2024 HSC Software Design and Development Marking Guidelines

Section I

Multiple-choice Answer Key

Question	Answer
1	D
2	B
3	C
4	D
5	B
6	A
7	D
8	A
9	B
10	B
11	C
12	D
13	C
14	C
15	B
16	A
17	B
18	D
19	A
20	C

Section II

Question 21 (a)

Criteria	Marks
• Outlines TWO benefits of using a team of developers	3
• Identifies TWO benefits OR • Outlines ONE benefit	2
• Identifies ONE benefit	1

Sample answer:

Since the resource will involve working with images, video and itineraries, a larger development team will ensure that people with appropriate skills can be utilised.

As the resource is needed by the end of the school year, it can be assumed that a short development time is needed. A larger team may be able to meet time restraints.

Question 21 (b)

Criteria	Marks
• Describes the use of a Gantt chart in the management of this project	3
• Shows some understanding of the use of a Gantt chart	2
• Provides some relevant information	1

Sample answer:

A Gantt chart allows individual tasks to be scheduled (such as creation of video clips) and allocated to members of the development team with the required skills. It will enable the project manager to schedule the work required to complete all of the components by the required deadline. It allows the project manager to visually see the timeline and task dependencies and respond appropriately if there are delays.

Question 22 (a)

Criteria	Marks
• Provides reason(s) for consulting the shopkeeper	2
• Provides some relevant information	1

Sample answer:

The developer should interview the shopkeeper to determine the required time frame and budget. The developer could gather data about the nature of the business and products available, enabling them to create a system that will meet their needs.

Question 22 (b)

Criteria	Marks
• Distinguishes between user and technical documentation for this system	3
• Outlines aspects of both user and technical documentation	2
• Shows some understanding of documentation	1

Sample answer:

User documentation is specifically designed to assist the customer in accessing the system and purchasing groceries and will be in the form of online documentation, accessed through a help function. Technical documentation is aimed at recording the software development process to assist both in development and in subsequent maintenance. This can include modelling tools, records of testing procedures and algorithms.

Question 22 (c)

Criteria	Marks
• Provides a substantially correct data dictionary	3
• Provides some correct entries	2
• Provides some relevant information	1

Sample answer:

Data Item	*Data type*	*Storage size bits/bytes*	*Description*	*Examples*
Surname	String	40 bytes	Surname of customer	Faruga
Item name	**String**	20 bytes	Name of item	2 L full cream milk
Quantity	**Integer**	**2 bytes**	Number of items	4
Order	**Array of records**		Details of items in the order	Dozen eggs 1 $5.00 Flour 1 kg 2 $2.00 2 L skim milk 4 $3.50
Total cost	**Floating point**	4 bytes	Total amount due	**$45.20**
Paid?	**Boolean**	**1 byte**	Has payment been made	**T (or Y)**

Question 22 (d)

Criteria	Marks
• Recommends TWO ways of testing this website	3
• Outlines TWO relevant ways of testing OR • Recommends ONE relevant way of testing	2
• Shows some understanding of testing	1

Sample answer:

Once the website is hosted on a server, access should be tested using multiple devices with a variety of transactions. This would enable the ability of the website and the system to cope with the anticipated volume and variety of interactions without having to make it available to the broader public.

It is important that the website produces correct totals for orders. This can be checked by entering carefully selected test data and comparing actual output with expected output.

Question 23 (a)

Criteria	Marks
• Completes a substantially correct desk check	3
• Completes a partially correct desk check	2
• Provides some relevant information	1

Sample answer:

Max	Num_Students	Mark	Average	Output
100	0	20		
	1		20	
		60		
	2		30	
		30		
	3		10	
		40		
	4		10	100, 10

Question 23 (b)

Criteria	Marks
• Provides the necessary modifications	3
• Provides ONE correct modification OR • Identifies TWO errors	2
• Provides some relevant information	1

Sample answer:

Line 20 should be Max = 0

Line 15 Sum = 0

Line 95 Sum = Sum + Mark

Line 110 change to Average = Sum/Num_Students

Question 24 (a)

Criteria	Marks
• Constructs a correct IPO diagram for this module	3
• Constructs a partially correct IPO diagram	2
• Provides some relevant information	1

