Nov 2017 HL P1

Section A

1. Identify **two** essential features of a computer language. [2]

Fixed vocabulary;
Unambiguous meaning;
Consistent grammar;
Consistent syntax;
Provide a way to define basic data types and operations on those types (ability to write functions/procedures);
Provide ability of Input and output handling;
Provide some kind of loop that can be stopped / conditional statement / branching (conditional and unconditional branching);
It should have variables that reference computer memory, syntax for basic arithmetic and logical
operations on those memory locations;
It has to run on/be processed by a computer (*ie* it must have a compiler/interpreter);

Note: *do not accept aspects that address interoperability/portability/standards/ user friendliness*

2. In the context of a networked world, state the role of

(a) a client. [1]

A piece of computer hardware or software that accesses a service made available by a server /
The role of a client is to access a service made available by a server by sending a request for service;

Note*: the term client is to be understood only from the computing perspective, ie this is not a human.*

(b) a server. [1]

A program/host computer that awaits and fulfills requests from client programs (in the same or other computers) /
The role of a server is to fulfill requests from client programs (which can reside in the same or in other computers)

Note*: the term server is to be understood only from the computing perspective, ie this is not a human.*

3. Identify **one** method of inputting data that can improve the accessibility of a computer system for some users. [1]

*Award up to **[1 max]**.*
Text-to-speech;
Voice recognition;
Braille keyboards;
Touch screen;
Input from scanner;

4. Copy and complete the following truth table. [3]

A	B	A NOR B	(A NOR B) OR A
FALSE	FALSE	…	…
…	…	…	…

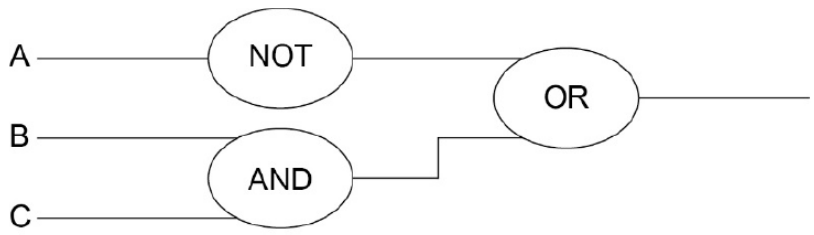
*Award **[1]** for all correct input values, **[1]** for a correct A NOR B column and **[1]** for a correct (A NOR B) OR A column.*

A	B	A NOR B	(A NOR B) OR A
FALSE	FALSE	TRUE	TRUE
FALSE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE
TRUE	TRUE	FALSE	TRUE

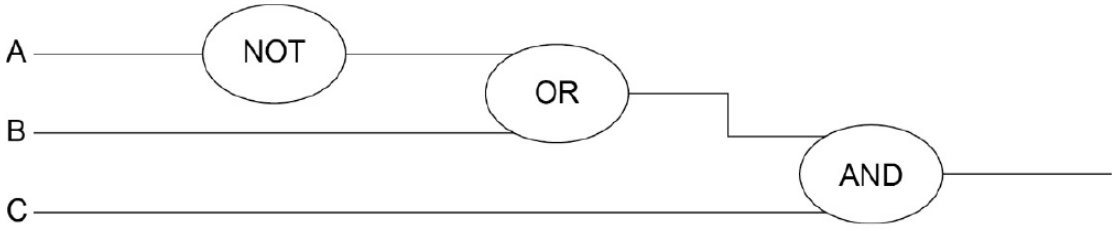
5. Construct a logic diagram for the Boolean expression

NOT A OR B AND C. [3]

*Award **[1]** for each correctly placed gate, up to **[3 max]**.*



***Note:** Award only **[1]** for an answer of:*



6. Consider the following recursive method, where N is a positive integer

```
mystery(N)
   if (N > 0) AND (N mod 2 = 0) then
     mystery(N-2)
   end if
   output N
end mystery
```

(a) Determine the output produced by the method call *mystery(5)*. [1]

5

(b) Determine the output produced by the method call *mystery(4)*. [3]

