2
		3	TRUE	3
		4	FALSE	
3	TRUE	4	FALSE	3
4	FALSE			

(c) *Award [1]. Note: the algorithm shows a nested loop.*

$O(n^2)$;

– 15 –

(d) Award **[5 max]** as follows

opening *employees* file for reading / default mode;
open *surnames* file for writing;
looping through *employees* file;
use of a counter or equivalent;
writing surnames only to *surnames* file;
closing both files;

*Note: Award **[1]** for first 2 marking points if files are opened without indicating access modes*

Algorithm is:

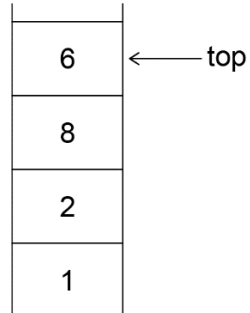
```
open employees file for reading / in default mode
open surnames file for writing
set counter to 0
repeat until end of employees file
    read next line from employees file
    if counter%3 = 0 *
        write line to surnames file
    increment counter
close both files
```

* there are alternatives ways of using counter

– 16 –

(e) *Award [2].*

correct elements in stack **[1]**;
correct indication of top **[1]**;



(f) *Award [3] as follows:*

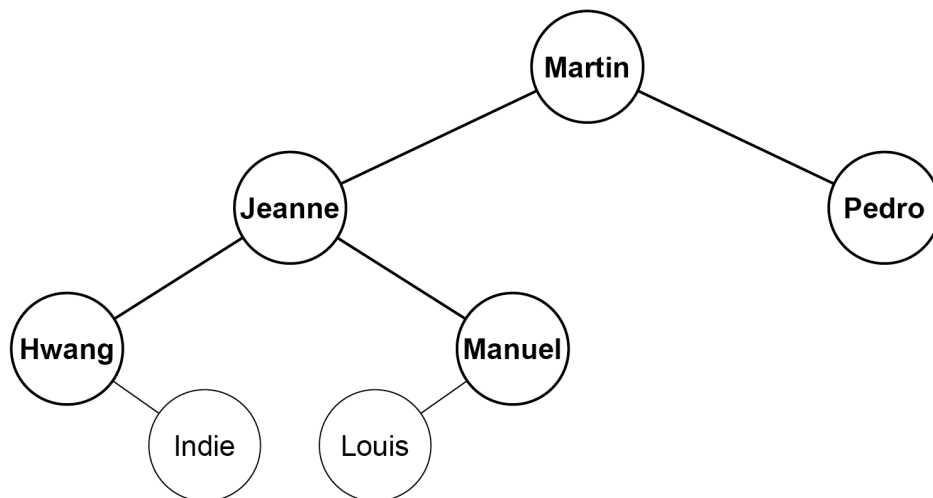
Award [1] for each correct line (or equivalent).

```
repeat until isEmpty is true / while isEmpty is false;  
    pop an element and save its value/save in a variable/push it onto a new stack (stack2);  
    print the last value saved/print the variable/pop stack2 and print value;
```

Note: Answer must demonstrate an algorithmic approach.

11. (a) **Award [2 max]**

award [1] for each correct entry;

(b) **Award [3 max]**

The insert method is initially passed the root of the tree as a parameter (where a comparison is made);

Then either the left or right subtree is selected (depending upon the result of the comparison);

The method is called again, using the root of this subtree as the parameter;

This is repeated until the insertion point is found (base case) / subtree is null;

(c) (i) **Award [1 max]**

21;

(ii) **Award [4 max]**

Award [1] for including a recursive call that will lead to the base case;

Award [1] for having a base case;

Award [1] for return;

Award [1] for a completely correct algorithm;

```

def recur(x):
    if x != 0:
        return x + recur(x-1)
    else:
        return x
  
```

(iii) **Award [3 max]**

data is stored each in the memory / on a stack every time a recursive call is made;

if the value of x is very large ...;

... the stack might overflow / system might run out of memory;

APOLLO

B2 Programming

2 questions

2. (a) Award [3 max]

It is easier to work with one single structure than several individual ones;
Individual elements that have the same value can be quickly accessed via their indices;
Iteration through arrays is more efficient / Methods associated with arrays can make operations more efficient;

(b) Award [4 max]

repeat for each entry/index in the `Name` array
 identify the position of the space in each name
 save the substring to the left as `s1` and the substring to the right as `s2`
 save the new entry as `s2+space+s1`

(c) Award [1]

```
MAT[1][2];
```

(d) Award [1] each;

