A number of issues arose in the marking of this exam:

Whilst it is possible to get a B award by just doing the C and B questions (similarly get a C award by just doing the C questions) this is a risky strategy. Many students make mistakes in these questions and the marker then relies on the A section (or B section) to confirm the B (or C) award. I would encourage all students to complete as much of the paper as possible. Frequently, parts of the A question can be answered without fully understanding the program (Section B) or the Toy code (Section D).

A greater number of students left sections completely blank this year. This can only result in a Z award which makes the subject impossible to pass. You must attempt every section. Students must show working as it is difficult to give partial marks if no working is shown.

**Section A**

**Question 1**

(a) The extra cost for opting for the power processor is $200

(b) If the cost was not set to 400 at start of “Calculate Button” code then each time the button is pressed it would increase the cost.

(c) Initially

\[
\text{set more_memory = “No”}
\]

\[
\text{set power_processor = “No”}
\]

When text is entered into “Want more memory?” TextField

set more_memory to value in “Want more memory” TextField

if power_processor is “No” and more_memory is “Yes”

\[
\text{set more_memory to “No”}
\]

display “If the more powerful process is not selected more memory cannot be selected.”

display “More Memory?” more_memory

When text is entered into “Want power processor?” TextField

set power_processor to value in “Want power processor” TextField

if power_processor is “No” and more_memory is “Yes”

\[
\text{set power_processor to “Yes”}
\]

display “If the more memory is selected the more powerful process must be selected.”

display “More Memory?” more_memory

display “Power Processor?” power_processor

When the “Calculate” button is pressed

set cost = 400

if more_memory is “Yes”

\[
\text{set cost = cost + 80}
\]

if power_processor is “Yes”

\[
\text{set cost = cost + 200}
\]

display “Cost of computer: $” cost

This section proved to be challenging for most students.

Part (a) was generally well answered but students need to read the question carefully as answers of “600” or “400+200=600” were all too common.

Part (b) proved too difficult. Most students were unable to recognise the subtly that the calculate button could be used repeatedly and hence the cost needed to be reset each time.
Part (c) also proved to be challenging. Better answers automatically changed power-processor to yes when want-more-memory was set to yes but then failed to prevent the When text is entered into “Want power processor?” TextField method from changing want-power-processor back to no. Many students did the following:

if more_memory is “Yes”
    set cost = cost + 80
    set power_processor to “Yes”

Whilst not advising the user that something was wrong with their data, this method still attracted good marks.

Question 2

Initially

set random = 0
set count = 0
set score = 1
set back = 0
set finished = false
set gone_back = false

When a number is entered into “Go back by” TextField

if gone_back is false and finished is false
    set gone_back to true
    set back to value in “Go back by” TextField
if back is greater than 3
    set back = 3
set score = score - back
if score is less than 1
    set score = 1
display “Score is ” score

When the “New Random value” button is pressed

if finished is false
    set gone_back = false
    set random to random value between 2 and 5
    display “new random value is” random
    set score = score + random
    set count = count + 1
    if score equals 10
        display “Perfect 10! ... in ” count “ turns.”
        set finished = true
    else
        if score is greater than 10
            set score = 1
        display “Score is ”score

Many students compensated for poor answers in Q1 by doing better in this section.
Part (a) was generally well done but some answers set the value back to 0 rather than 1 or forgot to change it at all. Students need to make sure they follow the instructions.

Part (b) was also well done with a number of strategies deployed. Many checked that back >= 1 as well although there was no instruction that back was not allowed to be 0 (checking back >= 0 was a good idea). Many students neglected to reset the score to 1 if the score had gone below 1 (as instructed to do). Others checked in advance if the score would go below 1 and then blocked the score = score - back instruction. Many students attempted to advise the user that their value for back was invalid but then failed to clearly show an else to run the remaining code when the data was valid.
Part (c) was only attempted by better students. A Boolean variable was clearly the best way to do this but some students neglected to re-set gone-back to false in the When the “New Random value” button is pressed method or they forgot to initialise gone-back.