*Award up to **[3]** as follows:*
***[3]** for fully correct response (sequence of output) "0;2;4";*
***[2]** for response (sequence) "4;2;0" (all elements are correct, but they are in inverse order);*
***[1]** for response "0" (only base case is correct);*
OR
"0;2" (incomplete output, but initially correct, and with correct order);
OR
"–2;0;2;4","0;2;4;6" (correct sequence immersed in some unnecessary and incorrect context);

***[0]** in all other cases (eg responses "2", "4", "2;0", "2;4", "4;2");*
0
2
4

(c) Construct an iterative algorithm for the method *mystery()*, which uses a single *while* loop instead of recursion. [4]

Example answer 1
*Award marks as follows up to **[4 max]**. (There are 5 marking points);*
*Award **[1]** for determining whether `N` is odd/even;*
*Award **[1]** for correctly initializing and changing the value of the loop controlling variable (`K`);*
*Award **[1]** for the correct condition in the `while` loop;*
*Award **[1]** for output within the loop for an even `N`;*
*Award **[1]** for output after the loop for an odd `N`;*

```
mystery(N)
  if N mod 2 = 0 then
    K = 0
    loop while K <= N
       output K
       K = K + 2
    end loop
  else
    output N
  end if
end mystery
```

Example answer 2
*Award marks as follows up to **[4 max]**. (There are 5 marking points);*
*Award **[1]** for determining whether `N` is odd/even;*
*Award **[1]** for correctly initializing and changing the value of the loop controlling variable (`K`);*
*Award **[1]** for the correct condition in while loop (note `K < N`);*
*Award **[1]** for output within the loop for an even `N`;*
*Award **[1]** for outputting `N` after the loop;*

```
mystery(N)
  K = 0
  loop while (K < N) AND (N mod 2 = 0)
     output K
     K = K + 2
  end loop
  output N
end mystery
```

***Note:** No marks for any attempt of program that contains recursive calls.*

***Reminder:** in the Spanish version `mystery()` is called `incognita()`.*

***Remark:** A correct program produces in output numbers in an ascending order, only.*

7. The machine instruction cycle is the process by which a program instruction is fetched, decoded, executed and the results are stored.

   (a) State where all instructions and data are stored. [1]

Primary memory / RAM

   (b) Outline the role of the data bus and address bus in this process. [2]

*Award up to **[2 max]**.*
***Note**: there must be explicit reference to both address and data bus.*

Example 1
Buses are used as physical connections to carry information to the CPU;
The data bus transports data from/to CPU, whereas the address bus the memory address where the data is supposed to go/be;

Example 2
Data bus is a physical connection to transport data from-to CPU to be processed;
Address bus is a physical connection to transport an address of memory storage where data (transported in the data bus) should be read/written;

***Note:** Award **[1]**, for responses that show some understanding of use of buses in CPU, for address location and data transport without using specialist terminology.*

8. Define the term *bit*. [1]

Binary digit;
(Minimal) unit of storage that can be set to 0 or 1;

9. Outline what is meant by beta testing. [2]

*Award up to **[2 max]**.*
It involves sending sample software to the intended audience;
(Selected audience does not pay for this software);
To try/use the software product;
And give the feedback to the authors (which help in correcting bugs);

Section B

10. An application package used in an office includes a word processor.

(a) Describe how a spellchecker checks whether a word in a text file is correctly spelt or not. [2]

Each word in the text file is compared with words in a dictionary (held in memory/online);
If the word is found in the dictionary it is correctly spelt / if the word is not found in the dictionary, spellchecker will recognize that it is incorrectly spelt;

The office manager decides to buy and install new software and hardware.

(b) Outline **one** problem that may arise from the installation of new hardware and software in the office. [2]

*Award **[1]** for stating a problem and **[1]** for an elaboration, up to **[2 max]**.*
Users/employees might be afraid of these changes (for various reasons);
And not willing to help in this change;

Data migration problems;
For example, different file formats so conversion must be performed;

Employee efficiency may drop;
As they learn to use the new system;

Issue of compatibility with legacy software/hardware;
So features of new software/hardware may not work correctly;

The changeover to the new system can be achieved by either direct changeover or phased conversion.