The value of `a` is: 6
The value of `b` is: 17

– 9 –

(e) *Award [5 max] as follows:**Award [1] for initial variables correctly set;**Award [1] for both loops;**Award [1] for checking seat for False;**Award [1] for updating row value;**Award [1] for resetting tempTotal;**Award [1] for printing the correct row;*

```

public void least(boolean[][] seats)
{
    int falseTotal = 0;
    int leastRow = 0;

    for (int x = 0; x < 40; x++)
    {
        int tempTotal = 0;
        for (int y = 0; y < 30; y++)
        {
            if (seats[x][y] == false)           // if(!seats[x][y] )
                tempTotal = tempTotal + 1;
        }
        if (tempTotal > falseTotal)
        {
            leastRow = x;
            falseTotal = tempTotal;
        }
    }
    System.out.println ("Row with least seats sold is " + leastRow);
}

```

8. (a) Award [3 max]

It is easier to work with one single structure than several individual ones;
 Individual elements that have the same value can be quickly accessed via their indices;
 Iteration through arrays is more efficient / Methods associated with arrays can make operations more efficient;

(b) Award [4 max] as follows:

repeat for each entry/index in the `Name` list
 identify the position of the space in each name
 save the substring to the left as `s1` and the substring to the right as `s2`
 save the new entry as `s2+space+s1`

repeat for each entry/index in the `Name` list
 use the `split()` function
 to separate the 2 strings into 2 list values (`s1` and `s2`)
 save the new entry as `s2+space+s1`

(c) Award [1]

```
MAT[1][2];
```

(d) Award [1] for each:

The value of `a` is: 6
 The value of `b` is: 17

(e) Award [5 max] as follows:

Award [1] for initial variables correctly set;
Award [1] for both loops;
Award [1] for checking seat for `False`;
Award [1] for updating row value;
Award [1] for resetting `temp_total`;
Award [1] for printing the correct row;

```
def least(seats):
    false_total = 0
    least_row = 0

    for x in range(40):
        temp_total = 0
        for y in range(30):
            if seats[x][y] == False:
                temp_total = temp_total + 1
        if temp_total > false_total:
            least_row = x
            false_total = temp_total

    print("Row with least seats sold is ", least_row)
```

APOLLO

B3 Object-oriented programming

2 questions

3. (a) (i) Award [3 max]

The `(Passenger)` constructor is called;

A new object is created / memory is assigned for a new object ...;

... identified by the variable `p1`;

The `passengerName`, `email` and `ticketPrice` variables / all of the variables are initialized;

(ii) Award [2 max]

provides greater security for the values stored in these variables/for the code;

restricts access to such variables from outside of the class /only allows access through public / getter, setter methods;

(helps to) prevents accidental/unintended changes to the variables (from outside of the class);

(b) Award [2 max];

composition;

The variable `ticketPrice` in the `Passenger` class depends upon the `Flight` class;

(c) Award [4 max]

Award [1] for calculating the % seats sold;

Award [1] for all 3 conditions correct;

Award [1] for if ... else structure correct;

Award [1] for return;

```
public double updatedTicketPrice()
{
    double price;
    double percentSold = (seatsBooked*100)/maxPassengers;
    if (percentSold >= 10 && percentSold <= 50)
        price = initialPrice*1.5;
    else
        if (percentSold > 50)
            price = initialPrice*3;
        else
            price = initialPrice;
    return price;
}
```

– 11 –

(d) **Award [4 max]***Award [1] for calling* `updatedTicketPrice()`;*Award [1] for creating new* `Passenger` **object**;*Award [1] for updating* `passengers` **array**;*Award [1] for incrementing* `seatsBooked`;

```

public void addPassenger(String name, String email)
{
    double s = updatedTicketPrice();
    Passenger p = new Passenger(name, email, s);
    passengers[seatsBooked] = p;
    seatsBooked = seatsBooked + 1;
}

```

(e) **Award [5 max]**the `name` field of each `Passenger` object is used in the sorting process;

the array is divided into two parts: a sorted subset and an unsorted subset;

the smallest element is found in the unsorted part of the array and compared to the first (leftmost) element in the unsorted set;

if smaller, the two objects are swapped (and the first element in the unsorted set joins the sorted set);

the two steps above are repeated until the whole array is sorted / there are no more elements in the unsorted set;

9. (a) (i) Award [3 max]

The `(Passenger)` constructor is called;

A new object is created / Memory is assigned for a new object ...;

... identified by the variable `p1`;

The `passenger_name`, `email` and `ticket_price` variables / all of the variables are initialized;

(ii) Award [2 max]

provides greater security for the values stored in these variables/for the code;

restricts access to such variables from outside of the class /only allows access through public / getter, setter methods;

(helps to) prevents accidental/unintended changes to the variables (from outside of the class);

(b) Award [2 max]

composition;