**Question 3**

(a)

```
Lawn area (m²):

Garden furniture?:

Garden Waste (m³):

General Waste (m³):

Garden maintenance

Quote($):
```

(b)

Initially

set lawn_area = 0
set garden_waste = 0
set general_waste = 0
set garden_maintenance = 0
set garden_furniture = “No”
set quote = 0

*When a number is entered into “Lawn area” TextField*

set lawn_area to value in “Lawn area” TextField

*When a number is entered into “Garden Waste” TextField*

set garden_waste to value in “Garden Waste” TextField

*When a number is entered into “General Waste” TextField*

set general_waste to value in “General Waste” TextField

*When a number is entered into “Garden maintenance” TextField*

set garden_maintenance to value in “Garden maintenance” TextField

*When text is entered into “Garden furniture?” TextField*

set garden_furniture to value in “Garden furniture?” TextField

*When the “Calculate” button is pressed*

if lawn_area is greater than 10000

set quote = lawn_area * 2.00

else

if lawn_area is greater than 1000

set quote = lawn_area * 3.50

else

set quote = lawn_area * 3.00

if garden_furniture is “Yes”

set quote = quote + (quote * 0.20)
if lawn_area is greater than zero
    set garden_waste = garden_waste - 1
if gardenWaste is greater than zero
    set quote = quote + garden_waste * 10
if generalWaste is greater than zero
    set quote = quote + 20 + general_waste * 15
set quote = quote + gardenMaintenance * 70
display “Quote is: $” quote

Many students did well in this section and I would encourage all students to attempt the A rating questions as valuable marks towards a C or B rating can be obtained. The main fault was students either not reading the instructions clearly or forgetting to implement them!

I would encourage students to collect all the data from each textfield and then make the calculation in a calculate method. Doing partial calculations as data is entered into each textfield is possible but then the sequence in which the users enter the data in the textfields can lead to unforeseen errors.

The main errors were not checking that generalWaste is greater than zero (and hence adding $20 to the cost when there was no general waste), neglecting to reduce the garden waste by 1 when the lawns had been mowed (not having a variable to indicate if there was furniture) and some confusion that there was either garden waste or general waste but not both (read the question carefully).

When answering this type of question, you do need to clearly show a well-structured, initial/when model.

Section B
Comment
A number of students who were only a few marks short of a “C” missed out on valuable partial marks either by failing to explain their answers or by stopping at Question 4.

Question 4
(a)  (i) \[ a = 4 + (1 \times 3) = 4 + 3 = 7 \]
Generally answered well, although many students either ignored or misapplied order of operations, coming up with answers such as \((4 + 1) \times 3 = 15\), or \((4 \times 3) + 1 = 13\), respectively.

(ii) \[ b = 2 + 3 = 5 \]
This was the most challenging part of Question 4. Many students were not able to correctly handle the modulo (\%) operator:

(b)  (i) \[ c = 3 \text{ then } c = 2 \]
Examiner comment
Generally answered well. A common mistake was to apply both the first and second if statements, as if they weren’t nested, giving \(c = 5\).

(ii) \[ d = 0 \text{ then } d = 4 \text{ then } d = 8 \text{ then } d = 6 \text{ then } d = 10 \]
Generally answered well.

<table>
<thead>
<tr>
<th>i</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
</tr>
</tbody>
</table>

A number of students either terminated the loop one cycle too early or one cycle too late. Other misapplied the cases within the switch statement.
Question 5
(a)  
(i)  \( f = 11.0 \) then \( f = 10.0 \) then \( f = 30.0 \)

Generally answered well.

(ii) \( h = 2 \)

Some students correctly recognised that the method call to \( \text{aaa} \) would return false, but gave an answer of 6, as if the result used in the method call (\( h * 3 \)) had been assigned to \( h \).

(b)  
Array \( k \) is

\[
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
'B' & 'D' & 'F' & 'G'
\end{array}
\]

This was difficult for most students. A relatively common mistake was to continue the loop an extra cycle. A number of students seemed to have difficulty with converting from char to int and back again, with some treating the addition within the loop as concatenation.

(c)  
Array \( m \) is

\[
\begin{array}{cccc}
m & 0 & 1 & 2 & 3 \\
0 & 0 & 2 & 3 & 4 \\
1 & 1 & 3 & 3 & 4 \\
2 & 1 & 2 & 6 & 4
\end{array}
\]

Generally answered well by students that attempted it.