(c) Compare direct changeover and phased conversion. [4]

*Award **[1]** for outlining what is meant by direct changeover, **[1]** for outlining what is meant by phased conversion, and then **[1]** for an advantage or disadvantage **of each**, up to **[4 max]**.*

Example answer
In direct changeover, the old software and hardware is completely replaced, in one move, by the new software and hardware;
Phased conversion involves selecting one section in the office for the direct changeover and other sections will be switched when the first section selected is running satisfactorily. Eventually the whole office has been changed;
A phased conversion is less risky than a direct changeover as any problems that might arise will be isolated in only one section in the office;
Direct changeover means everyone in the organization has same software/hardware and so there are no compatibility issues;

The new software allows basic text summaries and analysis to help check text files, including functions such as calculating word frequency.

(d) Identify **one** way of testing this software. [1]

*Award **[1 max]**.*
Debugging (***Accept***: white-box testing – *ie* structural testing/flow testing;
black-box testing / requirement testing);
User acceptance testing (alpha-testing) / Beta-testing;

One of the methods in this software is *findFirst(CH, CHARARRAY)* which accepts a character, *CH*, and a one-dimensional array of characters, *CHARARRAY*, and returns the position of the first occurrence of character *CH* in *CHARARRAY*. It returns *−1* if *CH* does not appear in the array *CHARARRAY*.

For example, consider the character array *MESSAGE*, which is of length 19.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]
A	U	R	O	R	A		M	U	S	I	S		A	M	I	C	A	.

For this array:

- The character at position 8 in the sentence is "*U*" and hence *MESSAGE[8] = "U"*.
- The character at position 18 in the sentence is "." and hence *MESSAGE[18] = "."*.
- The method *findFirst('A', MESSAGE)* returns *0*.
- The method *findFirst('S', MESSAGE)* returns *9*.
- The method *findFirst('Z', MESSAGE)* returns *−1*.

(e) Construct an efficient algorithm for the method *findFirst()*. You should use the function *len()*, which returns the number of characters in an array (for example, *len(MESSAGE)* returns *19*). [6]

*Award **[6 max]** according to the instructions of the closest template of example answer.*
***Remark**: Accept reference to either array `MESSAGE` or `CHARARRAY`.*

Example answers 1, 2, 3 and 4
*Award marks as follows up to **[6 max]**.*
*Award **[1]** for correctly initializing and changing indexes in `MESSAGE` (variable `K`);*
*Award **[1]** for comparing `CH` with the character at position `K` in `MESSAGE`;*
*Award **[1]** for terminating the `while` loop when there are no more characters to check (Accept 19 instead of `len(MESSAGE)`);*
*Award **[1]** for terminating the loop when `CH` is found in `MESSAGE`;*
*Award **[1]** for correctly returned value of `POS` when `CH` does not appear in `MESSAGE`. (**Note:** output not accepted);*
*Award **[1]** for correct assignment (or value) of `POS` when `CH` appears in `MESSAGE`;*

Example answer 1

```
POS = -1
FOUND = false
K = 0
loop while (K < len(MESSAGE)) AND (NOT FOUND)
     if MESSAGE[K] = CH then
          FOUND = true
          POS = K
     end if
     K = K + 1
end loop
return POS
```

Example answer 2

```
POS = -1
FOUND = false
K = 0
loop while (FOUND = false) OR (K < len(MESSAGE) //this order!
     if MESSAGE[K] = CH then
          FOUND = true
          POS = K
     end if
     K = K + 1
end loop
return POS
```

Example answer 3

```
K = 0
loop while (MESSAGE[K]! = CH) AND (K < len(MESSAGE))
     K = K + 1
end loop
if len(MESSAGE) = K then
     POS = -1
else
     POS = K
end if
return POS
```

Example answer 4

```
K = 0
loop while (MESSAGE[K]! = CH) OR (K < len(MESSAGE))
     K = K + 1
end loop
if K = len(MESSAGE) then
     POS = -1
else
     POS = K
end if
return POS
```