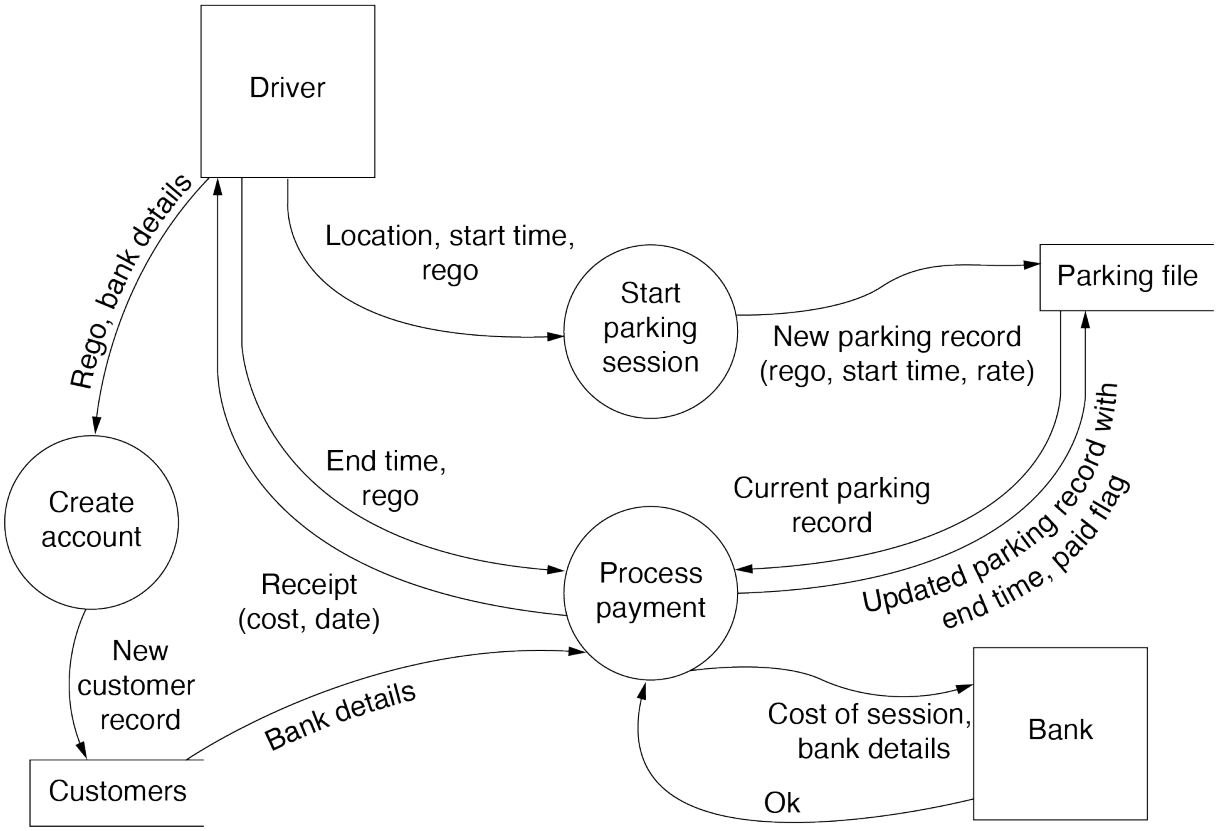
Sample answer:

I	P	O
Location	Determine rate	Rate
Start_time End_time Rate	Cost = (End_time – Start_time)* Rate	Cost

Question 24 (b)

Criteria	Marks
• Constructs a substantially correct data flow diagram that includes processes, data flows, storage and external entities	4
• Constructs a data flow diagram that is relevant to the parking app and includes an external entity and most of the processes and data flows	3
• Constructs a data flow diagram with some elements relevant to the parking app	2
• Shows a basic understanding of data flow diagrams	1

Sample answer:



Question 24 (c)

Criteria	Marks
• Describes actions that address an inclusivity issue AND a privacy issue for this system	3
• Outlines an action that addresses an inclusivity issue OR a privacy issue	2
• Shows some understanding of inclusivity or privacy issues	1

Sample answer:

Drivers may use a range of devices. This is an inclusivity issue that the developer could have addressed by ensuring that this new system is available across all common platforms and screen sizes.

A privacy concern arises if unauthorised people access the generated data about drivers' movements. The developer could have implemented a security process that guards against external access to the stored banking details.