Question 6
(a)  
(i)  

\[
\begin{array}{cccccc}
\text{secret} & 0 & 1 & 2 & 3 & 4 & 5 \\
C & O & V & E & R & T \\
\text{encoding} & 67 & 79 & 86 & 69 & 82 & 84 \\
\text{encoding} & 86 & 79 & 82 & 69 & 67 & 84 \\
\text{encoding} & 88 & 81 & 84 & 71 & 69 & 86 \\
\text{encrypted} & X & Q & T & G & E & V
\end{array}
\]
This was a particularly challenging part for most students, with very few getting it entirely correct. Out of the five traces, the first two were the easiest, and the third trace was the most difficult.

a) (ii) Original text:
   1. COVERT

ii. Encrypted text:
   1. XQTVGEV

Of the students that had attempted the previous part, about half successfully answered this part. Most of the other students didn’t attempt this one, even though partial marks could be obtained without even attempting the previous part.

b) The method `finalise` would produce a variable undefined error as `intArray` which was a parameter would now be undefined.

c) When the method `finalise` is called the value in the array variable `encoding` is placed in the array variable `intArray`. The value of the variable `encoding` is the pointer to the array contents. This means both array variable are pointing to the same array contents so any change to `encoding` array will in fact be a change to `intArray` array contents.

Relatively few students attempted the last two parts of this question. Of those that did, more than half were able to demonstrate a good understanding of the principles involved.

Section C

Section C was answered poorly by most students. The common denominator that I have been able to determine is that students do not seem to be using their reading time and are simply not reading what the basic requirement of each question is.

Question 7

(a)(i) codeField: 1
   aField: X
   bField: X
   cField: Z

This question asks what would be displayed after the code is run with a given input. The answer requires three outputs to be completed. Many students gave only the outputs that had changed, which was an unacceptable answer, whilst other students gave me the names of variables (or numeric answer) both of which are not "displayed" by an Applet – the values of the variables are and as such it should have been the values that were provided.

(ii) codeField: 2
    aField: X
    bField: Y
    cField: ZYX

This question asks what would be displayed after the code is run with a given input. The answer requires three outputs to be completed. Many students gave only the outputs that had changed, which was an unacceptable answer, whilst other students gave me the names of variables (or numeric answer) both of which are not "displayed" by an Applet – the values of the variables are and as such it should have been the values that were provided.
There was a tendency amongst a majority of students to either get both of the previous questions right and fail at this one or succeed at this one and fail at the previous two. It was rare to get a student that was competent across all three questions. Many of the people that understood how this question (and hence the substring method) worked often failed at the last hurdle by placing the word as “CAKED!” rather than “CAKE!” This is surprising as the previous two steps were done correctly using the same method but for some reason they lost that focus on the last section. For those of you that believe the cake is a lie – well done.

**Question 8**

This question was above all others in this section the most poorly answered question. 75%+ of respondents made no attempt at utilising the code that was on the facing page although all questions related to it and required it’s use.

(a)  
```java
Leg leg1 = new Leg('A', 5.3, 32);
Leg leg2 = new Leg('B', 3.1, 61);
```

There was confusion in this questions by students because they felt the layout of the question meant that there was an ID object. This was clearly just an identifier for two legs that needed to be declared and instantiated. For those that did not attempt to create their own solution by using the Leg object code on the other page many instantiated well but few actually declared the objects;

```java
Leg a, b;  
Leg a = new Leg(...
```

are examples of declaration whilst;

```java
a = new Leg('A', 5.3,32);  
Leg a = new Leg('A', 5.3,32);
```

are examples of Instantiation and the question asked for them both to be present.

(b)  
```java
leg1.setDistance(5.6);
```

Most students attempted to create their own methods or just change the variable directly. The above answer was the only valid answer for the question and few students managed to provide it.

(c)  
```java
double angle = leg2.bearingChange(leg1);
```

Many students missed the variable to catch the return of this method that managed to put it in place and most students also missed the fact that it had to be the second leg that called the method using the first leg as the parameter.

**Question 9**

This question was attempted by many students regardless of if they had attempted the B level question. This question also showed that many more students had the ability to create their own class rather than simply use a pre-created one as shown by Question 8.

