2226 – 4418M

Diploma Programme
Programme du diplôme
Programa del Diploma

Markscheme

May 2026

Computer science

Higher level

Paper 1

22 pages

Subject details: **Computer science HL paper 1 markscheme**

Mark allocation

Section A: Candidates are required to answer **all** questions. Total 25 marks.
Section B: Candidates are required to answer **all** questions. Total 75 marks.
Maximum total = 100 marks.

General

A markscheme often has more specific points worthy of a mark than the total allows. This is intentional. Do not award more than the maximum marks allowed for that part of a question.

When deciding upon alternative answers by candidates to those given in the markscheme, consider the following points:

Each statement worth one point has a separate line and the end is signified by means of a semi-colon (;).

An alternative answer or wording is indicated in the markscheme by a "/"; either wording can be accepted.

Words in (…) in the markscheme are not necessary to gain the mark.

If the candidate's answer has the same meaning or can be clearly interpreted as being the same as that in the markscheme then award the mark.

Mark positively. Give candidates credit for what they have achieved and for what they have got correct, rather than penalizing them for what they have not achieved or what they have got wrong.

Remember that many candidates are writing in a second language; be forgiving of minor linguistic slips. In this subject effective communication is more important than grammatical accuracy.

Occasionally, a part of a question may require a calculation whose answer is required for subsequent parts. If an error is made in the first part then it should be penalized. However, if the incorrect answer is used correctly in subsequent parts then **follow through** marks should be awarded. Indicate this with "**FT**".

General guidance

Issue	Guidance
Answering more than the quantity of responses prescribed in the questions	In the case of an "identify" question, read all answers and mark positively up to the maximum marks. Disregard incorrect answers. In the case of a "describe" question, which asks for a certain number of facts *eg* "describe two kinds", mark the first two correct answers. This could include two descriptions, one description and one identification, or two identifications. In the case of an "explain" question, which asks for a specified number of explanations *eg* "explain two reasons …", mark the first two correct answers. This could include two full explanations, one explanation, one partial explanation *etc*.

Section A

1. *Award **max** [3].*

Primary memory capacity/size;
Processor speed (frequency, word length, number of cores);
Secondary storage capacity/ type (SSD/HDD);
Quality of I/O devices (A*ccept factors that affect the quality of I/O devices, such as screen resolution, keyboard designs, etc*.);
Network connectivity option (wired or/and wireless);
Type of OS;
Graphic processing ability/type of GPU;
Cost;
Weight;
Energy consumption/ battery life;
Portability;
Ports available;
Ability to upgrade RAM/storage/GPU;
Durability/usability;
Compatibility with other systems;

2. *Award **max** [2].*

Hard drives/SSD/Backup storage devices;
Output devices/printers/speakers, etc;
Input devices/scanners/mice/keyboards/game controllers, etc.;
Database/data files/folders;
Software/Applications/OS;
Internet connection;
Server/ Mail Server/ File Server/ *other services*;
Bandwidth;

3. *Award **max** [1].*

To add an element to the end of the queue;

4. *Award **max** [1].*

To add an element to the top of the stack;

5. *Award **max** [3].*

Colours are represented numerically/binary/hexadecimal;
Primary colours: red, green, blue (One byte/8 bits/2 hexadecimal digits are assigned to each primary colour);
Combination of the three colour values (red, green, and blue) create a specific colour/ a specific colour is represented by 24 bits /3 bytes/ 6 hexadecimal digits/ hexadecimal values from 000000 to FFFFFF;
Colour intensity ranges from $0_{(10)}$ to $255_{(10)}$/ from $000000_{(16)}$ to $FFFFFF_{(16)}$ / Zero is completely dark and $255_{(10)}$ is as bright as possible in that colour (white);
The number of possible combinations: 2^{24} /16777216/ 16^6 colours;

***Note:** Accept 48 bits for representing a colour, 16 bits for each intensity of red, green and blue, range 0 for dark and 65 535 as bright as possible in one colour, and 2^{48} possible combinations because modern colour system uses 48 bits.*

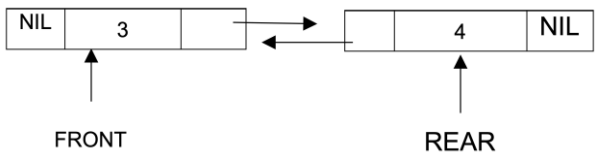
6. *Award **max** [4].*

*Award [1] for every 2 correct rows in the truth table**, ×4**.*

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

7. *Award **max** [4].*

*Award [1] for **only two** nodes in the DLL and all internal links correct.*
Award [1] for FRONT pointing to the node containing number 3.
Award [1] for REAR pointing to the node containing number 4.
Award [1] if the previous pointer of the first node and the next pointer of the last node are NIL.