Question 25

Criteria	Marks
• Explains the use of various types of documentation by the contract programmer	4
• Describes a number of relevant types of documentation OR • Explains the use of ONE type of documentation	3
• Outlines some relevant types of documentation	2
• Provides some relevant information	1

Sample answer:

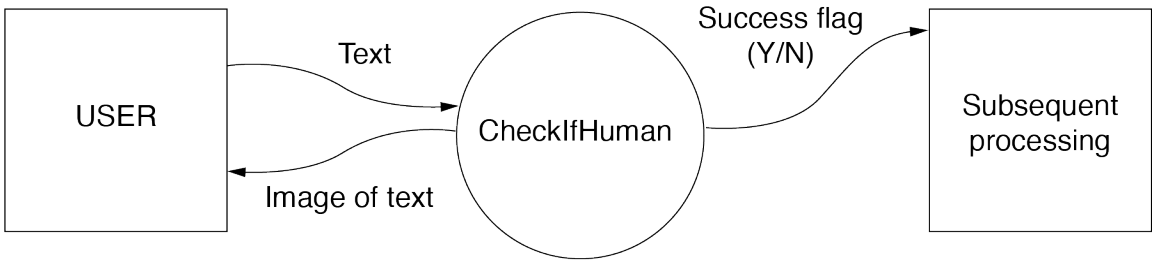
By referring to technical documentation such as DFDs, algorithms and structure charts, the contract programmer can familiarise themselves with how their particular module fits into the overall project.

They can read existing source code to see the purpose of various modules and lines of code. This allows them to make necessary changes to ensure the new code is compatible with the existing code. A Gantt chart can be used to see the expected timing for their allocated tasks so they can plan their workflow. A data dictionary can be referred to, to confirm the use of particular variables.

Question 26 (a)

Criteria	Marks
• Creates a substantially correct context diagram	2
• Creates a recognisable attempt at a context diagram	1

Sample answer:



Question 26 (b)

Criteria	Marks
• For BOTH subroutines, explains the importance of implementing the logic as subroutines	4
• For ONE of the subroutines, explains the importance of implementing the logic as a subroutine	3
• Outlines ONE advantage of using a subroutine	2
• Demonstrates some understanding of subroutines	1

Sample answer:

It is useful to define CheckIfHuman as a subroutine to allow for further developments in the relevant technology. It is a trivial matter to replace the current logic for the subroutine with code that utilises the new technology, while still providing the same output, ie a success flag.

DisplayMessage appears to be a generic subroutine that appropriately formats and displays a relevant message. This makes it easy to change the overall format of the display if required, once only, in the updated subroutine. It is also easy to allow for the inclusion of a much larger range of messages in the future, simply by increasing the numbers of variables that are appropriately initialised and passed to the subroutine.

Question 27

Criteria	Marks
• Provides correct explanations for BOTH modifications	3
• Provides a correct explanation for ONE modification	2
• Demonstrates some understanding of the algorithm	1

Sample answer:

UNTIL F>N/2 – once the process has reached halfway, the same checks would be repeated unnecessarily.

Adding 'OR palindrome = FALSE' – there is no need to keep checking once a non-matching letter is found.

In both cases, the loop can be exited early, improving efficiency.

Question 28

Criteria	Marks
• Distinguishes between the use of breakpoints and debugging output statements	3
• Describes the use of breakpoints OR debugging output statements OR • Outlines reasons for the use of breakpoints AND debugging output statements	2
• Provides some relevant information	1

Sample answer:

A debugging output statement is added to the code by the developer and can be used to indicate that a certain section of code has been reached or to show the contents of a variable at any specific stage of execution. A debugging output statement does not interrupt execution.

A breakpoint can be added by the developer to halt execution and to allow the programmer to then check the value of specific variables or to step through the program line by line from then on.

Question 29

Criteria	Marks
• Demonstrates the similarities and differences between compilation and interpretation, and their uses	4
• Demonstrates some understanding of the use of compilation AND interpretation	3
• Shows some understanding of compilation AND interpretation	2
• Provides some relevant information	1

Sample answer:

Both of these translation methods produce machine code that can be executed. Compilation checks the program for syntax errors and produces a list of all errors. Once errors have been removed, the compiler translates the source code to machine code and stores this as an executable. On the other hand, interpretation translates and executes source code one line at a time until it finds the next syntax error.