Example answer 5 (for-loop, decrementing the index)
*Award up to **[5 max]** for an answer with a `for` loop (as there are no marks awarded for efficiency);*
*Award **[1]** for starting from the **last** index of `MESSAGE`;*
*Award **[1]** for ending at the first index of `MESSAGE`;*
*Award **[1]** for comparing `CH` with the character at position `K` in `MESSAGE`;*
*Award **[1]** for correct defined/returned value of `POS` when `CH` does not appear in `MESSAGE`;*
*Award **[1]** for correct assignment of `POS` when `CH` appears in `MESSAGE`;*

```
POS = -1
loop for K from len(MESSAGE)-1 to 0
  if MESSAGE(K) = CH then
     POS = K
  end if
end loop
return POS
```

***Note**: Attempts of solutions that use an incremental for-loop, starting from the first index of the array, necessarily require the presence of a temporary variable to store the index!;*
*If none is present, then it is likely to be an incorrect attempt to the question (NAQ): award **[3 max]**;*

11. A wireless local area network (WLAN) is used to extend access to a school's wired local area network.

    (a) Identify **one** hardware component of the WLAN, other than computers. [1]

*Award up to **[1 max]**.*
Wireless router/modem;
Access points;
Switch;
Wireless repeater/extender/booster;

The advantages of this WLAN are user-mobility and economical access points.

(b) Outline **two** disadvantages of this WLAN. [4]

*Award **[1]** for identifying a disadvantage and **[1]** for an expansion, for **two** disadvantages up to **[4 max]**.*

Data transfer will decrease (compared with a wired LAN);
Because the number of computers using the network increases;
(and because) WLAN has lower bandwidth than a wired LAN;

Less data security;
As devices from outside the school can access the network/intercept transmissions;

More easily open to misuse;
As teacher/administrator cannot directly monitor a specific student/teacher/ machine;

Intermittent connectivity due to physical barriers (walls);
Results in low transfer/speed and may hinder operations.

***Note:** Accept any reasonable points, provided they are appropriately elaborated.*

(c) Identify **three** ways in which the network administrator can reduce the risk of unauthorized access to confidential data. [3]

*Award up to **[3 max]**.*
Give each user appropriate login details/passwords;
Different access rights for students, teachers, school administrators (file-level and share-level security);
All passwords and files /disk should be encrypted;
Use the latest WiFi protocol/WPA2;
Require MAC address authentication;
Password protect the documents;

***Note:** the focus of the question is on protection of confidential data (Firewalls not accepted)*

The concept of packet data transmission is used within this network. Figure 1 shows the simplified structure of a data packet.

Figure 1: The structure of a data packet

Header (12 bytes)	**Data** (112 bytes)	**Trailer / Footer** (4 bytes)
• address of sender • address of receiver • protocol • sequence number • …	Actual data to be transmitted (payload)	• transmission codes • error checking codes • control bits • …

(d) Define the term *protocol*. [1]

Set of rules for data transmission;

(e) With reference to **Figure 1**, explain how data is transferred by packet switching. [6]

*Award up to **[6]** as follows:*

*Award up to **[3 max]** for a general description of how data is transferred by packet switching. (**[1 max]** if this general description is very simplistic and refers to just the Header/Data/Trailer already shown in the question paper).*

*Award up to **[3 max]** for added detail that references the contents of the given data packet in the answer.*

Example of general description
Data is organized in specially formatted units (data packets) which are routed from source to destination using network switches and routers;

Network switches and routers determine how best to transfer the packet between a number of intermediate devices (routers and switches) on the path to its destination (rather than flowing directly over a single wire on the path to its destination);

Data packets are reassembled at the destination;

Example of referencing content
Addresses have to be in a standard format so that each switch/routing station recognizes the address;
Address of sender identifies the sending computer, so that any packets not received can be re-requested;
Address of receiver identifies intended recipient so it can be forwarded on correctly;
The ***protocol*** used must be identified so that the correct rules are followed;
Size of packet / size of fields in packet – All packets/fields must have the same size so that the data can be reassembled;
Sequence number so that packets can be reassembled in correct order;
Transmission codes to show whether the data packet is transmitted or re-transmitted;
Control bits, to maintain the integrity of the data by ensuring that the data received is the same as the data sent;
Error checking code – when an error is detected, an algorithm either corrects the error or requests that the packet is resent;

12. A biotechnology company owns a resource centre which collects and classifies organisms for use in research.

Only authorized employees are allowed access to some laboratories in the resource centre.