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

A set of rules for transmitting (sending and receiving) data between computers (and other electronic devices) / set of rules for coordinating data transmission;
A set of rules that allows successful communication between devices;
A (standardized) set of rules for formatting / transmitting/ processing data across various networked devices;

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

Enable devices to communicate;
Assure data integrity;
Prevent congestion;
Prevent deadlock;
Detect/correct errors that occur during data transmission;
Specify the data format for transmission (e.g., headers, packets);
Incorporate security features (such as encryption and authentication);
Manage the flow of data/ Support addressing/routing (ensuring that data is delivered to the correct destination);
Facilitate interoperability/compatibility;
Prevent collision;

***Note**: Reward other reasonable answers.*

9. *Award **max** [1].*

To translate code/entire program written in a high-level programming language into machine code (that can be run by a computer/ understandable and executable by a computer);
To convert source code into object code (checks for syntax errors and produces an error report before an object file is generated);

10. *Award **max** [2].*

Example answers:

SaaS is a software distribution model in which cloud provider hosts apps (and data);
And makes them available to end users over the internet (a web browser and an internet connection needed to access);
on a subscription basis;

SaaS is a cloud-based method of providing software to users;
Users who subscribe to an application (rather than purchasing it once and installing it);
can log into and use the SaaS application from any (compatible) device over the internet;

***Note**: Reward other reasonable answers.*

Section B

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

Might not meet the needs/expectations of employees/clients;
Might not be technically possible;
Might not comply with laws and regulations;
Development cost might be higher than expected / over the company's budget;
Might take too long to design/implement;
Might be too complicated to use/ run;
Better alternative might be found during evaluation;
Might need more training/skills than planned;
Might create security/reliability/compatibility/usability problems;

***Note**: Reward other suitable answers such as poor usability, compatibility issues, security issues, slow/unstable- poor performance.*
.

(b) *Award **max** [4].*

Award up to [2] for strengths and up to [2] for weaknesses.

Strengths:
The new system can be evaluated with no consequences in case of failure (as it can fall back on the old system)/ Reduces risk of data loss during migration as both systems maintain records;
Staff can learn how to use the system at a pace;
Safer/secure/less risky, every transaction can be compared with the results of the old system/ enables performance/efficiency comparison with the old system;
Errors and bugs in the new system can be detected/fixed without downtime as the old system is running in parallel;
It can be cheaper running two system simultaneously than losing all data in case of failure;
***Note**: Reward other reasonable answers.*

Weaknesses:
It can be costly as two systems should be maintained/ more staff should be hired (trained);
The systems cannot be synchronized with each other;
Double data entry increases the chance of mistakes/inconsistencies/mismatched records between systems;
Having to do everything twice slows down the process/ increased employee workload;
Time consuming to complete the implementation of the new system/ to fully replace the old one adopting the new system;
***Note**: Reward other reasonable answers.*

(c) (i) *Award **max** [3].*

Internal/insider security threats*; Accept answers such as data theft, sabotage from employees.*
Cyberattacks caused by human error; *Accept answers such as careless decision-making, weak passwords, data leaks.*
Third-party security breaches; *Accept answers such as a cloud provider storing this data can be hacked, security hacking attacks.*
Man in the middle attacks (data interception during transmission);
Malware/viruses/malicious software; *Accept answers such as programs used to gather information through compromised devices, attached to downloadable files from emails or websites to disrupt network traffic/steal data, phishing scams.*
Ransomware (software that encrypts files/data and holds them for ransom forcing victims to pay for a decryption key to unlock the data);
SQL injections (a malicious SQL code inserted into user-input fields to read/modify/delete data in a database) / cyberattacks that exploit vulnerabilities in web applications / manipulate databases;
Distributed denial-of-service (DDoS) (attacks that causes websites to crash);

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

Email server manages the flow of messages between email clients and the internet;
Email server is responsible for processing and delivering incoming emails to the appropriate recipients and relaying outgoing emails to their intended destinations;
Email server handles the sending/receiving/storing email messages;
Email server manages user accounts and their mailboxes/ verifies sender and recipient identities (to prevent misuse)/ filters spam/blocks malicious emails/prevents unauthorized access;