(a)  
```java
public class Player
{
    public String name;
    public int goals, points, total;

    public Player(String newName, int newGoals, int newPoints, int newTotal)
    {
        name = newName;
        goals = newGoals;
        points = newPoints;
        total = newTotal;
    }
```

2015 Assessment Report
public String getName()  
{  
    return name;  
}  
public int getGoals()  
{  
    return goals;  
}  
public int getPoints()  
{  
    return points;  
}  
public int getTotal()  
{  
    return total;  
}  

Most students that attempted this question did reasonably well. The most common errors were students providing setter methods rather than getters or providing a return type for the constructor (which does not exist for the constructor)

(b) public Player(String newName, int newGoals, int newPoints)  
{  
    name = newName;  
    goals = newGoals;  
    points = newPoints;  
    total = goals * 6 + points;  
}  

Some students actually provided this in question (a). This question though had most students simply creating a method for this to occur when the question specifically asks for another constructor to do it. As a result there were many students that failed this question.

(c) (i) public void addWeek(int newGoals, int newPoints)  
{  
    goals = goals + newGoals;  
    points = points + newPoints;  
    total = total + goals * 6 + points;  
}  

On the whole, done very well. The most common problem encountered was that the total was not updated at the end of the method. Some updated it at the start (order matters) whilst others did not update it at all.

(ii) public boolean accurate()  
{  
    return (goals > points);  
}  

This question seemed to cause most to try and create a class called “accurate” and this was all done very poorly. If that was not the case many gave method headers that included void as the return type though it had to return a boolean. Others returned something only when it was true. It did seem that this question was only answered well by those that achieved high level A results for this section.
Section D
Question 10

(a) (i)  

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>~B</th>
<th>A ∨ ~B</th>
<th>~A ∧ ~B</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(ii)  

E = ¬A ∧ ¬B  or  E = ¬( A ∨ B )

(b) (i)  

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>~B</th>
<th>~B ∧ C</th>
<th>A ∨ (~B ∧ C)</th>
<th>~(A ∨ (~B ∧ C))</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

(ii)  

(c) For a Java Applet to be run by an Internet Browser it must first be compiled by a Java compiler to bytecode. Then the Internet Browser must have the Java Virtual Machine (JVM) built into the browser. The JVM converts the bytecode into the machine language of that specific computer for it to run.

This section was attempted successfully for most students.

Many students made an error in a(i) in the very final column, though partial credit was still awarded and follow through marks were awarded for the answer in a(ii)

Part (b) was done very well. The most common error was in misreading the question and missing the “(“ between the ∼ and the A. This caused a problem in part (i), including ∼A in a column, even though it wasn’t necessary. This caused students to run out of columns. It caused a problem in part (ii) where the NOT gate was put before the OR gate, rather than afterwards. Some students simplified the statement first, giving a simplified logic circuit, for which full marks were awarded.

Part (c) got lots of strange and varied answers, many giving instructions on writing a Java applet (requires an init() method, etc). Lots of partial credit given for anyone mentioning Bytecode or the JVM.
Question 11

(a) 

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(i) 

(ii) \( G = (\neg C \land \neg D) \lor (B \land \neg D) \lor (\neg B \land C \land D) \)

(iii) 

(b) 

(i) 

\[ c = 1; \]
\[ \text{if } ((a == 1) \&\& ((b == 2) || (a != 1))) \]
\[ c = 4; \]

\[ c = 1; \]
\[ \text{if } ((a == 1) \&\& (b == 2)) \] [L28]
\[ c = 4; \]
(ii) \[ D = (\neg C) \land ((L \land \neg R) \lor (\neg L \land R)) \]

OR \[ D = (\neg C) \land (L \oplus R) \]

OR \[ D = (\neg C) \land ((L \lor R) \land \neg(L \land R)) \]

(iii) With the von Neumann architecture the programs were then stored in the memory device within the computer as a series of code numbers. This meant that modification of programs was achieved by changing the code numbers used, with no change in hardware required.

Part (a)(i) was very well done

Part (a)(ii) was done poorly, with many students showing little understanding of developing a logic expression from a Karnaugh map. Many students also gave a correct but inefficient solution, to which a minor marking penalty was applied.

Part (a)(iii) was done very well, with follow through marks awarded if the student draw a circuit that correctly modelled their incorrect answer from part (ii)

Part (b)(i) was done poorly, with many students incorrectly invoking the Law of Excluded Middle (L14) to finish with an incorrect answer. Many students realised through common sense that the \((a \neq 1)\) is redundant and came up with the correct answer without invoking a logic law. Partial credit was awarded for this.

Part (b)(iii) was done very well, with many students using the XOR symbol “\(\oplus\)” correctly. It should be noted that this symbol is not in the course content, but is included in the information booklet for completeness. It was also quite possible to do this problem without using the XOR symbol.