The variable `ticket_price` in the `Passenger` class depends upon the `Flight` class;

(c) Award [4 max]

Award [1] for calculating the % seats sold;

(allow `len(self.passengers)` for `self.seats_booked`)

Award [1] for all 3 conditions correct;

Award [1] for if ... else structure correct;

Award [1] for return;

```
def updated_ticket_price(self):
    percent_sold = self.seats_booked * 100 / self.max_passengers
    if percent_sold >= 10 and percent_sold <= 50.0:
        price = self.initial_price * 1.5
    elif percent_sold > 50.0:
        price = self.initial_price * 3
    else:
        price = self.initial_price
    return price
```

(d) **Award [4 max]**

Award [1] for calling `updated_ticket_price()`;

Award [1] for creating new `Passenger` **object**;

Award [1] for updating `passengers` **list**;

Award [1] for incrementing `seats_booked`;

```
def add_passenger(self, name, email):  
    s = self.updated_ticket_price()  
    p = Passenger(name, email, s)  
    self.passengers.append(p)  
    self.seats_booked = self.seats_booked + 1
```

(e) **Award [5 max]**

the `name` field of each `Passenger` object is used in the sorting process;

the list is divided into two parts: a sorted subset and an unsorted subset;

the smallest element is found in the unsorted part of the list and compared to the first (leftmost) element in the unsorted set;

if smaller, the 2 objects are swapped (and the first element in the unsorted set joins the sorted set);

the two steps above are repeated until the whole list is sorted / there are no more elements in the unsorted set;

APOLLO

B4 Abstract data types-HL only

4 questions

4. (a) Award [3 max]

The *HashMap* is set up using pairs of string values;
representing the airport code and airport name / diagram to show this;
when the airport code is used as a parameter for the *HashMap*, the matching/corresponding
value of the airport name will be returned;

(b) Award [2 max]

the arrays are combined to form a single set;
the set will automatically eliminate any duplicates / will only hold unique values;

(c) (i) Award [1 max]

false;

(ii) Award [1 max]

[2,6];

6. (a) Award [5 max]*Award [1] for making use of a temporary node;**Award [1] for Boolean;**Award [1] for list traversal;**Award [1] for ending traversal if found;**Award [1] for correct comparison;**Award [1] for returning the correct values (in both cases);*

```

public void findPlayer(String name)
{
    Node temp = head;
    boolean found = false;
    while(temp != null)
    {
        if (temp.name.compareTo(name) == 0) // allow = name
        {
            System.out.println(temp.score);
            found = true;
            break;
        }
        else
            temp = temp.next;
    }
    if (found == false)
        System.out.println("Not Found");
}

```

(b) Award [5 max]

The algorithm will:

- make a temporary node equal to `head`;
- make use of a “previous” node;
- traverse the linked list (using the temporary node) comparing the score field of each node to the score field of the new node;
- until the insertion point is found;
- if insertion is at the beginning, link the new node to the head and make the head equal to the new node;
- if insertion is at the end, link the previous node to the new node;
- copy the *next* pointer of the node previous to the insertion point to the *next* pointer of the new node;
- link the previous node to the new node;

Note: Allow the use of diagrams to illustrate any of the above points.

Option B — Python

10. (a) Award [3 max]

The dictionary is set up using pairs of string values;
representing the airport code and airport name / diagram to show this;
when the airport code is used as a parameter for the dictionary, the matching/corresponding
value of the airport name will be returned;

(b) Award [2 max]

the lists are combined to form a single set/HashSet;
the set/HashSet will automatically eliminate any duplicates / will only hold unique values;

(c) (i) Award [1 max]

false;

(ii) Award [1 max]

{2, 6};

12. (a) Award [5 max]

Award [1] for making use of a temporary node;
Award [1] for Boolean;
Award [1] for list traversal;
Award [1] for ending traversal if found;
Award [1] for correct comparison;
Award [1] for returning the correct values (in both cases);

```
def find_player(self, name):
    temp = self.head
    found = False
    while temp != None:
        if temp.name == name:
            found = True
            print (temp.score)
            break
    else:
        temp = temp.next
    if found == False:
        print ("Not found")
```

(b) Award [5 max]

The algorithm will:

- make a temporary node equal to `head`;
- make use of a “previous” node;
- traverse the linked list (using the temporary node) comparing the score field of each node to the score field of the new node;
- until the insertion point is found;
- if insertion is at the beginning, link the new node to the head and make the head equal to the new node;
- if insertion is at the end, link the previous node to the new node;
- copy the *next* pointer of the node previous to the insertion point to the *next* pointer of the new node;
- link the previous node to the new node;

Note: Allow the use of diagrams to illustrate any of the above points