(iii) *Award **max** [2].*

All messages that have been sent and received can be stored and searched through/email allows for easy referencing/ record keeping;
Email allows for the mass sending of messages/one particular message can be sent to several recipients at once;
Email allows automated reminders/automated replies/personalized invitations/embedded surveys/feedback forms;
Employees can read and respond at convenient times/ supports remote jobs/allows secured world-wide communication;
Large amounts of data/attachments (documents, images, videos) can be quickly transferred/ accessed from any device;
Hard copy of the message can be obtained;
Email is paperless, so not harmful to the environment/ eco-friendly;
Email is free/ simple to use/ cost effective/ quickly delivered;

***Note**: Reward other reasonable answers.*

(iv) *Award **max** [3].*

Award [1] for the reason why it is needed, [1] for an expansion and [1] for further elaboration.

Example 1:

Encryption ensures data confidentiality / only the intended recipient can understand the data / protection of sensitive data;
as encryption transform a plaintext message into a scrambled ciphertext format;
so that the message cannot be understood without the decryption key;

Example 2:

Encryption is needed to help maintain data security/ support legal/regulatory compliance;
as encryption involves using an algorithm and a cryptographic key to alter data/text;
so that even if intercepted, the message cannot be understood without the decryption key;

Example 3:

Encryption is needed to secure all outgoing and incoming email communication;
The channel used to send email to the recipient can be encrypted using Transport Layer Security (TLS);
to provide confidentiality / integrity/ authenticity;

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

*Award [1] for each clear difference between primary and secondary memory, **x3**.*

Primary memory stores data permanently (ROM) or temporarily (RAM), whilst secondary memory stores data permanently/ Nature of parts of primary memory varies (RAM- volatile, ROM- non-volatile), secondary memory is always non-volatile (data is retained even after shutdown);

CPU can directly access the data stored in primary memory (RAM), CPU cannot directly access the data stored in secondary memory;

RAM holds the data/programs that the computer is currently using/ holds the data while it is being executed by the CPU whilst secondary memory holds programs/data when not in use (various types of data/programs: OS, applications, videos, images, etc.);

Primary memory provides fast access to data/instructions, secondary memory provides slower access (compared to primary memory);

Storage capacity of the primary memory is limited/less capacity, secondary memory has more storage capacity;

Primary memory examples: RAM/ROM/cache memory/registers, secondary memory examples: hard disk drives (HDDs)/solid-state drives;

Primary memory is more expensive per unit of storage (due to high speed and complexity), secondary storage is more affordable per unit (making it suitable for bulk storage);

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

*Award [1] for a function and [1] for a reasonable expansion, **x2**.*

Memory allocation/OS allocates a portion of memory/ the computer's RAM to each running program;
which ensures that each program gets the memory it needs;

Memory deallocation/after execution of the program this memory is reallocated/freed/OS releases the memory the program was using;
making it available for other programs;

Memory protection/OS safeguards memory space/OS prevents one program from accessing or modifying the memory assigned to another program;
ensuring data integrity/security;

OS maps logical addresses (used by programs) to physical addresses (used by the hardware);
ensuring that the program can access the correct physical memory location;

OS keeps track of primary memory/of which bytes of memory are used by which user program/of memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used;
ensuring efficient utilization of available memory;

OS controls how much memory a program can use/ which pages to load which pages to store on disk;
to optimize memory usage;

Virtual memory management/ using disk space as an extension of RAM;
to run larger programs/more programs simultaneously/when RAM is full/overloaded;

Paging (a process is divided into fixed-size blocks called pages/physical memory is divided into frames of the same size);
allows efficient utilization of memory/ supports execution of large programs/ enables dynamic memory management;

Swapping (OS memory management technique where processes are moved between main memory/RAM and secondary storage);
allows the system to handle more processes than the available RAM can support/ prevents crashes by providing additional memory when RAM is exhausted/ enables memory-intensive programs to run on systems with limited RAM;

***Note**: Reward other correct responses.*

.