These laboratories are protected by locked doors. Each door is controlled by a separate microprocessor. A digital camera is used to scan the iris of an employee who wishes to enter the lab. If the employee is authorized the doors are unlocked.

(a) Identify **two** benefits of using a digital camera as an input device in this control system. [2]

*Award up to **[2 max]**.*
***([1]** for each of the **two** benefits).*

Digital cameras are relatively cheap;
Robust;
No need for AD conversion;
Generally very high quality pictures (useful to prevent malpractice);
Generally quick;

Example answer 1
No need for conversion because image is in digital format;
Thousands/millions of photos of eyes could be taken before needing replacement;

Example answer 2
It is directly connected to a microprocessor for image comparison;
It could be fitted (purchased), with a macro lens so that a close up scan of the Iris is possible;

(b) Outline the use of a microprocessor in this control system. [2]

*Award up to **[2 max]**.*

Processor compares the inputted image/pattern with the images stored in memory;

If a match is found, it sends a signal to unlock doors /
(if match is not found, it sends error message/the doors remain locked);

(c) Outline the function of an output transducer. [2]

Output transducer is a device (an actuator) which converts;
an electrical signal into physical quantity (a physical action);

Output transducer is a device which converts energy from one physical form to another;
eg electrical energy (signal) into electro-mechanical or kinetic energy / to produce action (lock/unlock door);

***Note**: Award one mark only for an answer that just says that the output transducer can be used to lock/unlock doors.*

The company is planning to use a centralized computer system to secure the resource centre's building.

(d) Compare a centrally controlled system with the system described above. [4]

*Award **[1]** for the meaning of centrally controlled system;*
***[1]** for the meaning of a distributed system, and;*
***[up to 2 max]** for an expansion/comparison addressing both kind of systems*
[up to 4 max];

Example answer
A centrally controlled system involves a central computer which controls all labs/doors;
A distributed system can have only a dedicated microprocessor with memory to control one of the labs/doors;

A centrally controlled system is more versatile;
Could be used in solving other business tasks (accept specific examples);
Can unlock all doors easily during an emergency;
Access rights can be updated easily;
Data is stored centrally and therefore easier to update;
A failure in a central system would affect all doors;

A distributed system can be programmed with ad hoc OS depending on the technologies used;
It may be practical choice when dealing with legacy systems/ specific devices/ old infrastructure/while updating the facilities;
It contributes a higher sense of partition of the physical space (territory) to groups / individuals;
Data are stored locally, so there is local consistency (and smaller size mean easier to manage);
Changes to data may be done locally, and if some data are common in two or more different
systems, global inconsistencies across different systems may be introduced;

The operating system has an important role in this system.

(e) Identify **two** functions of the operating system. [2]

*Award **[2 max]**.*
Memory management;
Resource allocation / Resource and Hardware management (printer, disk drives, *etc*.);
Booting / bootstrapping;
Loading and execute / provide service for applications software;
Disk/File system management;
Data security;
Provides a user interface to other levels of the machine;
etc.

Polling and interrupt are two operating system management techniques.