Compilation is used when an executable software application is required as a stand-alone package for a given operating system, where the source code is not required.

Interpretation is useful during debugging because the developer can concentrate on one error at a time. Interpreted source code is distributed with interpretation software, meaning the source code is accessible, potentially leading to copyright issues.

Question 30 (a)

Criteria	Marks
• Provides a substantially correct algorithm that includes the following features: – Randomly allocates customers one at a time from the customer array using a loop – Allocates the correct number of tickets – Updates array(s) correctly – Deals correctly with last few remaining tickets	5
• Provides an algorithm that addresses most of the key aspects of the problem	4
• Provides an algorithm that addresses some key aspects of the problem	3
• Provides an algorithm that shows some understanding of the problem	2
• Shows some understanding of the problem	1

Sample answer:

```
BEGIN AllocateTickets
    TixIndex = 1
    WHILE TixIndex < 10000
        Pos = Rand (1, 2000)
        IF (CustomerArray[Pos].Allocated = FALSE) THEN
            IF (TixIndex + CustomerArray[Pos].Num_Tickets > 10000) THEN
                NumToInsert = 10000 – TixIndex
            ELSE
                NumToInsert = CustomerArray[Pos].Num_Tickets
            ENDIF
            UserToInsert = CustomerArray[Pos].CustId
            Allocate(UserToInsert, NumToInsert, TixIndex)
            CustomerArray[Pos].Allocated = TRUE
        ENDIF
    ENDWHILE
END AllocateTickets

BEGIN Allocate(CustId, Num_Tickets, TixIndex)
    FOR i = 1 to Num_Tickets
        Ticket_Array[TixIndex].AllocatedCustomer = CustId
        Add 1 to TixIndex
    NEXT i
END Allocate
```

Question 30 (b)

Criteria	Marks
• Provides an explanation of the use of this stub in testing	2
• Shows some understanding of a stub	1

Sample answer:

For the function Rand(), it is critical that it returns relevant values (in this case a random number in the specified range). It is not possible to write a stub that can pass back the required values to fully test the main program logic without including the logic to generate the random numbers required by the calling program, to enable it to allocate tickets meaningfully.

Section III

Question 31 (a) (i)

Criteria	Marks
• Outlines possible causes of incorrect output	2
• Provides some relevant information	1

Sample answer:

Incorrect output could be caused by facts that are incorrect, missing or contradictory, or by rules being incorrectly defined.

Question 31 (a) (ii)

Criteria	Marks
• Produces a correct rule	2
• Provides some relevant information	1

Sample answer:

SmallerPlanet(A, B) :– Size(A, B), Weight(A, B)

Question 31 (a) (iii)

Criteria	Marks
• Describes how backward chaining is used to determine the truth of this query	3
• Provides a description of backward chaining	2
• Provides some relevant information	1

Sample answer:

Backward chaining assumes that the query is true, ie Mercury is smaller than Saturn. Then the inference engine would search through the available facts to find those facts that are relevant to the query, ie, any facts that pertain to Mercury and/or Saturn. The inference engine would search through the available facts to establish if the query can be sustained. If it can, then TRUE would be returned. If the facts available cannot establish the truth of the query, then FALSE is returned.

Question 31 (b)

Criteria	Marks
• Proposes a relevant paradigm for this game with appropriate justification	4
• Proposes a paradigm for this game with some justification	3
• Demonstrates some understanding of a paradigm	2
• Provides some relevant information	1

Sample answer:

The OOP paradigm is an appropriate choice. Encapsulation can be used to protect the relevant attributes of the objects from being modified from outside the object, thus ensuring that multiple instances of characters are uniform in appearance and cannot be altered. Because this feature is an integral part of OOP, the programmer does not need to specifically provide code for this.

Inheritance can be used to efficiently implement a class that deals with the appearance of the character, but which could be used to alter attributes, such as hat, through the use of separate subclasses.

Question 31 (c) (i)

Criteria	Marks
• Correctly describes the output and potential issue(s)	3
• Correctly describes the output OR • Describes a potential issue	2
• Provides some relevant information	1

Sample answer:

This code will generate 16 instances of the tree class that are all oak trees (with the same height of 5 m) which are placed in random locations on the screen. One potential issue that could occur is that two or more trees may have very similar x and y coordinates, meaning that they either overlap or completely cover each other.