(b) (i) *Award **max** [1].*

Holds data/instruction read from/to be written into the main memory/RAM;

(ii) *Award **max** [3].*

Cache memory acts as a buffer between the main memory and the CPU/ is placed between the main memory and the CPU/ is located closer to the CPU;
Cache memory is a high-speed memory (high-speed SRAM technology)/ significantly faster than RAM (DRAM technology);
It stores the data/instructions that the CPU uses more frequently;
For any data, the CPU first checks the cache and then the main memory;
which decreases the average time to access the main memory (data)/ improves overall system performance;

(iii) *Award **max** [4].*

A system of communication paths used to connect various hardware components (CPU, memory, and input/output devices);
Buses enable the transfer of signals/data between these components ensuring smooth operation/efficient use of resources during fetch-execute cycle;
Two types of buses listed (data bus, address bus, control bus);
Data bus carries data between the CPU/ memory / I/O devices;
Address bus carries the physical location (address) of data to be accessed or written by the CPU;
Control bus transfers control signals for coordinating the overall system’s operation;

13. (a) (i) *Award **max** [2].*

Timer;
Keypad;
Touch screen;
Microphone;
Odometer;
A GPS receiver (acts as an input device in modern taximeters, providing position, velocity, and time);
Motion sensor;
Distance sensor;
RFID reader;

(ii) *Award **max** [3].*

The microprocessor uses the pre-programmed/pre-set fare per unit of distance and time; and takes in the various inputs (distance inputs, wait time inputs and inputs made by the driver);
Continually executes instructions stored in the memory;
to determine/calculate fare in real-time;
Sends signals to output transducers/actuators to display the fare/to print out the receipt;

(b) (i) *Award **max** [6].*

GPS consists of a network of satellites which orbit around the earth / works by communication with satellites;
Each satellite continuously broadcasts (radio) signals;
that contain information about the satellite: time of transmission and coordinates of the satellite;
A GPS receiver connected to a car/a GPS receiver in the taxi driver's cell phone receives these signals;
The difference between the (atomic) time of transmission and (atomic) time of receiving the signal is used to calculate the distance to each satellite;
By applying some basic mathematics (distance to the satellite = speed × travel time);
Using distances from (at least) three satellites the receiver can determine the position;
The method of trilateration (a fourth satellite is used to correct any errors);
is used to determine the receiver's exact/accurate position at various intervals;

(ii) *Award **max** [2].*

Convenience - passengers can book a taxi with just a few clicks from any place/ anytime they need it/ the app shows the nearest driver;
Passengers waiting for the taxi can track the location of their booked taxi to know the estimated arrival time;
Get an estimated fare before starting the ride/ get the total amount to be paid at any given moment (together with other relevant details of the trip);
Get an estimated time of arrival at your destination;
Passengers can see the route taken, ensuring the driver follows an efficient path;
Use ridesharing to reduce the total fare/ cut cost matching passengers on the same route;
Passengers can feel safe as they can share the ride details/location (with a family member in the event of late nights/isolated zones);

(iii) *Award **max** [2].*

The app shows the demand in real-time, so drivers can go to busy/high-traffic areas increasing their income;
Helps drivers easily find customers/ not to drive around the region looking for clients/ select clients based on the distance between their current position and client's location;
Get notifications for actions/events that take place during the ride (works, traffic accidents, objects on the road) and can change the route;
Helps drivers operate in unfamiliar areas;
Helps driver reach destinations quickly/ track their customer’s location and reach out to them quickly/ find the shortest routes;
Helps drivers get accurate/automatic/errorless calculation of fare;
Helps drivers review the ride/ payment statistics/ past rides/ look for history of trips done for the day/month;

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

Award [1] for a disadvantage and [1] for a reasonable expansion (how/why it is a disadvantage connecting with a feature of a dynamic data structure)

Unpredictable memory usage;
As size is not fixed/can shrink and grow during program execution;

Unpredictable/inefficient execution time;
because memory allocation/deallocation is managed at run-time;

Risk of memory leaks/unintended data modifications/fragmentation;
Memory needs to be explicitly allocated/deallocated for entire run-time;

Might cause overflow/the data structure might exceed maximum memory limit;
Dynamic memory allocation uses a lot of memory / uses additional pointers (uses more than, for example, memory needed for static data structures);

Slow access time;
Elements cannot be directly accessed/sequential access;

Slow searching;
which involves pointer traversal/sequential search;

Increased complexity in implementation and maintenance;
Managing pointers and dynamic allocation increases the likelihood of errors;