Part (c) once again got many varied answers. Credit was awarded for any answers that mentioned the use of memory to store operations or instructions and partial credit for mentioning something else about von Neumann architecture (usually “Fetch-Decode-Execute”)

**Question 12**

(a) (i)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0100 1001 0101 0100 &lt;sub&gt;2&lt;/sub&gt;</td>
<td>0000 0000 &lt;sup&gt;1111 1111&lt;/sup&gt; &lt;sub&gt;2&lt;/sub&gt;</td>
<td>0000 0000 0101 0100 &lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

The contents of location 04 will be 0000 0000 0101 0100<sub>2</sub>

(ii)

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000 0000 1000 &lt;sub&gt;2&lt;/sub&gt;</td>
<td>0100 1001 0000 0000 &lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
<tr>
<td>0000 0000 0100 1001 &lt;sub&gt;2&lt;/sub&gt;</td>
<td>0000 0000 0100 1001 &lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

The contents of location 04 will be 0000 0000 0100 1001<sub>2</sub>
(b) (i)

<table>
<thead>
<tr>
<th>Memory Address</th>
<th>Contents</th>
<th>Pseudocode</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>4954</td>
<td>data</td>
<td>(0100 1001 0101 0100_2, 18772_10)</td>
</tr>
<tr>
<td>02</td>
<td>00FF</td>
<td>data</td>
<td>(0000 0000 1111 1111_2, 255_10)</td>
</tr>
<tr>
<td>03</td>
<td>0008</td>
<td>data</td>
<td>(0000 0000 0000 1000_2, 8_10)</td>
</tr>
<tr>
<td>04</td>
<td>0000</td>
<td>data</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>0000</td>
<td>data</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>8101</td>
<td>R[1] ← mem[01]</td>
<td>Set Register 1 to the contents of location 01</td>
</tr>
<tr>
<td>11</td>
<td>8202</td>
<td>R[2] ← mem[02]</td>
<td>Set Register 2 to the contents of location 02</td>
</tr>
<tr>
<td>12</td>
<td>8303</td>
<td>R[3] ← mem[03]</td>
<td>Set Register 3 to the contents of location 03</td>
</tr>
<tr>
<td>18</td>
<td>9405</td>
<td>mem[05] ← R[4]</td>
<td>Store Register 4 in location 05</td>
</tr>
</tbody>
</table>

(ii)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0100 1001 0101 0100_2</td>
<td>0000 0000 1111 1111_2</td>
<td>0000 0000 0000 1000_2</td>
<td>0000 0000 0101 0100_2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(c) (i)

Memory location 04 will contain 0000 0000 0101 0100_2 or 84_10 which corresponds to the letter ‘T’

Memory location 05 will contain 0000 0000 0100 1001_2 or 73_10 which corresponds to the letter ‘I’

(ii)

Char char4 = ‘T’;  // character stored in location 04

char char5 = ‘I’;  // character stored in location 05

int int1 = (int)Math.pow(2,8);

int int2;

int2 = (int)char5*int1+(int)char4;
Part (a)(i) was done incredibly poorly, with about 1% of students able to correctly use the Bitwise AND operation.

Part (a)(ii) was done better, since most students better understood the Right Shift operation. However, there were still a lot of errors demonstrating poor understanding, including dividing the answer by 8 (instead of by $2^8$) or by rightshifting the decimal representation (18772) to get 0.00018772.

Part (b)(i) was written to use the Bitwise AND and the Right Shift operation from parts (a)(i) and (ii). Given that so few students understood the Bitwise AND operator, not many were able to use it correctly here. However, many creative solutions were given to the problem that avoided using the Bitwise AND operator. The most common method of stripping the leading byte was to Left Shift by 8, then Right Shift by 8. Full marks were awarded for any solution that worked. Many students incorrectly used subtraction to get a result, either subtracting $R[2]$ from $R[1]$ or left shifting $R[2]$ first.

Part (c) was done well by any students who attempted it, but most did not. This question simply relied on looking up values on the ASCII table to earn some marks, and then understanding that multiplying by $2^8$ is the same as a bitwise left shift by 8 to earn full marks on this part. It should be reiterated that students should attempt every question on the exam, and that just because Q12 a) and Q12 b) were difficult TOY Machine questions, Q12 c) was relatively straightforward.