(f) Suggest with reasons which of these two techniques is the most appropriate for this centrally controlled system. [3]

*Award **[1]** for choosing "interrupt", **[1]** for explaining "interrupts" and **[1]** for justifying the choice in this context, up to **[3 max]**.*

Example answer:
Interrupt;
A signal sent from an input device to a computer causes the processor, and the main program that operates the computer (the operating system), to stop and figure out what to do next;
Interrupt is better in this situation because it does not waste central computer's time (other tasks could be performed);
(example of emergency – it must give quick response);

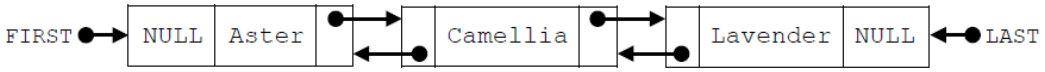
***Note:** Award **[1 max]** for an answer of "polling", but only if reasonably justified.*

Polling – the continuous checking of all input devices by processor to see what state they are in/to see whether they are still connected/want to communicate;
So a faulty device that is polled will not reply;

13. (a) Describe the features of a dynamic data structure. [2]

*Award up to **[2 max]**.*
Each node contains data and also a link to other nodes;
Links between nodes are implemented by pointers (a pointer references a location in memory or holds a memory address);
List size is not fixed / predetermined;

Consider the following doubly linked list which holds the names of flowers in alphabetical order.



(b) Explain how "*Primrose*" could be inserted into this doubly linked list. You should draw a labelled diagram in your answer. [6]

*Award up to **[6 max]** as follows. (There are 7 marking points)*
***[1]** create new node;*
***[1]** instantiation of values and pointers in new node;*
***[1]** state where the search starts from;*
***[1]** how to detect position for insertion;*
***[1]** update pointers in new node;*
***[1]** update pointers from the node at the insertion point, to the new node;*
***[1]** update external pointers;*
***Remark**: Some answers may just use illustrations alone, or very minimal explanations: see note below;*

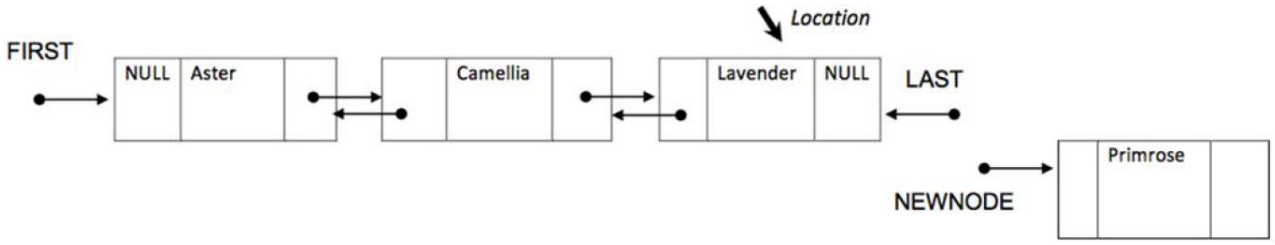
Create a new node (with pointer `NEWNODE`) with data field Primrose and two pointer fields (next and previous), to be inserted;

Perform a linear search, either from the beginning or end of the list (using pointers `FIRST` and `LAST`, on the alphabetically order list;

The location/position of insertion, is found by comparing nodes (Primrose to be inserted after Lavender, `LOCATION` points to Lavender) (***Accept*** *any description to that effect*);

(At the end of this phase, the situation looks as in ***Figure 1****)*

Figure 1



Then, continue by setting the "next" field/pointer in the newly created node to `NULL`;

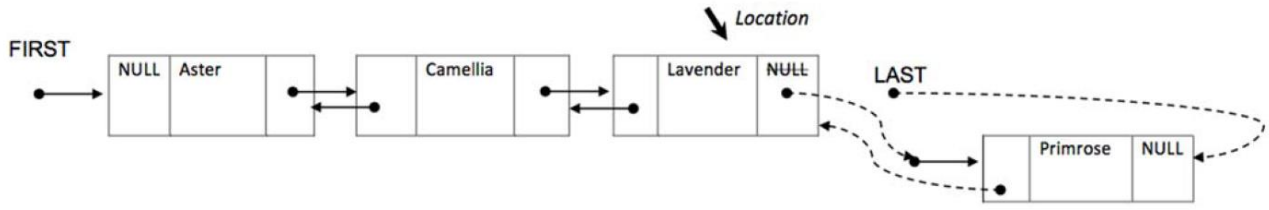
Set the "previous" pointer in the newly created node to the current `LAST` / to point to Lavender/ to point to the node detected by `LOCATION`;