Question 31 (c) (ii)

Criteria	Marks
• Describes relevant benefit(s) of using subclasses for different types of trees	3
• Demonstrates some understanding of subclasses	2
• Provides some relevant information	1

Sample answer:

Having different subclasses of the class tree will allow for common tree features to be defined in a class. Features such as flowering, colour or texture mapping that is individual to a type of tree, should be defined in a subclass. If this were to be done in the class tree, the developer would need to include attributes that are irrelevant to other tree types, making the code inefficient.

Question 31 (c) (iii)

Criteria	Marks
• Provides substantially correct pseudocode	3
• Provides pseudocode that addresses some of the requirements	2
• Demonstrates some understanding of the problem	1

Sample answer:

```
tree Background[10]
Set offsetX, offsetY = 10
Set index = 0
Set row to 0
FOR y = 0 to 3
    FOR x = 0 to row
        Background[index].height = 600
        Background[index].type = “maple”
        Background[index].xcoord = x * offsetX
        Background[index].ycoord = y * offsetY
        Increment index
    NEXT x
    Add 1 to row
NEXT y
```

Question 32 (a)

Criteria	Marks
• Performs the subtraction correctly with working provided	3
• Demonstrates some understanding of the binary subtraction process	2
• Demonstrates some understanding of binary arithmetic	1

Sample answer:

–00010001 = 11101110 + 1
= 11101111

```
  01011100
  11101111
-----------
1 01001011
```

(ignore the left-most bit)

Question 32 (b)

Criteria	Marks
• Explains why the binary values differ by 32	2
• Provides some relevant information	1

Sample answer:

The binary code for each uppercase letter and its equivalent lowercase letter are identical except for the 5th bit. The numerical value of that bit is 32 ($=2^5$), which makes it a simple matter of adding 32 to the binary equivalent of an uppercase letter (turning on the 5th bit) to derive its lowercase equivalent.

Question 32 (c)

Criteria	Marks
• Provides a substantially correct algorithm that includes the following features: – Loop for each pair of bits – Determination of sum and carry for each pair of bits – Use of carry forward	3
• Provides an algorithm that addresses some of the key features of the problem	2
• Demonstrates some understanding of the problem	1

Sample answer:

```
BEGIN
    carry = 0
    FOR n = 8 TO 1 STEP –1
        CASEWHERE FirstNum(n) + SecondNum(n) + carry =
            0   :   Sum(n) = 0
                    carry = 0
            1   :   Sum(n) = 1
                    carry = 0
            2   :   Sum(n) = 0
                    carry = 1
            3   :   Sum(n) = 1
                    carry = 1
        ENDCASE
    NEXT
END
```

Question 32 (d)

Criteria	Marks
• Provides an explanation that demonstrates sound understanding of the problem	3
• Describes how binary integers are stored, identifying limitations	2
• Provides some relevant information	1

Sample answer:

A 32-bit signed integer can store positive numbers from 1 to $2^{31} – 1$. The left most bit represents the sign.

At some stage this maximum value will be reached with the right most 31 bits all being 1. All dates and times after that are not able to be represented in this format because, once the value has passed $2^{31} – 1$, the left most bit will be a 1, causing the number to be interpreted as a negative number.

Question 32 (e) (i)

Criteria	Marks
• Constructs a correct data block	2
• Provides some relevant information	1

Sample answer:

010110100101

Question 32 (e) (ii)

Criteria	Marks
• Provides a correct data stream design with all components identified	3
• Provides a data stream design with some correct components	2
• Provides some relevant information	1

Sample answer:

Header

Sensor number 2 bits

Data

Date and time 4 bytes (32 bits)

Detected speed (kph) 8 bits

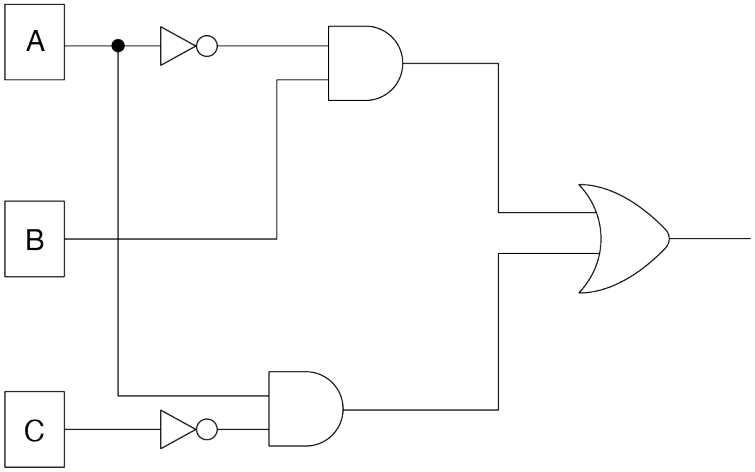
Trailer

Error detection 8 Bits

Question 32 (f)