(b) *Award **max** [5].*

***Note:** All steps and recursive calls should be clearly shown. The work may be presented differently.*

```
func(4, ARR) = 4 *  func(3, ARR)                ;
      = 4  *  3 *  func(2, ARR)                 ;
      = 4 * 3  * 2 *  func(1, ARR)              ;
      = 4 *  3 * 2 *  1  *  func(0, ARR)        ;
      = 4 * 3  * 2 *  1  *  1 * func(-1, ARR))  ;
      = 4 * 3  * 2 *  1  * 1 * 1  =  24         ;
```

(c) *Award **max** [6].*

Example 1:

Award [1] for correctly initializing LEFT and RIGHT.
Award [1] for the correct loop.
Award [1] for correctly determining the middle index within the loop.
Award [1] for returning TRUE if X is equal to the middle element of the array.
Award [1] for changing the value of LEFT (LEFT = MID+1) if X is greater than the middle element of the array.
Award [1] for changing the value of RIGHT (RIGHT = MID–1) if X is smaller than the middle element of the array.
Award [1] for returning FALSE after the loop (when LEFT > RIGHT).

```
isThere (X, ARR)  //the subprogram heading may not appear
      LEFT = 0
      RIGHT = ARR.length - 1  //accept 9
         //initially the search space is the whole array
      loop while LEFT <= RIGHT
                  //getting the middle position of the search space
            MID = (LEFT + RIGHT) div 2
                  // accept div or '//' or '/'
                  //if X is the middle positions value
                  // it is contained in ARR
            if ARR[MID] = X
                then
                     return TRUE    //terminates loop
            end if
                // choose which half will be used
                // as the next search space
            if X < ARR[MID]
                        //discard the right half
                        //search only in the left half
                then
                   RIGHT = MID - 1
                else
                     // discard the left half
                   LEFT = MID + 1
             end if
   end loop
   return FALSE  // this will be executed only if
                 // X is not contained in ARR
end isThere
```

Example 2:

Award [1] for correctly initializing LOW, HIGH and POS.
Award [1] for the correct loop.
Award [1] for calculating the middle index of the array within the loop.
Award [1] for changing the value of POS if X is equal to the middle element of the array.
Award [1] for changing the value of LOW if X is greater than the middle element of the array.
Award [1] for changing the value of HIGH if X is smaller than the middle element of the array.
Award [1] for correctly returning TRUE or FALSE after the loop.

```
LOW = 0
HIGH = 9
POS = -1
    // loop till the search space is exhausted
    //or the position of X is found
loop while (LOW <= HIGH) and (POS = -1)
     if X == ARR[(LOW + HIGH) div 2]
         then
             POS = (LOW + HIGH) div 2
         else
             if X < ARR[(LOW + HIGH)div 2]
                 then
                     HIGH = (LOW + HIGH) div 2 - 1
                 else
                     LOW = (LOW + HIGH) div 2 + 1
             end if
     end if
end loop
if POS != -1
       then
            return TRUE
       else
            return FALSE
end if
```

Example 3 (sub-program `binarySearch` called within `isThere`):

*Award **max** [6].*
Award max [5] for the subprogram `binarySearch`:
[1] for the correct loop.
[1] for calculating the middle index of the array within the loop.
[1] for changing the value of POS if X is equal to the middle element of the array.
[1] for changing the value of LOW if X is greater than the middle element of the array.
[1] for changing the value of HIGH if X is smaller than the middle element of the array.
Award max [2] for the algorithm in subprogram `isThere`:
[1] for the correct `binarySearch` call.
[1] for correctly returning TRUE or FALSE (after calling the sub-program `binarySearch`).

```
binarySearch(ARR, LOW, HIGH, X)
// loop till the search space is exhausted
//or the position of X is found
      POS = -1
      loop while (LOW <= HIGH) and (POS = -1)
           if X == ARR[(LOW + HIGH) div 2]
               then
                   POS = (LOW + HIGH) div 2
               else
                   if X < ARR[(LOW + HIGH) div 2]
                       then
                           HIGH = (LOW + HIGH) div 2 - 1
                       else
                           LOW = (LOW + HIGH) div 2 + 1
                   end if
            end if
      end loop
      return POS // returns -1 if X not found
 end binarySearch


isThere (X, ARR)   //this subprogram heading may not appear

       P = binarySearch(ARR, 0, 9, X)
              // correct initial values in the sub-program call
              // corresponding parameters should match
       if P != -1
            then
               return TRUE
            else
               return FALSE
       end if
end isThere
```