Change/Set/Update the Lavender's "next" pointer to point to the new node / to link with the `NEWNODE` pointer (delete `NULL` in the field and link to the existing `NEWNODE` pointer);

Update the `LAST` pointer to point to the newly created node;

Eventually the final doubly linked list looks like this (**Figure 2**);

Figure 2



Note: *Award* ***[4 max]*** *for responses that return one or more drawings without any explanation at all, for evidence of these features:*
[1] *Evidence of creation of an initial new node for Primrose out of the list;*
[1] *The order of nodes Aster/Camellia/Lavender/Primrose is eventually correct;*
[1] *The two unidirectional links between Lavender and Primrose are (eventually) correctly displayed, from-to the appropriate fields;*
[1] *LAST points correctly to the appropriate field in the new node Primrose,* ***and*** *NULL fills the last field of the new node;*

Consider the two stacks: *FLOWERS* and *FRUITS*.

FLOWERS

Aster
Broom
Camellia
Day Lily
Lavender
Primrose
Yarrow

FRUITS

Apple
Cherry
Orange
Pear

(c) Show the output produced by the following algorithm. [4]

```
loop while (NOT FRUITS.isEmpty()) AND (NOT FLOWERS.isEmpty())
  X = FRUITS.pop()
  Y = FLOWERS.pop()
  if X < Y then
    output X
  else
    output Y
  end if
end loop
```

Award ***[1]*** *for each one in the correct order.*
Apple;
Broom;
Camellia;
Day Lily;
Note: *Solution for the Spanish version (in this order):*
Aster; Camelia; Lavanda; Lirio;

A third stack, *FLOFRU*, is needed. It should contain all the data from *FLOWERS* and *FRUITS* and will store it as shown below

FLOFRU

Yarrow
Primrose
Lavender
Day Lily
Camellia
Broom
Aster
Pear
Orange
Cherry
Apple

(d) Describe how the *FLOFRU* stack could be created. [3]

*Award marks as follows up to **[3 max]**.*

Example answer 1
Create an empty stack (`FLOFRU`);
pop all elements from `FRUITS` and **push** them onto `FLOFRU`;
Then **pop** all elements from `FLOWERS` and **push** them onto `FLOFRU`;

Example answer 2
Create an empty stack (`FLOFRU`);
While `FRUITS` is not empty
    pop an element from `FRUITS` and **push** it onto `FLOFRU`;
While `FLOWERS` is not empty
    pop an element from `FLOWERS` and **push** it onto `FLOFRU`;

***Note:** Award **[2 max]** for generic descriptions that do not use appropriate terminology on data structures and their operations.*

14. Consider the following two-dimensional array, *MAT*, with dimensions *6 × 6*.

	[0]	[1]	[2]	[3]	[4]	[5]
[0]	7	0	0	0	0	0
[1]	0	0	0	0	0	0
[2]	0	0	−3	0	9	0
[3]	0	0	0	0	0	0
[4]	0	0	−1	0	0	0
[5]	0	−6	0	0	−5	1

The value −1 is stored in MAT at position [4][2]. The position [4][2] means row 4 and column 2.

(a) State the total number of elements stored in MAT. [1]

36

(b) State the number of non-zero elements in MAT. [1]

7

A two-dimensional array in which most of the elements are zero is called a **sparse matrix**. A sparse matrix can be compressed by storing only non-zero elements using three one-dimensional arrays.

The **first array**, *VALUES*, stores all non-zero elements taken from the sparse matrix in row-major order (left-to-right then top-to-bottom order).

The length of the array *VALUES* is equal to the number of non-zero elements in the sparse matrix. For the sparse matrix above, *MAT*, the array *VALUES* is:

[0]	[1]	[2]	[3]	[4]	[5]	[6]
7	−3	9	−1	−6	−5	1

The **second array** is *ROWC*. *ROWC[i]* stores the number of non-zero elements, from row *0* to row *i* of the sparse matrix, **inclusive**.