Criteria	Marks
• Provides a correct simpler circuit	4
• Provides a substantially correct simpler logic circuit OR • Provides a correct truth table	3
• Provides a partially correct simpler logic circuit OR • Provides a substantially correct truth table OR • Provides a correct Boolean expression	2
• Provides some relevant information	1

Sample answer:



Answers could include:

$\bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$

$= \bar{A}B\,(\bar{C} + C) + A\bar{C}\,(\bar{B} + B)$

$= \bar{A}B + A\bar{C}$

A	B	C	$\bar{A}B$	$A\bar{C}$	$B\bar{C}$	Out
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	1	0	1	1
0	1	1	1	0	0	1
1	0	0	0	1	0	1
1	0	1	0	0	0	0
1	1	0	0	1	1	1
1	1	1	0	0	0	0

2024 HSC Software Design and Development Mapping Grid

Section I

Question	Marks	Content	Syllabus outcomes
1	1	Software Licences	H3.1
2	1	Development Approaches	H4.2
3	1	Installation Methods	H4.2
4	1	CASE Tools	H5.3
5	1	Data Types	H1.3
6	1	Software Market	H3.1
7	1	Test Data	H4.2
8	1	Data Types	H1.3
9	1	Social and Ethical Issues	H3.1
10	1	Railroad Diagrams	H5.3
11	1	EBNF	H5.3
12	1	Desk Checking	H4.3
13	1	Machine Code	H1.1
14	1	CPU Components	H1.3
15	1	Desk Check	H4.3
16	1	Structure Chart/Algorithms	H5.2
17	1	Error Types	H4.2
18	1	Translation Processes	H1.1
19	1	Sort Methods	H4.2
20	1	Validation	H4.3

Section II

Question	Marks	Content	Syllabus outcomes
21 (a)	3	Development By Teams	H4.2, H6.3
21 (b)	3	Gantt Charts	H5.1
22 (a)	2	Development Cycle	H4.2
22 (b)	3	User/Technical Documentation	H5.2
22 (c)	3	Data Dictionary	H5.2
22 (d)	3	Testing	H4.2
23 (a)	3	Desk Check	H4.3
23 (b)	3	Modification of Algorithm	H4.3, H5.2
24 (a)	3	IPO Chart	H5.2
24 (b)	4	DFD	H5.2
24 (c)	3	Social/Ethical Issues	H3.1, H3.2
25	4	System Documentation	H5.2

Question	Marks	Content	Syllabus outcomes
26 (a)	2	Context Diagram	H5.2
26 (b)	4	Subroutines	H4.3
27	3	Efficient Code	H4.3
28	3	Debugging	H4.2, H5.3
29	4	Interpreter vs Compiler	H1.3
30 (a)	5	Algorithm Design	H4.3, H5.2
30 (b)	2	Stubs	H5.2

Section III

Question	Marks	Content	Syllabus outcomes
31 (a) (i)	2	Unexpected Error from Logic	H9.4.1
31 (a) (ii)	2	Logic Paradigm Rule	H9.4.1
31 (a) (iii)	3	Backward Chaining	H9.4.1
31 (b)	4	OOP Concepts	H9.4.1
31 (c) (i)	3	Desk Checking OOP Code	H9.4.1
31 (c) (ii)	3	Subclasses	H9.4.1
31 (c) (iii)	3	Designing OOP Algorithm	H9.4.1
32 (a)	3	Binary Subtraction	H9.4.2
32 (b)	2	ASCII	H9.4.2
32 (c)	3	Binary Addition Algorithm	H9.4.2
32 (d)	3	Integer Representation	H9.4.2
32 (e) (i)	2	Construct a Data Stream	H9.4.2
32 (e) (ii)	3	Data Stream Design	H9.4.2
32 (f)	4	Circuit Design/Boolean Algebra	H9.4.2