15. (a) *Award **max** [4].*
*Award [1] for a feature of a 2D array and [1] for a reasonable expansion related to scenario, **x2**.*

The size of a 2D array is pre-determined/the size (rows × columns) of a two-dimensional array is set when it's declared (in compile time)/ Fixed size a two-dimensional array/ the size of array can't be dynamically altered/changed in run-time;
The total number of seats in the concert hall is known (150)/ The number of seats in the concert hall does not change (is fixed);

Dimensionality of 2D arrays/ 2D array has a height (number of rows) and a width (number of columns);
Seats in the concert hall are arranged in the form of a table/ data come naturally in the form of a table;

Accessing elements of two-dimensional arrays can be done using row index value and column index value/ direct access to array elements;
So that data stored about tickets/seats can be quickly accessed/updated/retrieved;

All elements in a two-dimensional array must be/are of the same data type/ homogeneity;
data stored about tickets/seats is of the same data type/ "char" data type;

(b) *Award **max** [4].*

Award [1] for a correct column loop (for/while).
Award [1] for initializing a counter variable (before the loop) and outputting its value (after the loop).
Award [1] for a correct condition in if statement (correct array element access and *correct comparison).*
Award [1] for increasing counter (if needed).

Example 1:

```
empty(EVENT, R)
   S = 0
    loop COL from 0 to 14  //accept EVENT[R].length-1
        if EVENT[R][COL] == 'F' then
            S = S + 1
         end if
  end loop
  output (S)
end empty
```

Example 2:

```
empty(EVENT, R)
    COUNT = 0
    loop ROW from 0 to 9     //accept EVENT.length-1
         loop COL from 0 to 14   //accept EVENT[ROW].length-1
            if ROW == R and EVENT[ROW][COL] == 'F'
                then
                    COUNT = COUNT + 1
               end if
         end loop
    end loop
    output (COUNT)
end empty
```

(c) *Award **max** [7].*

Award [1] for initializing total amount.
Award [1] for a correct row loop.
Award [1] for a correct column loop.
Award [1] for the correct access to elements in the EVENT array (use of indexes in the EVENT array).
*Award **[3 max]** for if statements, [1] for each correct if statement.*
Award [1] for outputting total amount.

Example 1:

```
total(EVENT)
       S = 0
       loop ROW from 0 to 9
            loop COL from 0 to 14
                 if EVENT[ROW][COL] == 'C' then
                     S = S + 10
                 end if
                 if EVENT[ROW][COL] == 'M' then
                     S = S + 12
                 end if
                 if EVENT[ROW][COL] == 'A' then
                     S = S + 20
                 end if
             end loop
        end loop
        output('Total amount: ', S)
end total
```

Example 2:

```
total(EVENT)
     TOT = 0
     loop J from 0 to 14  //column loop
         loop I from 0 to 9  // row loop
             if EVENT[I][J] == 'F'    //may not appear
                 then TOT = TOT + 0
                 else
                     if EVENT[I][J] == 'C'
                         then TOT = TOT + 10
                         else
                             if EVENT[I][J] == 'M'
                                 then TOT = TOT + 12
                                 else TOT = TOT + 20
                             end if
                     end if
            end if
         end loop
  end loop
  output('Total: ', TOT)
end total
```

Example 3 (uses the subprogram `counter`):

*Award **max** [3] for the subprogram `counter`.*
Award [1] for initializing and increasing the counter variable (`S`) if needed.
Award [1] for a correct row loop.
Award [1] for a correct column loop.
Award [1] for the correct access to elements in the two-dimensional array.
*Award **max** [4] for the subprogram `total`.*
*Award [1] for each correct subprogram call, **x3**.*
Award [1] for calculating correctly the total amount.
Award [1] for outputting total amount.

```
counter(ch, ARR)
  //returns the number of times the character ch occurs
  // in the 2D array ARR
     S = 0
     loop ROW from 0 to 9
          loop COL from 0 to 14
               if ARR[ROW][COL] == ch then
                    S = S + 1
               end if
          end loop
       end loop
       return S
end counter


total(EVENT)
      c1 = counter('A', EVENT)
                  //the number of Adult Tickets
      c2 = counter('M', EVENT)
                  //the number of Music Student Tickets
      c3 = counter('C', EVENT)
                  //the number of Child Tickets
     amount = c1 * 20 + c2 * 12  + c3 * 10
     output (amount)
end total
```