The length of *ROWC* is equal to the number of rows in the sparse matrix. For *MAT* the array *ROWC* is:

[0]	[1]	[2]	[3]	[4]	[5]
1	1	3	3	4	7

For example, *ROWC[2]* stores *3* because in *MAT* there are three non-zero elements from row *0* to row *2*, inclusive.

The **third array**, *COL*, stores the column index for each non-zero element in the sparse matrix. *COL[i]* stores the sparse matrix column index for the non-zero element stored in *VALUES[i]*. For *MAT* the array *COL* is:

[0]	[1]	[2]	[3]	[4]	[5]	[6]
0	2	4	2	1	4	5

(c) Construct an algorithm that compresses a *6 × 6* two-dimensional array, such as *MAT*, into the three one-dimensional arrays described on page 8. You may assume that the *6 × 6* array is inputted and all three one-dimensional arrays are initialized. [6]

*Award marks as follows up to **[6 max]**. (There are 7 marking points)*
*Award **[1]** for initialization and correct changes of `K` (index/position in arrays `VALUES` and `COL`);*
*Award **[1]** for initialization and correct changes of `COUNT` (counts non-zero elements);*
*Award **[1]** for correct conditions in "row" loop;*
*Award **[1]** for correct conditions in "column" loop;*
*Award **[1]** for placing non-zero element at correct position in array `VALUES`;*
*Award **[1]** for placing the "column" index of the non-zero element at correct position in array `COL`;*
*Award **[1]** for placing `COUNT` at correct position in array `ROWC`;*

```
COUNT = 0
K = 0
loop for I from 0 to 5
   loop for J from 0 to 5
      if MAT[I][J]! = 0 then
         VALUES[K] = MAT[I][J]
         COL[K] = J
         K = K + 1
         COUNT = COUNT + 1
      end if
    end loop
    ROWC[I] = COUNT
end loop
```

Consider the following three arrays. They hold the compressed contents of a 7 × 7 sparse matrix, *BIGMAT*.

VALUES

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
15	20	37	43	57	60	70	89	90	92

ROWC

[0]	[1]	[2]	[3]	[4]	[5]	[6]
1	3	4	5	5	7	10

COL

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0	1	2	1	3	0	1	0	5	6

(d) For a given column, *C*, in *BIGMAT*, outline how it could be determined that this column contains no non-zero elements. [2]

*Award up to **[2 max]**.*

Example answer 1
Array `COL` should be searched for (value) `C`;
If (value) `C` is not found in array `COL` then this column (the column whose index in BIGMAT is C) holds only zeros;

Example answer 2
If the number of occurrences of (value) `C` in array `COL`;
Equals zero then this column holds only zeros;

***Example answer 3**: Award **[1 only]**.*
If `COL[C]` = `COL[C-1]`, then the column with index C in `BIGMAT` contains no non-zero-elements;
(Accept words to that effect: "if the difference between `COL[C]` and `COL[C-1]` is zero, then...");

(e) State how many rows in *BIGMAT* contain only zeros. [1]

1

(f) (i) State the index in *VALUES* of the first non-zero element in row *5* of *BIGMAT*. [1]

5

(ii) For a given row, *R*, in *BIGMAT*, determine the range of indexes in *VALUES* where non-zero elements in row *R* of *BIGMAT* are placed. You may assume that there is at least one non-zero element in row *R*. [3]

*Award marks as follows up to **[3 max]**.*
*Award **[1]** for realizing that the range should be determined differently for the first row (when row index `R` is `0`) OR correct range when row index is `0`;*
*Award **[1]** for correct first index in range (when row index `R` is not `0`);*
*Award **[1]** for correct last index in range;*

If row index `R` is equal to `0` then the range is from `0` to `ROWC[0]-1` ;
If row index `R` is not equal to `0` then the range is from `ROWC[R-1]`;
To `ROWC[R]-1`;

***Note**: Award **[2 max]** for a correct calculation of the indexes, but no unifying expression showing how they have been calculated is given).*