**Question 13**

(a) 

\[
\begin{array}{cccc}
1 & 1 & 1 & 1 \\
+ & 0 & 0 & 1 \\
= & 1 & 0 & 1 & 0 \\
\end{array}
\]

(b) (i) 6D

(ii) $13 - 6 = 01101 - 00110 = 01101 + (11001+1) = 01101 + 11010 = 100111 = 00111 = 7$

(iii) 16

(c) Letters can be represented by 0 to 25 this will require 5 bits (max 31 values). Digits go from 0 to 9 this will require 4 bits (max 15 values). Total is $2 \times 5 + 3 \times 4 = 22$ bits.

13(a) Well answered.

13 (b) (i) Many candidates gave the ASCII code in decimal or binary, and failed to convert to the required hexadecimal value.

13 (b) (ii) Well answered.

13 (b) (iii) Many candidates erroneously assumed '0' couldn’t be used to represent a colour.

13(c) Most candidates had difficulty with this question. Most attempted to calculate total number of combinations, rather than the amount of space required to store the data.

**Question 14**

(a) (i) The representations used to store integers will use the fact that the most significant bit is one to identify the value as negative. So it needs to be zero for a positive value.
(ii) The largest value in a 12-bit word in binary is 01111111111. This has the most significant bit as 0 and all the rest as 1. This is equal to $2^{11} - 1 = 2047$

(b) If a 12-bit word was divided up into a sign bit a 5 bit exponent and the 6 bit mantissa it would be able to store larger values than the integer representation. In the floating point representation it is the mantissa that determines the accuracy. Since only 6 bits are used for the mantissa in will be less accurate than the integer representation which uses 10 bits to store the value.

(c) Integer calculations are done with 100% accuracy. However, floating point calculations are done with less than 100% accuracy. In this case the value of $a \times a + b \times b - c \times c$ should be 0 but due to inaccuracy in the calculation it is not exactly zero.

The second calculation allows for this inaccuracy by accepting any value within 0.00001 of zero and so is able to determine that it is a right angled triangle.

14 (a) (i) Generally well answered.
14 (a) (ii) Most students interpreted this question to be asking for the largest unsigned 12-bit integer, ie $2^{12} - 1 = 4095$ which was accepted. Alternatively, as a continuation of 14 (a) (i) (which referred to signed integers), $2^{11} - 1 = 2047$ - was also acceptable. Many candidates neglected to subtract the 1 to account for zero.

14 (b) Generally poorly explained - though many candidates appeared to have some rudimentary understanding of the basic principle regarding accuracy.

14 (c) Generally poorly explained - though many candidates appeared to have some rudimentary understanding of the basic principle regarding accuracy.

**Question 15**

(a) (i) Values from 0 to 4095 uses 12 bits. Each pixel will use $4 \times 12$ bits = 48 bits
The image will use $48 \times 640 \times 480 = 14745600$ bits.

(ii) Values from 0 to 256 uses 8 bits. Each pixel will use $3 \times 8$ bits = 24 bits
The image will use $12 \times 640 \times 480 = 7372800$ bits. (This is half the size of the RAW file.)

(b) Value of array variable $p$:

<table>
<thead>
<tr>
<th>Memory Address</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC00</td>
<td>5</td>
</tr>
<tr>
<td>AC01</td>
<td>7</td>
</tr>
<tr>
<td>AC02</td>
<td>9</td>
</tr>
<tr>
<td>AC03</td>
<td>8</td>
</tr>
<tr>
<td>AC04</td>
<td>6</td>
</tr>
<tr>
<td>AC05</td>
<td>4</td>
</tr>
<tr>
<td>AC06</td>
<td>-</td>
</tr>
<tr>
<td>AC07</td>
<td>-</td>
</tr>
<tr>
<td>AC08</td>
<td>-</td>
</tr>
<tr>
<td>AC09</td>
<td>-</td>
</tr>
</tbody>
</table>

15 (a) (i) Most students included the $640 \times 480$ as part of their calculations, but often were unable to accurately determine what else was required for the calculation.

15 (a) (ii) Always answer all parts of the question. Some candidates performed the calculation, but neglected to make the comparison with 15a (i).

15 (b) Many candidates gave the length for $p$ as their answer, rather than supplying the appropriate memory address.

15 (c) The array length was often omitted from the contents of memory. Also, the array variable $p$ often had non-memory address data entered for the answer.